Topological Dualities
in
Semantics

Marcello M. Bonsangue

October 7, 1996

Topological Dualities in Semantics

Bonsangue, Marcello
Topological Dualities in Semantics / Marcello M. Bonsangue
Amsterdam, Vrije Universiteit Amsterdam
Ph.D. Thesis with index, keywords index, and summary in Dutch
ISBN 90-9009958-1

Printed by Universal Press, Veenendaal

Cover design: Frouke M. Visser
Cover photo: Dolomiti Auronzane (Italia)

The work in this thesis has been carried out under the auspices of the research school IPA (Institute for Programming research and Algorithmics).

VRIJE UNIVERSITEIT

Topological Dualities in Semantics

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor aan
de Vrije Universiteit te Amsterdam,
op gezag van de rector magnificus
prof.dr. E. Boeker,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der wiskunde en informatica
op donderdag 21 november 1996 te 15.45 uur
in het hoofdgebouw van de universiteit,
De Boelelaan 1105

door

MARCELLO MARIA BONSANGUE

geboren te Caltanissetta, Italië

Promotoren:	prof.dr. J.W. de Bakker
	prof.dr. J.N. Kok
Referent:	prof.dr. M.W. Mislove

8 marzo 1996

8 marzo, festa della donna.
Tu l' hai sempre festeggiata,
ma quest'anno non sei riuscita a vedere
le mimose che ti abbiamo portato.
Il tuo sorriso ci ha lasciato
ma non lo dimenticheremo.

Adesso cara mamma riposi in pace
e la calma è con te.

A mia madre e a mio padre

Preface

When I was a child I had a dream: climbing a mountain. And it remained just a dream till I came to Amsterdam for the first time as an Erasmus-student in 1989. It was Vincenzo Lo Cascio of the Universiteit van Amsterdam who showed me the attractive mountaineering area which the Dutch academic world is. Back in Italy, my interest for the Dutch mountains was noticed by Nicoletta Sabadini and Giancarlo Mauri of the Università degli Studi di Milano. They not only believed that climbing a mountain in a land that finds itself for a considerable part below sea-level would create a perfect surrealistic scenery which suited me best, but also that it was worth while to provide me financial assistance by means of two grants: one of the Università degli Studi di Milano and another of the Centro Nationale delle Ricerche (CNR).

In Amsterdam I decided to pitch my base-camp at the Centrum voor Wiskunde en Informatica (CWI) since I thought it the perfect point of departure for the realization of my dream. The CWI is located in a scientific valley from which countless paths start in more directions of mathematics and computer science than anyone would care to count. The atmosphere at the CWI is peaceful and friendly, the resources are excellent (not to forget the well-equipped library), and people are always open to discussion.

It gave me much pleasure to discover that I was supposed to spend my first Dutch winter in the hottest bivouac I had ever seen. It was both for me and for my Indian colleague Krishna Rao the perfect place to become acclimatized to the Dutch climate. When spring came, I moved to the office M335 where I shared my passion for mountains with Franck van Breugel and Daniele Turi. I will always remember our endless discussions about which itinerary to take or the way to tackle a mathematical problem.

At the CWI I met Jaco de Bakker who immediately accepted to guide me through the Dutch highlands, allowing me total freedom to choose my own direction. His advice, appropriate questions and critical comments have strongly influenced my decisions in the past years. His attitude towards theoretical computer science has always been an example to me of how to do research in this field. I am particularly grateful to him for introducing me to the Amsterdam Concurrency Group. My mountaineering experience along the slopes of the semantical crags of programming languages owes much to the insights and the criticisms of the members of this group, headed by Jaco de Bakker and including Frank de Boer, Franck van Breugel, Arie de Bruin, Jerry den Hartog, Bart Jacobs, Jean-Marie Jaquet, Kees van Kemenade, Joost Kok, Jan Rutten, Daniele Turi, Erik de Vink, Michiel van Wezel, and Herbert Wiklicky.

At the CWI I also met Joost Kok; I was pleased when he accepted me as his rope-mate and introduced me in the rudiments of climbing techniques. Later

on he became, together with Jaco de Bakker, my instructor. I would never have conquered so many mountains if it were not for his enthusiasm, his positive attitude and his friendship. His perseverance, stimulating questions and, above all, his continuous encouragements meant, and still mean, a great deal to me.

I will always remember my first careful steps along the mathematical paths I was most attracted to. I am grateful to Jeroen Warmerdam for having shown me the beautiful prospects of 'domain theory'. I am grateful to Jan Rutten and Daniele Turi for their willingness to climb with me the relatively young hills of 'category theory'. Daniele's excitement for category theory has always been a great stimulus for me to go forward. Bart Jacobs often choose the right moment to be there, ready to correct my innumerable mistakes and to suggest a different path to follow. I am thankful to Jan Rutten and Franck van Breugel for their company during my expeditions through the ancient mountains of 'topology' and 'metric spaces'. It was during one of these expeditions that I got to know Michael Mislove. He immediately showed interest for my work and encouraged me to continue along the itinerary of 'topological dualities' I had just started.

When my financial assistance from Italy finished, Jaco de Bakker provided me another base-camp in Amsterdam: the Vrije Universiteit. My knowledge of the Dutch language improved and I broke new ground with teaching. I am grateful to Franck van Breugel for offering me his generous help which, of course, I accepted. I thank all the students who attended my tutoring classes: I hope they learned something from me, I certainly learned a lot from them.

At the Vrije Universiteit I had many inspiring occasions during which I exchanged ideas with the members of the research group on theoretical computer science: Jaco de Bakker, Allegra Bencivenni, Stefan Blom, Mirna Bognar, Franck van Breugel, Jerry den Hartog, Jan Willem Klop, Vincent van Oostrom, Yde Venema, Erik de Vink, and Roel de Vrijer. Moreover, I spent a very enjoyable time with Vincent, Franck, Allegra and Mirna as roommates. I thank them all for having created such a good atmosphere to work in.

There are many routes to reach the top of a mountain, and there are several ways to approach them. I learned different techniques for mountain-climbing from each of my co-authors. I am grateful to Joost Kok, Bart Jacobs, Erik de Vink, Marta Kwiatowska, Jan Rutten, and Franck van Breugel for a very fruitful and enjoyable collaboration.

Being a member of the Dutch climbing-school Instituut voor Programmatuurkunde en Algoritmiek (IPA), I had the possibility to exchange my experiences with many other young climbers. The annual meetings organized by IPA were always a nice occasion to get to know other people with whom I could share my interest for the highlands.

Often paths cross each other and so did mine: during my hikes I often met people who strongly influenced my thoughts. Among these people I am particularly thankful to Samson Abramsky, Ralph Back, Bob Flagg, Paul Johnson, Peter Knijnenburg, Paul-André Mellies, Prakash Panangaden, Mike Smyth, Philippe Sünderhauf, and Fer-Jan de Vries.

Since I came to like staying abroad, from time to time I moved around the world, always in search of different sights. I gratefully acknowledge the financial assistance received from the Vrije Universiteit Amsterdam, the Netherlands Organization for Scientific Research (NWO), Shell Nederland B.V., the NATO Advanced Study Institute, the Dutch project 'Research and Education in Concurrent Systems' (REX), and the European Community SCIENCE project 'Mathematical Structures in Semantics of Concurrency' (MASK) which made possible my visits to several conferences and workshops, and to a summer school. Thanks also to the people who invited me to present parts of the results on which this thesis has been based: Maurice Nivat and Paul Gastin of the Université de Paris, Marta Kwiatowska of the University of Birmingham, and Willem-Paul de Roever and Kai Engelhart of the Christian-Albrechts Universität zu Kiel. During my visits to them I benefited from many fruitful discussions and I had a very enjoyable time.

Like all travelers, I kept a journal during my mountaineering years in Holland. It was not an easy task to write down the scientific experiences I acquired during my climbing-expeditions. Since I knew better where to end than where to start, I decided to write this thesis starting from the last chapter and ending with the preface. I wish to express all my gratitude to my supervisors, Jaco de Bakker and Joost Kok, who carefully went through all successive drafts, often adapting themselves to my reverse view on my research; they provided an uncountable number of improvements both in the presentation as in the content. Without their indefatigable help this thesis would not have been as readable as it now is.

I am also grateful to Kiki Hannema, Jerry den Hartog, Vincent van Oostrom, Jan Rutten, Yde Venema, and Erik de Vink for critically reading parts of my thesis. A special thank goes to Frouke Visser for her improvements in the presentation of this preface. Frouke and Jan were of great help in translating the abstract of this thesis in Dutch. I am indebted to Michael Mislove for his willingness to referee my work and for his incisive comments. Thanks also to Wim Hesselink, Jan-Willem Klop, Jan van Mill, Jan Rutten, and Yde Venema for taking part in the opposition and for their interest in my work.

I enjoyed wonderful views from the summit of the mountains I climbed, every time looking forward to the next expedition, and never forgetting that some people were always with me in thought. Therefore I would like to thank my family and Frouke for their love and support. Frouke's encouragements through all the years that we happily shared together have made possible the realization of this, and of several other dreams of mine. My parents always had a never ending interest in my work. My mother, who understood so well my enthusiasm for mountaineering since she was born in the Dolomites, will never know that my climbing-dream came true.

The diagrams in this thesis were designed using the TEX macros package of Paul Taylor.

Contents

Chapter 1

Introduction

A *language* is a systematic means of communicating ideas or feelings among people by the use of conventionalized signs. In contrast, a *programming language* can be thought of as a syntactic formalism which provides a means for the communication of computations among people and (abstract) machines. Elements of a programming language are often called *programs*. They are formed according to formal rules which define the relations between the various components of the language. Examples of programming languages are conventional languages like Pascal [Wir71] or C++ [Str91], and also the more theoretical languages such as the λ-calculus [Chu32, Bar84] or CCS [Mil80].

A programming language can be interpreted on the basis of our intuitive concept of computation. However, an informal and vague interpretation of a programming language may cause inconsistency and ambiguity. As a consequence different implementations may be given for the same language possibly leading to different (sets of) computations for the same program. Had the language interpretation been defined in a formal way, the implementation could be proved or disproved correct. There are different reasons why a formal interpretation of a programming language is desirable: to give programmers unambiguous and perhaps informative answers about the language in order to write correct programs, to give implementers a precise definition of the language, and to develop an abstract but intuitive model of the language in order to reason about programs and to aid program development from specifications.

Mathematics often emphasizes the formal correspondence between a notation and its meaning. For example, in mathematical logic, we interpret a formal theory on the basis of a more intuitive mathematical domain which properly fits the theory (that is, the interpretation of all theorems must be valid). Similarly, the formal *semantics of a programming language* assigns to every program of the language an element of a mathematical structure. This mathematical structure is usually called the *semantic domain*. Several mathematical structures can be used as semantic domain, and the choice as to which one is to be preferred often depends upon the programming language under consideration. Since a programming language is a formal notation, its semantics can be seen as a translation of a formal system into another one. The need for a formal semantics of a programming language can thus be rephrased as the need for a suitable mathematical structure closer to our computational intuition. From this mathematical structure we expect to gain insights into the language considered.

There are several ways to formally define the semantics of a programming language. Below we briefly describe the three main approaches to semantics, namely the operational, the denotational and the axiomatic approach. Other important approaches to the semantics of programming languages are given by the *algebraic semantics* [GTW78, Mil80, Hoa85, BK86, BWP87, Hen88], with mathematical foundations based on abstract algebras [GTW$^+$77, EM85], and the *action semantics* [Mos92], based on three kinds of primitive semantics entities: actions, data and yielders.

In the *operational semantics* one defines the meaning of a program in terms of the computations performed by an abstract machine that executes the program. For this reason the operational semantics is considered to be close to what actually happens in reality when executing a program on a real computer. *Transition systems* are the most commonly used abstract machines which support a straightforward definition of a computation by the stepwise execution of atomic actions. There are different ways to collect the information about the computations of a transition system which give rise to different operational semantics. Moreover, transition systems support a structural approach to operational semantics as advocated by Plotkin [Plo81b]: the transition relation can be defined by induction on the structure of the language constructs.

The *denotational* approach to the semantics of programming languages is due to Scott and Strachey [SS71]. Programs are mapped to elements of some mathematical domain in a compositional way according to the 'Fregean principle' [Fre77]: the semantics of a language construct is defined in terms of its components. Due to the possibility of self-application given by some programming languages, the semantic domain must sometimes be defined in a recursive way. This is often impossible with an ordinary set-theoretical construction because of a cardinality problem. Therefore often a topological structure is associated with the semantic domain which takes into account qualitative or quantitative information about the computations. Typical topological structures used for the denotational semantics of programming languages are complete partial orders, put forward by Scott [Sco70, Sco76], and complete metric spaces, introduced in semantics by Arnold and Nivat [AN80] and extensively studied by the Amsterdam Concurrency Group (for an overview see [BR92] and also [BV96]). The denotational semantics is close to the operational semantics but abstracts from certain details so that attention can be focussed on issues at a higher level.

The *axiomatic* approach characterizes programs in a logical framework intended for reasoning about their properties. Proof systems are usually used for axiomatic semantics: computations are expressed by relating programs to assertions about their behaviour. The most well-known axiomatic semantics is Hoare logic [Hoa69] for total correctness. Assertions are of the form $\{P\}\, S\, \{Q\}$ meaning that the program S when started at input satisfying the predicate P terminates and its output satisfies the predicate Q. There are many other kinds of axiomatic semantics using proof systems such as temporal logic [Pnu77], dynamic logic [Pra81] and Hennessy-Milner logic [HM85]. Axiomatic semantics can also be given without the use of formal proof systems: the behaviour of a program can be expressed as a function which transforms predicates about the program. For example, Dijkstra's weakest precondition semantics [Dij76] regards a program S as a function which maps every predicate Q on the output state space of S to the weakest predicate among all P's such that the Hoare assertion $\{P\}S\{Q\}$ is valid. Axiomatic semantics is closely related to the verification of the correctness of programs with respect to a given specification. An axiomatic semantics should preferably be such that the verification of the correctness of a program can be done by verifying the correctness of its components, as advocated by Turing [Tur49] and Floyd [Flo67] (see also the discussion in [Bak75]).

The choice among the operational, the denotational or the axiomatic semantics for a programming language will depend on the particular goals to be achieved. To take advantage of these different semantic views of a program it is important to study their relationships.

The denotational semantics of a programming language is, by definition, compositional. Since an operational semantics is not required to be compositional we cannot have, in general, an equivalence between the two semantics. Two criteria about the relation between denotational and operational semantics are commonly accepted. The first criterion says that the denotational semantics has to assign a different meaning to those programs of the language which in some context can be distinguished by the operational semantics. This can be achieved, for example, by proving the existence of an abstraction function that when composed with the denotational semantics gives exactly the operational semantics. In this case the denotational semantics is said to be *correct*, or adequate, with respect to the operational semantics. The second criterion looks for the most abstract denotational semantics which is correct with respect to a given operational semantics. This can formally be expressed by requiring that the denotational semantics assigns a different meaning to two programs of the language if and only if they can be distinguished in some context by the operational semantics. In this case the denotational semantics is said to be *fully abstract* with respect to the operational semantics [Mil73].

The relationship between the denotational and the axiomatic semantics is the main topic of this thesis. Depending on which kind of information has to be taken in account, there are different transformations which ensure the correctness of one semantics in terms of the other. The common factor in all these transformations is that they form dualities rather than equivalences: the denotational meaning of a program viewed as a function from the input to the output space is mapped to a function from predicates on the output space to predicates on the input space. Conversely, the axiomatic meaning of a program regarded as a function from predicates on the output to predicates on the input is mapped to a function from the input space to the output space.

The dualities between the denotational and the axiomatic views of a program are often *topological* in the sense that they are set in a topological framework. This is motivated by the tight connection between topology and denotational semantics: topology has become an essential tool for denotational semantics and denotational semantics has influenced new activities in topology [Smy83b, Abr87, Vic89, Smy92, Bre94, Mis95, BV96]. The fundamental insight due to Smyth [Smy83b] is that a topological space may be seen as a 'data type' with the open sets as 'observable predicates', and functions between topological spaces as 'computations'. These ideas form the basis for a computational interpretation of topology.

Abramsky [Abr87, Abr91b], Zhang [Zha91] and Vickers [Vic89, AV93] carried the ideas of Smyth much further by systematically developing a propositional program logic from a denotational semantics. The main ingredient in their work is a duality (in categorical terms a contravariant equivalence) between the category of certain topological spaces and a corresponding category of frames (algebraic structures with two classes of operators representing finite conjunctions and infinite disjunctions). On one side of the duality, topological spaces can arise as semantic domains for the denotations of programs; on the other side of the duality, frames can arise as the Lindenbaum algebras of a propositional program logic with properties as elements and proof rules provided by the various constructions. Accordingly, topological dualities are considered as the appropriate framework to connect denotational semantics and program logics [Abr87, Vic89, Zha91].

From a broader perspective, topological dualities (in the form of representation theorems) can be used to characterize models of abstract algebraic structures in terms of concrete topological structures. Therefore the ultimate purpose of setting up a topological duality is to capture axiomatically the class of properties we have in mind. Let us quote Johnstone [Joh82a, page XX] to summarize the importance of topological dualities: 'Abstract algebra cannot develop to its fullest extent without the infusion of topological ideas, and conversely if we do not recognize the algebraic aspects of fundamental structures of analysis our view of them will be one-sided.'

The contributions of this thesis may roughly be classified into the following three kinds. The first kind of contribution consists in the characterization for a given language of an axiomatic semantics using insights from a denotational semantics. For example, we define a weakest precondition semantics for a sequential language with a backtrack operator using a simple denotational interpretation. We also characterize a compositional predicate transformer semantics for a concurrent language with a shared state space. The semantics is based on a denotational interpretation of the language given by considering programs to be functions abstracted from a transition system modulo bisimulation [BZ82, GR89].

The second kind of contribution is dual to the first one: the characterization of a denotational semantics for a language using an axiomatic semantics. We characterize a denotational semantics for the refinement calculus (a language with an associated axiomatic semantics based on monotonic predicate transformers). We use the denotational semantics to derive a new operational interpretation of the refinement calculus based on hyper transition systems. The denotational semantics of the refinement calculus is proved fully abstract with respect to the operational interpretation (in fact they are equivalent).

The third kind of contribution is more abstract in nature. We have set up a framework for a systematic development of a propositional logic for the specification of programs from a denotational semantics. In particular, it gives a conceptual foundation which answers the question posed by Abramsky [Abr91b, page 74] about the possibility of expressing infinite conjunctions in the logic of domains. The logic derived from a denotational semantics by means of the duality between the category of certain topological spaces and the corresponding category of frames is not expressive enough to be used for specification purposes: infinite conjunctions should be added [Abr87, Zha91]. However, such an extension would necessarily takes us outside open sets. Our contribution consists in the the development of an abstract algebraic framework which allows both infinite conjunctions and infinite disjunctions of abstract open sets. This framework is related to ordinary topological spaces by means of a representation theorem, and it is applied by deriving an infinitary logic for transition systems.

Outline of the chapters

This thesis is divided into three parts. In the first part we consider predicates as subsets of an abstract set of states. In the second part we refine the notion of predicates by considering affirmative predicates. They are open subsets of an abstract set of states equipped with a topology. Finally, in the third part we forget about states and we take predicates to be elements of an abstract algebra with algebraic operations to represent unions and intersections.

We start by introducing in Chapter 2 some basic concepts in category theory, partial orders, and metric spaces. Category theory is not needed for understanding the first two parts of this thesis. Metric spaces will only play a major role in Chapter 7.

With Chapter 3 we start the first part of the thesis. The chapter is about the semantics of sequential languages. In particular we consider the weakest precondition and the weakest liberal precondition semantics, and the relationships to various state transformer semantics. These relationships generalize the duality of Plotkin [Plo79] between predicate transformers and the Smyth powerdomain. We also discuss the weakest precondition semantics of a sequential non-deterministic language with a backtrack operator.

In Chapter 4 we extend sequential languages with specification constructs. We use the language of the refinement calculus introduced by Back [Bac80]. The refinement calculus is based on a predicate transformer semantics which supports both unbounded angelic and unbounded demonic non-determinism. We give a state transformer semantics for the refinement calculus and relate it to the predicate transformer semantics by means of a duality. We give also an operational interpretation of the refinement calculus in terms of the atomic steps of the computations of the programs. The latter operational view is connected to the state transformer semantics.

The second part begins with Chapter 5. In this chapter we refine the notion of predicate introduced in Chapter 3. Following the view of Smyth [Smy83b], affirmative predicates are introduced as open sets of a topological space. Several basic concepts taken from topology are introduced and motivated from the point of view of affirmative predicates.

In Chapter 6 we rework in a topological framework the dualities between predicate and state transformers that were introduced in Chapter 3. These dualities show us how to generalize predicate transformers to topological predicate transformers. The latter can be used as domain for a backward semantics of non-sequential programming languages. Our starting point is Smyth's duality between the upper powerspace of a topological space and certain functions between affirmative predicates. We show that Smyth's duality holds in a general topological context. Also, we propose dualities for the lower powerspace and the (more classical) Vietoris construction on general topological spaces. In passing, several topological characterizations of metric and order based powerdomains constructions are investigated.

Chapter 7 is devoted to the semantics of a sequential non-deterministic language extended with a parallel operator. A domain of metric predicate transformers is defined as the solution of a recursive domain equation in the category of complete metric spaces. A compositional predicate transformer semantics is given to the language, and it is shown to be isometric to a state transformer semantics based on the resumption domain of De Bakker and Zucker [BZ82]. Partial and total correctness properties are studied for the above language using a connection between the domain of metric predicate transformers and the two domains of predicate transformers given in Chapter 3. As a consequence, the semantics of a sequential language is obtained as the abstraction of the unique fixed point of a metric-based higher-order transformation, and is proved correct with respect to three order-based semantics obtained as least fixed points of three higher-order transformations, respectively. Also, we briefly discuss the study of temporal properties of a concurrent language via our metric predicate transformer semantics.

The third and last part of the thesis starts with Chapter 8. We abstract from open sets and regard predicates as elements of an abstract algebra. We consider a topological space as a function from the abstract set of affirmative predicates (with algebraic operations representing arbitrary unions and finite intersections) to the abstract set of specifications (with algebraic operations representing arbitrary unions and arbitrary intersections). This structure is called an observation frame. We show that in certain cases topological spaces can be reconstructed from observation frames. We obtain a categorical duality between the category of certain topological spaces (not necessarily sober) and a corresponding category of observation frames. We also give a propositional logic of observation frames with arbitrary conjunctions and arbitrary disjunctions. The logic is shown to be sound and complete if and only if the observation frame corresponds, canonically, to a topological space. Finally we apply the above theory in order to obtain dualities for various sub-categories of topological spaces.

Chapter 9 relates topological spaces seen as frames to topological spaces seen as observation frames. A new characterization of sober spaces in terms of completely distributive lattices is given. This characterization can be used for freely extending the geometric logic of topological spaces to an infinitary logic. We also show that observation frames are algebraic structures in a precise categorical sense.

The thesis ends with Chapter 10. In this chapter an extension of Abramsky's finitary domain logic for transition systems to an infinitary logic with arbitrary conjunctions and arbitrary disjunctions is presented. To obtain this extension we apply the theory developed in the previous two chapters. The extension is conservative in the sense that the domain represented in logical form by the infinitary logic coincides with the domain represented in logical form by Abramsky's finitary logic. As a consequence we obtain soundness and completeness of the infinitary logic for the class of all finitary transition systems.

Interdependence of the chapters

The three parts of the thesis can be read almost independently. The logical interdependence between the chapters is schematically represented by the following diagram.

$$
\begin{tikzcd}
 & (2.1) \arrow[r] & 8 \arrow[r] & 9 \arrow[r] & 10 \\
 & (2.2) \arrow[ur] \arrow[d] \arrow[dr] & 5 \arrow[u] \arrow[d] & ([\text{Abr91a}]) \arrow[ur] & \\
 {} \arrow[r, Rightarrow] & 3 \arrow[r] \arrow[dr] & 6 \arrow[r] & 7 & \\
 & & 4 & (2.3) \arrow[u] &
\end{tikzcd}
$$

The sections between parentheses and the article [Abr91a] are only necessary as references to proofs.

Origins of the chapters

Several results of this thesis appeared in publications. Chapter 3 is mostly based on [BK93b] and [BK94b]. Chapter 6 is an extension of [BK93a] and [BK94a]. The first half of Chapter 7 is based on the paper [BKV95] while the second half is new. Finally, Chapter 8 is a revised version of the paper [BJK95]. Chapters 4, 9 and 10 contain mostly original material, while Chapter 5 follows ideas of Smyth originally presented in [Smy83b] and [Smy92].

Chapter 2

Mathematical preliminaries

In this chapter, we present some basic notions and properties from category theory, partial orders and metric spaces which we will need in the subsequent chapters. This chapter is not intended to provide a comprehensive introduction to the subjects treated. Rather, it is aimed to list those major facts we shall assume in the next chapters (providing references for their proofs), and to make the reader familiar with the notation we will use. The only original material is Proposition 2.3.3 (stemming from [BKV94]).

2.1 Category theory

In this section we introduce some concepts of category theory and universal algebra. Statements and facts in this section will be used only in the third part of this thesis. The first two parts will not need any categorical prerequisites. Unless stated otherwise, our reference for the categorical concepts below is to the book of Mac Lane [ML71].

We begin with the following three fundamental notions. A *category* $\mathbf{C}$ consists of

(i) a class of *objects*,

(ii) a class of *morphisms*. Each morphism has a domain and a codomain which are objects of the category. The collection of all morphisms with domain A and codomain B is denoted by $\mathbf{C}(A, B)$.

(iii) a *composition law* which assigns to each pair of morphisms $f \in \mathbf{C}(A, B)$ and $g \in \mathbf{C}(B, C)$ a morphism $g \circ f \in \mathbf{C}(A, C)$.

(iv) an *identity morphism* $id_A \in \mathbf{C}(A, A)$ for each object A such that for all morphisms f, $f \circ id_A = f$ and $id_A \circ f = f$ whenever the composites are defined.

For example, there is a category **Set** whose objects are sets and whose morphisms are functions with the usual composition. Similarly, $\mathbf{Set^2}$ is the category whose objects are functions $f : A \to B$ between two sets, and whose morphisms between $f : A \to B$ and $g : A' \to B'$ are pairs of functions $\langle h : A \to A', k : B \to B' \rangle$ such that $g \circ h = k \circ f$.

A *functor* $F : \mathbf{C} \to \mathbf{D}$ is a morphism between two categories. It assigns to each object A of $\mathbf{C}$ an object $F(A)$ of $\mathbf{D}$, and to each morphism $f \in \mathbf{C}(A, B)$ a morphism $F(f) \in \mathbf{D}(F(A), F(B))$ such that

(i) it preserves identity, i.e. $F(id_A) = id_{F(A)}$, and
(ii) it preserves composition whenever defined, i.e. $F(f \circ g) = F(f) \circ F(g)$.

A *natural transformation* $\eta : F \to G$ is a morphism between two functors $F, G : \mathbf{C} \to \mathbf{D}$. It maps each object A of $\mathbf{C}$ to a morphism $\eta_A \in \mathbf{D}(F(A), G(A))$ such that for all morphisms $f \in \mathbf{C}(A, B)$ we have $G(f) \circ \eta_A = \eta_B \circ F(f)$.

In a category $\mathbf{C}$ a morphism $f \in \mathbf{C}(A, B)$ is said to be an *isomorphism* (and A and B are called *isomorphic*) if there exists another morphism $g \in \mathbf{C}(B, A)$ such that $f \circ g = id_B$ and $g \circ f = id_A$. An object A of $\mathbf{C}$ is called a *fixed point* of a functor $F : \mathbf{C} \to \mathbf{C}$ if A is isomorphic to $F(A)$. A functor $F : \mathbf{C} \to \mathbf{D}$ is said to *reflect isomorphisms* if for each morphism f of $\mathbf{C}$, f is an isomorphism whenever $F(f)$ is.

The *opposite category* $\mathbf{C}^{op}$ of a category $\mathbf{C}$ has the same objects of $\mathbf{C}$ and $f \in \mathbf{C}^{op}(A, B)$ if and only if $f \in \mathbf{C}(B, A)$. Composition and identities are defined in the obvious way. A category $\mathbf{C}$ is a *full sub-category* of $\mathbf{D}$ if every object in $\mathbf{C}$ is an object in $\mathbf{D}$, and $\mathbf{C}(A, B) = \mathbf{D}(A, B)$.

The main concept of category theory is that of *adjunction*. For two functors $F : \mathbf{C} \to \mathbf{D}$ and $G : \mathbf{D} \to \mathbf{C}$, we say F is *left adjoint* of G if there is a bijection, natural in A and B, between $\mathbf{C}(A, G(B))$ and $\mathbf{D}(F(A), B)$. In this case the functor G is said to be *right adjoint* of F. Every adjunction induces two natural transformations: the *unit* $\eta : id_{\mathbf{C}} \to G \circ F$ (for each A, $\eta_A : A \to G(F(A))$ is the morphism which corresponds to $id_{F(A)} : F(A) \to F(A)$), and its dual, the *counit* $\varepsilon : F \circ G \to id_{\mathbf{D}}$ (for each B, $\varepsilon_B : F(G(B)) \to B$ is the morphism which corresponds to $id_{G(B)} : G(B) \to G(B)$). In general we do not need to know that F is a functor [ML71, page 81, Theorem 2]:

2.1.1. Proposition. *A functor $G : \mathbf{D} \to \mathbf{C}$ has a left adjoint if for each object A of $\mathbf{C}$ there exists an object $F(A)$ in $\mathbf{D}$ and a morphism $\eta_A : A \to G(F(A))$ such that for every other morphism $f \in \mathbf{C}(A, G(B))$ there is a unique $h \in \mathbf{D}(F(A), B)$ satisfying $G(h) \circ \eta_A = f$.* □

A *reflection* is an adjunction for which the counit ε_B is an isomorphism for all B. If both unit and counit are isomorphisms then the adjunction is called an *equivalence* and the categories involved are *equivalent*. We say that two categories $\mathbf{C}$ and $\mathbf{D}$ are *dual* if $\mathbf{C}$ is equivalent to $\mathbf{D}^{op}$. An adjunction is called *Galois* if it restricts to an equivalence between the categories $F(\mathbf{C})$ and $G(\mathbf{D})$ (here $F(\mathbf{C})$ denotes the full sub-category of $\mathbf{D}$ whose objects are in the image of F, and $G(\mathbf{D})$ denotes the full sub-category of $\mathbf{C}$ whose objects are in the image of G). An adjunction is Galois if and only if it restricts to a reflection from $\mathbf{C}$ into $F(\mathbf{C})$ [Isb72b].

Let $\mathbf{C}$ be a category and $\mathbf{J}$ be a *small category* (i.e. the objects and the morphisms of $\mathbf{J}$ form a set rather than a proper class). The category $\mathbf{C}^{\mathbf{J}}$ has as objects functors from $\mathbf{J}$ to $\mathbf{C}$ (in this context often called *diagrams*), and morphisms are natural transformations. There is a functor $\triangle : \mathbf{C} \to \mathbf{C}^{\mathbf{J}}$ which maps every object A to the constant functor with value A. We say that $\mathbf{C}$ has *limits* of type $\mathbf{J}$ if the functor $\triangle$ has a right adjoint $\lim_{\mathbf{J}}$, and we refer to $\lim_{\mathbf{J}}(D)$ as the limit of the diagram D. Dually, if $\triangle$ has a left adjoint then we say that $\mathbf{C}$ has *colimits* of type $\mathbf{J}$. If $\mathbf{J}$ is the empty category then the limit of a diagram is called *terminal object* while the colimit is called *initial object.* If $\mathbf{J}$ is a *discrete category* (i.e. one with only identity morphisms) then the limit of a diagram is called *product* and the colimit *coproduct.* Finally, if $\mathbf{J}$ is the category with two objects, two identity morphisms and two parallel morphisms between the two objects then the limit of a diagram is called *equalizer* and the colimit *coequalizer*.

A category is *complete* if it has all small limits, and *cocomplete* if it has all small colimits. Both categories $\mathbf{Set}$ and $\mathbf{Set^2}$ are complete and cocomplete.

A *monad* on a category $\mathbf{C}$ is a triple $\langle T, \eta, \mu\rangle$ where $T : \mathbf{C} \to \mathbf{C}$ is a functor, and $\eta : id_{\mathbf{C}} \to T$ and $\mu : T \circ T \to T$ are natural transformations satisfying some commutativity laws (see page 133, [ML71]). Every functor $F : \mathbf{C} \to \mathbf{D}$ which is a left adjoint of $G : \mathbf{D} \to \mathbf{C}$ induces a monad $\langle G \circ F, \eta, G(\varepsilon_{F(-)})\rangle$ on $\mathbf{C}$, where η is the unit of the adjunction and ε is the counit.

Conversely, given a monad $\langle T, \eta, \mu\rangle$ on $\mathbf{C}$ we can always find an adjunction inducing it. To this aim, denote the category of the algebras of the monad T on $\mathbf{C}$ by $\mathbf{C}^T$. Its objects are morphisms (called T-algebras) $h \in \mathbf{C}(T(A), A)$ such that $h \circ \eta_A = id_A$ and $h \circ \mu_A = h \circ T(h)$; its morphisms from a T-algebra $h \in \mathbf{C}(T(A), A)$ to a T-algebra $k \in \mathbf{C}(T(B), B)$ are morphisms f in $\mathbf{C}(A, B)$ such that $f \circ h = k \circ T(f)$. The obvious forgetful functor $G^T : \mathbf{C}^T \to \mathbf{C}$ which sends a T-algebra $h \in \mathbf{C}(T(A), A)$ to A has a left adjoint $F^T : \mathbf{C} \to \mathbf{C}^T$ which maps any object A in $\mathbf{C}$ to the free T-algebra $\mu_A \in \mathbf{C}(T(T(A)), T(A))$. The monad defined by this adjunction is trivially equal to the original monad T (see page 136, Theorem 1 in [ML71]), hence every monad is defined by its algebras.

A functor $G : \mathbf{D} \to \mathbf{C}$ is said to be *monadic* if it has a left adjoint F and the comparison functor $K : \mathbf{D} \to \mathbf{C}^T$ defined by $K(A) = G(\varepsilon_A)$ is an equivalence of categories, where T is the monad induced by $G \circ F$ and ε is the counit of the adjunction between F and G. The following version of Beck's Theorem (see page 147 in [ML71]) gives conditions on a functor G to ensure that G is monadic.

2.1.2. Proposition. *Let $G : \mathbf{D} \to \mathbf{C}$ be a functor with a left adjoint F. If $\mathbf{D}$ has all coequalizers, G preserves these coequalizers, and G reflects isomorphisms then G is monadic.* $\square$

We conclude this section by mentioning the connection between the theory of monads and universal algebra. For a more detailed account we refer to [Bir67, EM85] for the finitary case, to [Lin66] for the infinitary one, and to the comprehensive [Man76].

An *algebraic theory* $T = \langle \Omega, E \rangle$ consists of a class Ω of operators each with an associated set I denoting its arity, and a class E of identities of the form $e_l = e_r$, where e_l and e_r are expressions formed from a convenient set of variables by applying the given operators.

A *T-algebra* is a set A together with a corresponding function $\omega_A : A^I \to A$ for each operator ω of Ω of arity I, such that independently of the way we substitute elements of A for the variables, the identities of E hold in A. More formally, if e is an expression formed using a set of variables x_i for $i \in I$ by applying the given operators in Ω, the substitution of elements of A for the x_i's gives us a corresponding function $e : A^I \to A$. If A is a T-algebra and $e_l = e_r$ is an identity in E using free variables x_i for $i \in I$, then the two corresponding functions $e_l, e_r : A^I \to A$ must be equal.

A *T-homomorphism* between two T-algebras A and B is a function $f : A \to B$ such that $\omega_B \circ \Pi_{i \in I} f = f \circ \omega_A$ for each operator $\omega \in \Omega$ of arity I. The collection of all T-algebras and T-homomorphisms between them form a category T-**Alg**. There is a forgetful functor $U : T\text{-}\mathbf{Alg} \to \mathbf{Set}$ mapping every T-algebra A to its underlying set A (its action on morphisms is obvious). The following proposition [Man76, Chapter 1] summarizes some important features of the category of T-algebras.

2.1.3. Proposition. *For every algebraic theory T the category of T-algebras T-**Alg** has all small limits: they are constructed exactly as in **Set**, and the forgetful functor $U : T\text{-}\mathbf{Alg} \to \mathbf{Set}$ preserves them. Moreover, if the forgetful functor $U : T\text{-}\mathbf{Alg} \to \mathbf{Set}$ has a left adjoint, then the functor U is monadic and T-**Alg** has all small colimits.* $\square$

Let $T = \langle \Omega, E \rangle$ be an algebraic theory. A *presentation* $T\langle G \mid R \rangle$ consists of a set G (called in this context of *generators*) and a set R of pairs (called in this context *relations*) of the form $\langle e_l, e_r \rangle$, where e_l and e_r are expressions formed from generators in G by applying the given operators in Ω. A *model* for a presentation $T\langle G \mid R \rangle$ is a T-algebra A together with a function $[\![-]\!]_A : G \to A$ such that if $\langle e_l, e_r \rangle$ is a relation in R then $[\![e_l]\!]_A = [\![e_r]\!]_A$. In the latter equality, we have applied the function $[\![-]\!]_A : G \to A$ to an expression e (built up from generators in G and operators in Ω) by replacing the generators g by their interpretation $[\![g]\!]_A$, the operators ω by the corresponding operations ω_A, and evaluating the resulting function in A to give $[\![e]\!]_A \in A$.

Notice that in a presentation we make two different uses of equations: in the identities of E that are part of T equations contain variables and must hold whatever values from an algebra are substituted for the variables, and in the relations of R equations contain generators and must hold when the generators are given their particular values in a model.

A T-algebra A is *presented* by a presentation $T\langle G \mid R\rangle$ if it is a model for the presentation, and for every other model B there exists a unique morphism $h : A \to B$ in T-**Alg** such that $h([\![g]\!]_A) = [\![g]\!]_B$ for every generator $g \in G$. Clearly, the algebra presented by generators and relations if it exists is unique, up to isomorphisms in T-**Alg**. Once we know that the forgetful functor $U : T$-**Alg** $\to$ **Set** has a left adjoint F, the standard theory of congruences gives us a way to force the relations R on the free T-algebra $F(G)$. The resulting T-algebra is presented by $T\langle G \mid R\rangle$.

For a *finitary algebraic theory* T, i.e. one with a set (and not a proper class) of operators and such that each operator has finite arity, every presentation $T\langle G \mid R\rangle$ always presents a T-algebra. It can be constructed as a suitable quotient of the set of terms formed from G by applying the operators of T. This implies that the forgetful functor T-**Alg** $\to$ **Set** is monadic [Man76, Chapter 1] with left adjoint given by the assignment which takes every set G to the algebra which presents $T\langle G \mid \emptyset\rangle$.

More generally, given any monad on **Set** we can describe its algebras by operations and equations, provided that we allow infinitary theories T (ones with proper classes of operators and equations, respectively). The converse is, unfortunately, false: there are infinitary theories T for which a presentation $T\langle G \mid R\rangle$ does not present any T-algebra. Technically what goes wrong is that the collection of terms formed from G by applying the operators of Ω and satisfying the equations of E can be too big to be a valid set (i.e. it can be a proper class). Examples are the theory of complete Boolean algebras [Gai64, Hal64] and the theory of complete lattices [Hal64].

2.2 Partial orders

Partially ordered sets occur at many different places in mathematics, and their theory belongs to the fundamentals of any book on lattice theory. A classic reference on lattice theory, representative of the status of the theory in 1967, is the book of Birkhoff [Bir67]. A good introductory modern book on lattice theory and ordered structures is that of Davey and Priestley [DP90]. In recent years, partial orders on information have been successfully used in the semantics of programming languages [Sco70, SS71]. The mathematical part of this approach is called domain theory. Here the word *domain* qualifies a mathematical structure which embodies both a notion of convergence and a notion of approximation [AJ94]. A discussion on domain theory is presented in the subsection below. The reader who wishes to consult a more detailed introduction to domain theory and continuous lattices may find [Plo81a, AJ94] and [GHK$^+$80] useful references.

Let P be a set. A *preorder* $\lesssim$ on P is a binary, reflexive and transitive relation. A *partial order* on P is a binary relation $\leq$ which is reflexive, transitive and also antisymmetric. A *poset* is a set equipped with a partial order. A poset P is said to be *discrete* if the partial order coincides with the identity relation. Hence every set can be thought of as a discrete poset.

In every preordered set P we denote, for $x \in P$, by $\uparrow x$ the upper set of x, and by $\downarrow x$ the lower set of x, that is,

$$\uparrow x = \{y \in P \mid x \lesssim y\} \quad \text{and} \quad \downarrow x = \{y \in P \mid y \lesssim x\}.$$

The set $\uparrow x$ is also called the *principal filter* of X generated by x.

Let P be a poset and S be a subset of P. An element x of P is a *join* (or least upper bound) for S, written as $x = \bigvee S$, if for all $s \in S$, $s \leq x$ and if $s \leq y$ for all $s \in S$ then $x \leq y$. Since the partial order is antisymmetric, the join of S if it exists is unique. If S is a two element set $\{s, t\}$ then we write $s \vee t$ for $\bigvee S$; and if S is the empty set $\emptyset$ then we write $\perp$ for $\bigvee S$. Clearly if $\perp$ exists then it is the least element of P.

Dually, in any poset P we can define the notion of *meet* by reversing all the inequalities in the definition of the join. We write $\bigwedge S$, $s \wedge t$ and $\top$ for the meet of an arbitrary subset S of P, the binary meet and the empty meet. Notice that for every upper closed subset S of P, if $\bigwedge S$ exists in S then $S = \uparrow(\bigwedge S)$.

A function $f : P \to Q$ between two posets is said to be *monotone* if $p \leq q$ in P implies $f(p) \leq f(q)$. If the reverse implication holds then f is said to be *order reflecting*. The collection of all posets with monotone functions between them forms the category **PoSet**.

A function $f : P \to Q$ between two posets is said to be *strict* whenever $\perp \in P$ is the least element of P implies $f(\perp)$ is the least element of Q.

A function $f : P \to Q$ between two posets is said to be *finitely additive* if it preserves all finite joins of P, and *completely additive* if it preserves all joins of P. Dual notions are *finitely multiplicative* and *completely multiplicative*.

An element x of a poset P is said to be a *least fixed point* of $f : P \to P$ if $f(x) = x$, and $f(y) = y$ implies $x \leq y$ for all other $y \in P$. Dually, x is said to be a *greatest fixed point* of $f : P \to P$ if $f(x) = x$, and $f(y) = y$ implies $y \leq x$ for all other $y \in P$. The following proposition is due to Knaster [Kna28] and later reformulated by Tarski [Tar55] to assert that the set of fix-points of a monotone function f on a complete lattice forms a complete lattice (and therefore f has a least fixed point).

2.2.1. Proposition. *Let P be a poset and let $f : P \to P$ be a monotone function. If $y = \bigwedge\{x \mid f(x) \leq x\}$ exists in P, then y is the least fixed point of f. Similarly, if $z = \bigvee\{x \mid x \leq f(x)\}$ exists in P, then z is the greatest fixed point of f.* □

A poset in which every finite subset has a join is called *join-semilattice* and, analogously, a poset in which every finite subset has a meet is called *meet-semilattice*. A *lattice* is a poset in which every finite subset has both meet and join. By considering arbitrary subsets, and not just finite ones, we can define *complete join-semilattice*, *complete meet-semilattice* and *complete lattice*. A poset is a complete join-semilattice if and only if it is a complete meet-semilattice. Thus every complete semilattice is actually a complete lattice. However it is convenient to distinguish between the two concepts since a morphism of complete join-semilattices (a function preserving arbitrary joins) is not necessarily a morphism of complete lattices (a function preserving both arbitrary joins and arbitrary meets). We have thus two categories **CSLat** and **CLat** with the same objects but with different morphisms.

A lattice L is called *distributive* if

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

for all a, b and c in L. The above equation holds for a lattice if and only if so does its dual [Sc890]

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

The class of all distributive lattices together with functions preserving both finite meets and finite joins defines a category, denoted by **DLat**. A distributive lattice L is called a *Heyting algebra* if for all a and b in L there exists an element $a \to b$ such that, for all c,

$$c \leq (a \to b) \text{ if and only if } (c \wedge a) \leq b.$$

A Heyting algebra L is said to be a *Boolean algebra* if and only if for all $a \in L$,

$$(a \to \bot) \to \bot = a.$$

In this case, the element $a \to \bot$ is called the complement of a, denoted by $\neg a$.

A complete lattice L satisfying the infinite distributive law

$$a \wedge \bigvee S = \bigvee \{a \wedge s \mid s \in S\}$$

for all $a \in L$ and all subsets S of L is called a *frame*. Frames with functions preserving arbitrary joins and finite meets form a category called **Frm**. Every frame F defines a Heyting algebra by putting

$$a \to b = \bigvee \{c \in F \mid c \wedge a \leq b\}.$$

Conversely, every Heyting algebra which has a join for every subset is also a frame. However, frame morphisms do not need to preserve the $\to$ operation.

A complete lattice L is called *completely distributive* if, for all sets $\mathcal{A}$ of subsets of L,

$$\bigwedge \{\bigvee S \mid S \in \mathcal{A}\} = \bigvee \{\bigwedge f(\mathcal{A}) \mid f \in \Phi(\mathcal{A})\},$$

where $f(\mathcal{A})$ denotes the set $\{f(S) \mid S \in \mathcal{A}\}$ and $\Phi(\mathcal{A})$ is the set of all functions $f : \mathcal{A} \to \bigcup \mathcal{A}$ such that $f(S) \in S$ for all $S \in \mathcal{A}$. The above equation holds for a complete lattice if and only if so does its dual [Ran52]

$$\bigvee\{\bigwedge S \mid S \in \mathcal{A}\} = \bigwedge\{\bigvee f(\mathcal{A}) \mid f \in \Phi(\mathcal{A})\}.$$

Because of the presence of arbitrary choice functions in the statement of the above law, proofs involving complete distributivity require the axiom of choice. For example, the statement that the set $\mathcal{P}(X)$ of all subsets of a set X is a completely distributive lattice when ordered by subset (or superset) inclusion is equivalent to the axiom of choice [Col54]. Completely distributive lattices with functions preserving both arbitrary meets and arbitrary joins form a category, denoted by **CDL**. Every completely distributive lattice is a frame and every frame is a distributive lattice. Moreover, there are obvious forgetful functors **CDL** $\to$ **Frm** and **Frm** $\to$ **DLat**. Every *complete ring of sets*, that is, a set of subsets of X closed under arbitrary intersections and arbitrary unions, is a completely distributive lattice.

For a meet semilattice L, a non-empty subset $\mathcal{F}$ of L is said to be a *filter* if

(i) $\mathcal{F}$ is upper closed; i.e. $a \in \mathcal{F}$ and $a \leq b$ implies $b \in \mathcal{F}$; and
(ii) $\mathcal{F}$ is a sub-meet-semilattice; i.e. $a \in \mathcal{F}$ and $b \in \mathcal{F}$ imply $a \wedge b \in \mathcal{F}$.

The collection of all filters of a meet semilattice L is denoted by *Fil*(L). If L is also a lattice then a filter $\mathcal{F} \subseteq L$ is *prime* if for all finite subsets S of L, $\bigvee S \in \mathcal{F}$ implies there exists $s \in S \cap \mathcal{F}$. Finally, if L is a complete lattice then a filter $\mathcal{F} \subseteq L$ is *completely prime* if $\bigvee S \in \mathcal{F}$ implies there exists $s \in S \cap \mathcal{F}$. For example, for any $a \in L$ the subset $\uparrow a = \{b \in L \mid a \leq b\}$ is a filter.

For a complete lattice L, an element $p \in L$ is said to be *prime* if $p \neq \top$, and $a \wedge b \leq p$ imply $a \leq p$ or $b \leq p$. The collection of all prime elements of L is denoted by *Spec*(L). The map which sends each completely prime element $p \in L$ to $\uparrow \bigwedge(L \setminus \downarrow p)$ and the map which sends each completely prime filter $\mathcal{F}$ of L to $\bigvee(L \setminus \mathcal{F})$ form an isomorphism between the completely prime filters and the prime elements of a complete lattice L.

Directed complete partial orders

There is a special class of joins in a poset that we will consider next. A non-empty subset S of a poset P is said to be *directed* if for all s and t in S there exists $x \in S$ with both $s \leq x$ and $t \leq x$. For example, the set of elements of an ω-*chain* of a poset P forms a directed set, where an ω-chain is a countable sequence $(x_n)_n$ of element of P such that $x_n \leq x_{n+1}$ for all $n \geq 0$.

We say that P is a *directed complete partial order* (dcpo) if $\bigvee S$ exists for every directed subset S of P. A dcpo P with a least element $\bot$ is called a *complete partial order* (cpo). If a poset has finite joins and directed joins then it has arbitrary joins.

An element b of a dcpo P is *compact* if for every directed subset S of P, $b \leq \bigvee S$ implies $b \leq s$ for some $s \in S$. The set of all compact elements of P is denoted by $\mathcal{K}(P)$. In any dcpo, the join of finitely many compact elements, if it exists, is again a compact element. A dcpo P is said to be *algebraic* if every $x \in P$ is the join of the directed set of compact elements below it, that is, $x = \bigvee\{b \in \mathcal{K}(P) \mid b \leq x\}$. A dcpo P is ω-algebraic if it is algebraic and the set $\mathcal{K}(P)$ is countable. When a dcpo is ω-algebraic we do not need to consider general directed joins, but only joins of ω-chains are sufficient.

A monotone function $f : P \to Q$ between two dcpo's is said to be *continuous* (or Scott continuous) if it preserves all directed joins. The collection of all dcpo's with continuous functions forms a category, denoted by **DCPO**. The full sub-category of **DCPO** whose objects are complete partial orders is denoted by **CPO**, whereas the full sub-category of **DCPO** whose objects are algebraic dcpo's is denoted by **AlgPos**. The forgetful functor **CPO** $\to$ **DCPO** has a left adjoint $(-)_\perp$ mapping every dcpo P to the *lift* $P_\perp$, that is, the poset P with a new least element adjoined. If P is a set (e.g. discrete dcpo), then the lift $P_\perp$ is said to be a *flat cpo*. A flat cpo is algebraic and every element is compact. Dually to the lift, for a poset P we denote by $P^\top$ the poset P with a new top element adjoined.

The forgetful functor $U : \mathbf{AlgPos} \to \mathbf{PoSet}$ has a left adjoint $\mathit{Idl}(-)$ defined by the map which assigns to a poset P the algebraic dcpo $\mathit{Idl}(P)$ of *directed ideals* of P (i.e. the directed lower subsets of P) ordered by subset inclusion. Moreover, $\mathit{Idl}(P)$ is also a cpo if and only if P has a least element. The algebraic dcpo $\mathit{Idl}(P)$ is often referred to as the *ideal completion* of the poset P. More generally, for a preordered set P, the poset $\mathit{Idl}(P)$ of all directed ideals of P ordered by subset inclusion forms an algebraic dcpo with compact elements $\uparrow x$ for $x \in P$.

Let P be an algebraic cpo. There are three standard preorders defined on subsets X and Y of P:

- the *Hoare* preorder, defined as $X \lesssim_H Y$ if $\forall x \in X \exists y \in Y : x \leq y$;
- the *Smyth* preorder, defined as $X \lesssim_S Y$ if $\forall y \in Y \exists x \in X : x \leq y$; and
- the *Egli-Milner* preorder, defined as $X \lesssim_E Y$ if $X \lesssim_H Y$ and $X \lesssim_S Y$.

Powerdomains can be constructed from these preorders by ideal completion: the *Hoare powerdomain* $\mathcal{H}(P)$, the *Smyth powerdomain* $\mathcal{S}(P)$ and the *Plotkin powerdomain* $\mathcal{E}(P)$ of an algebraic cpo P are defined as the ideal completion of $(\mathcal{P}_{\mathit{fin}}(\mathcal{K}(P)), \lesssim_H)$, $(\mathcal{P}_{\mathit{fin}}(\mathcal{K}(P)), \lesssim_S)$, and $(\mathcal{P}_{\mathit{fin}}(\mathcal{K}(P)), \lesssim_E)$, respectively, where $\mathcal{P}_{\mathit{fin}}(\mathcal{K}(P))$ consists of all finite, non-empty sets of compact elements of P.

Let P and Q be two algebraic cpo's. The *coalesced sum* $P \oplus Q$ is defined as the disjoint union of P and Q with bottom elements identified, whereas the *separated sum* $P + Q$ is the disjoint union of P and Q with a new bottom element $\bot$ adjoined. The *product* $P \times Q$ is defined as the Cartesian product of the underlying sets ordered componentwise. All these constructions can be generalized to arbitrary sets of algebraic cpo's. For example, the separated sum, $\sum_I P_i$ is the algebraic cpo obtained by the disjoint union of all the algebraic cpo's P_i with a new bottom element $\bot$ adjoined.

Let P be a cpo. A *minimal upper bound* x of a subset S of P is an upper bound of S (that is, $s \leq x$ for all $s \in S$) such that for all $y \in P$,

$$\forall s \in S \colon s \leq y \;\&\; y \leq x \;\Rightarrow\; x \leq y.$$

In other words, x is a minimal upper bound of S if x is above every element in S and there is no other element y above every element in S but below x. Note that, in contrast with a least upper bound, a minimal upper bound need not to be unique. The set $mub(S)$ denotes the set of minimal upper bounds of S, and the set $mub^\star(S)$ is the smallest set $Y \subseteq P$ such that $S \subseteq Y$ and if $X \subseteq Y$ then $mub(X) \subseteq Y$, that is, $mub^\star(S)$ is the least set containing S and closed under $mub(-)$. An algebraic cpo P is said to be an *SFP-domain* if for every finite subset S of compact elements $\mathcal{K}(P)$:

(i) if y is an upper bound of S then $x \leq y$ for some $x \in mub(S)$, and
(ii) the set $mub^\star(S)$ is finite.

Alternatively, one can define SFP-domains as those algebraic cpo's which arise both as limits and as colimits in **CPO** of countable sequences (via embedding-projection pairs) of finite posets [Plo76]. The full sub-category of **AlgPos** whose objects are SFP-domains is denoted by **SFP**. The justification for studying this category is that it is the largest Cartesian closed category with ω-algebraic cpo's as objects and continuous functions as morphisms [Smy83a]. The only fact about **SFP** which we will need in the sequel is that it is closed under the following constructors: lift, coalesced sum, countable separated sum, and Plotkin powerdomain. Moreover **SFP** admits recursive definitions of SFP-domains by using the above constructors [Plo81a, Chapter 5, Theorem 1].

Fixed points

In Proposition 2.2.1 sufficient conditions are given to guarantee the existence of least and greatest fixed points of a monotone function on a poset. Next we recall other characterizations of least and greatest fixed points of monotone functions. For an overview of fixed point theorems we refer to [LNS82].

Let P be a poset and let $f : P \to P$ be a function. For any ordinal α define $f^{\langle\alpha\rangle}$ and $f^{[\alpha]}$ as the following elements of P (if they exists):

$$f^{\langle\alpha\rangle} = f(\bigvee\{f^{\langle\beta\rangle} \mid \beta < \alpha\}) \quad \text{and} \quad f^{[\alpha]} = f(\bigwedge\{f^{[\beta]} \mid \beta < \alpha\}). \tag{2.1}$$

In general they do not need to exist, since $\bigvee\{f^{\langle\beta\rangle} \mid \beta < \alpha\}$ and $\bigwedge\{f^{[\beta]} \mid \beta < \alpha\}$ may not exist. Notice that for $\alpha = 0$, $f^{\langle\alpha\rangle} = f(\bot)$ when the least element

$\perp \in P$ exists since in this case the join over an empty index set is the bottom element. Similarly, for $\alpha = 0$, $f^{[\alpha]} = f(\top)$ when the top element exists. The following proposition, originally formulated by Kleene [Kle52] in a different context, characterizes the least fixed point as a directed join in contrast with Proposition 2.2.1 where the least fixed point is characterized as an infinite meet.

2.2.2. Proposition. *Let P be a complete partial order. If $f : P \to P$ is a continuous function then $f^{\langle\alpha\rangle}$ exists in P for every ordinal α, and $f^{\langle\omega_0\rangle}$ is the least fixed point of f (here ω_0 is the first limit ordinal).* □

Hitchcock and Park have extended the above proposition by weakening the constraint on f from continuous to monotone [HP72].

2.2.3. Proposition. *Let P be a complete partial order. If $f : P \to P$ is a monotone function then $f^{\langle\alpha\rangle}$ exists in P for every ordinal α, and there exists an ordinal β such that $f^{\langle\beta\rangle} = f^{\langle\alpha\rangle}$ whenever $\beta \leq \alpha$. The latter implies that $f^{\langle\beta\rangle}$ is the least fixed point of f.* □

The dual of the above proposition holds as well. To guarantee the existence of certain meets we rephrase it for complete lattices.

2.2.4. Proposition. *Let P be a complete lattice. If $f : P \to P$ is a monotone function then $f^{[\alpha]}$ exists in P for every ordinal α, and there exists an ordinal β such that $f^{[\beta]} = f^{[\alpha]}$ whenever $\beta \leq \alpha$. The latter implies that $f^{[\beta]}$ is the greatest fixed point of f.* □

Under certain circumstances, the least fixed point of a function on a cpo can be enough to guarantee the existence of the least fixed point of another function on a poset which is not necessarily complete [AP86].

2.2.5. Proposition. *Let P be a cpo and let Q be a poset such that there is a strict and continuous function $h : P \to Q$. If $x \in P$ is the least fixed point of a monotone function $f : P \to P$ and $g : Q \to Q$ is another monotone function such that the following diagram commutes*

$$\begin{tikzcd} P \arrow[r, "f"] \arrow[d, "h"'] \arrow[dr, phantom, "*"] & P \arrow[d, "h"] \\ Q \arrow[r, "g"'] & Q \end{tikzcd}$$

then the least fixed point of g exists and equals $h(x)$. □

Several generalizations and applications of the above proposition (often called the transfer lemma) can be found in [Mey85].

2.3 Metric spaces

We conclude this chapter with a section on some basic notions related to metric spaces. The results in this section will play a key role only in the second part of this thesis. Like partially ordered sets, metric spaces are fundamental structures in mathematics, especially in topology. For details we refer the reader to Engelking's standard work [Eng89] and Dugundji's classical book [Dug66]. We use metric spaces as a mathematical structure for semantics of programming languages, following the work of Arnold and Nivat [AN80]. For a comprehensive survey of the use of metric spaces in the semantics of a large variety of programming notions, we refer the reader to [BV96].

A (one-bounded) *metric space* consists of a set X together with a function $d_X : X \times X \to [0,1]$, called metric or distance, satisfying, for x, y and z in X,

$$\begin{array}{llll} \text{(i)} & d_X(x,x) = 0 & \text{(ii)} & d_X(x,z) \leq d_X(x,y) + d_X(y,z) \\ \text{(iii)} & d_X(x,y) = d_X(y,x) & \text{(iv)} & d_X(x,y) = d_X(y,x) = 0 \ \Rightarrow\ x = y \end{array}$$

A set X with a function $d_X : X \times X \to [0,1]$ satisfying only (i), (ii), and (iii) is called a (one-bounded) *pseudo metric space*. A *quasi metric space* is a set X with a function $d_X : X \times X \to [0,1]$ satisfying axioms (i), (ii), and (iv). We shall usually write X instead of (X, d_X) and denote the metric of X by d_X. A metric space X with a distance function which satisfies, for all x, y and z in X,

$$d_X(x,z) \leq \max\{d_X(x,y), d_X(y,z)\}$$

is said to be an *ultra-metric space*. Clearly the above axiom implies axiom (ii).

A countable sequence of points $(x_n)_n$ of a metric space X is said to *converge* to an element $x \in X$ if

$$\forall \epsilon > 0 \exists k \geq 0 \forall n \geq k \colon d_X(x_n, x) \leq \epsilon.$$

Every sequence converges to at most one point which, if it exists, is said to be the *limit* of the sequence. It is denoted by $\lim_n x_n$. A countable sequence of points $(x_n)_n$ of a metric space X is said to be *Cauchy* if

$$\forall \epsilon > 0 \exists k \geq 0 \forall m, n \geq k \colon d_X(x_m, x_n) \leq \epsilon.$$

As can be easily seen, every convergent sequence is Cauchy. A metric space is called *complete* if every Cauchy sequence converges to some point in X.

The simplest example of a complete metric space is the following. A metric space X is called *discrete* if

$$\forall x \in X \exists \epsilon > 0 \forall y \in X \colon d_X(x,y) \ < \ \epsilon \ \Rightarrow \ x = y.$$

Every discrete metric space is complete since it has no non-trivial Cauchy sequences. A set X can be seen as a discrete metric space if endowed with a distance function which assigns to $x, y \in X$, distance 1 if $x \neq y$ and distance 0 otherwise.

Let X and Y be two metric spaces. A function $f : X \to Y$ is said to be *non-expansive* if $d_Y(f(x_1), f(x_2)) \leq d_X(x_1, x_2)$ for all $x_1, x_2 \in X$. The set of all non-expansive functions from X to Y is denoted by $X \xrightarrow{1} Y$. Complete metric spaces together with non-expansive maps form a category, denoted by **CMS**.

Of special interest in the study of metric spaces are *contracting* functions, that is, functions $f : X \to Y$ such that

$$\exists \epsilon < 1 \forall x, y \in X \colon d_Y(f(x_1), f(x_2)) \leq \epsilon \cdot d_X(x_1, x_2).$$

The following proposition is known as the Banach fixed point theorem [Ban22].

2.3.1. Proposition. *If X is a complete metric space and $f : X \to X$ is a contracting function then f has a unique fixed point x such that, for every $y \in X$, $x = \lim_n y_n$, where $(y_n)_n$ is the Cauchy sequence defined inductively by $y_0 = y$ and $y_{n+1} = f(y_n)$.* □

Next we define some of the constructors on metric spaces. For all $\epsilon \leq 1$ and metric space X, define the metric space $\epsilon \cdot X$ as the set X with distance function, for all x_1 and x_2 in X,

$$d_{\epsilon \cdot X}(x_1, x_2) = \epsilon \cdot d_X(x_1, x_2).$$

The *product* $X \times Y$ of two metric spaces X and Y is defined as the Cartesian product of their underlying sets together with distance, for $\langle x_1, y_1 \rangle$ and $\langle x_2, y_2 \rangle$ in $X \times Y$,

$$d_{X \times Y}(\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle) = \max\{d_X(x_1, x_2), d_Y(y_1, y_2)\}.$$

The *exponent* of X and Y is defined by

$$Y^X = \{f : X \to Y \mid f \text{ is non-expansive }\},$$

with distance, for f and g in Y^X,

$$d_{Y^X}(f, g) = \sup\{d_Y(f(x), g(x)) \mid x \in X\}.$$

Notice that if Y is a set endowed with the discrete metric then every function from Y to X is non-expansive. The *disjoint union* $X + Y$ of two metric spaces X and Y is defined by taking the disjoint union of their underlying sets with distance, for z_1 and z_2 in $X + Y$,

$$d_{X+Y}(z_1, z_2) = \begin{cases} d_X(z_1, z_2) & \text{if } z_1 \in X \text{ and } z_2 \in X \\ d_Y(z_1, z_2) & \text{if } z_1 \in Y \text{ and } z_2 \in Y \\ 1 & \text{otherwise.} \end{cases}$$

If both X and Y are complete metric spaces then also $X \times Y$, Y^X and $X + Y$ are complete metric spaces.

The *Hausdorff distance* between two subsets A and B of a metric space X is defined by

$$\begin{aligned} d_{\mathcal{P}(X)}(A, B) \quad = \max\{ \quad & \sup\{\inf\{d_X(a, b) \mid b \in B\} \mid a \in A\}, \\ & \sup\{\inf\{d_X(a, b) \mid a \in A\} \mid b \in B\} \} \end{aligned}$$

with the convention that $\inf \emptyset = 1$ and $\sup \emptyset = 0$. In general $(\mathcal{P}(X), d_{\mathcal{P}(X)})$ is a pseudo metric space: different subsets of X can have null distance. In order to turn sets of subsets of a metric space into a metric space, we need the following notions. A subset S of a metric space X is said to be *closed* if the limit of every convergent sequence in S is an element of S. For example, the set X itself is closed as well as the empty set. Also, every singleton set $\{x\}$ is closed. In general, every subset S can be extended to a closed set

$$cl(S) = \{\lim_n x_n \mid (x_n)_n \text{ is a convergent sequence in } S\}.$$

Clearly, $cl(S)$ is the smallest closed set containing S. Notice that if S is a closed subset then $cl(S) = S$. A subset S of a metric space X is *compact* if for every sequence in S there exists a sub-sequence converging to some element in S. Every compact set is closed, and every finite set is compact. A metric space X is compact if the set X is compact. It follows that every compact metric space is complete.

Both the collection of compact subsets of a metric space X, denoted by $\mathcal{P}_{co}(X)$, and the collection of closed subsets of X, denoted by $\mathcal{P}_{cl}(X)$, are metric spaces when taken with the Hausdorff distance. Moreover, if X is a complete metric space then $\mathcal{P}_{co}(X)$ is a complete metric space [Hah48], and also $\mathcal{P}_{cl}(X)$ is a complete metric space [Kur56]. We refer to them as the *compact* and *closed powerdomain* of X, respectively. Below we give two properties of the Hausdorff distance which will be useful later.

2.3.2. Proposition. *Let X be a metric space. For all $\delta \geq 0$ and subsets A and B of X, $d_{\mathcal{P}(X)}(A, B) \leq \delta$ if and only if for all $\epsilon > 0$,*

$$\forall a \in A \exists b \in B \colon d_X(a, b) \leq \delta + \epsilon \text{ and } \forall b \in B \exists a \in A \colon d_X(a, b) \leq \delta + \epsilon.$$

$\square$

While the above proposition is standard the following one seems to be new.

2.3.3. Proposition. *Let X be a metric space and V and W be two sets of subsets of X such that for all $C \subseteq X$, if $\bigcap V \subseteq C$ then $C \in V$, and if $\bigcap W \subseteq C$ then $C \in W$. Then*

$$d_{\mathcal{P}(\mathcal{P}(X))}(V, W) = d_{\mathcal{P}(X)}(\bigcap V, \bigcap W).$$

Proof: Put $V_0 = \bigcap V$ and $W_0 = \bigcap W$. Let also $\delta = d_{\mathcal{P}(\mathcal{P}(X))}(V, W)$ and $\delta_0 = d_{\mathcal{P}(X)}(V_0, W_0)$. We claim

$$\begin{aligned} &\forall X \in V \exists Y \in W : d_{\mathcal{P}(X)}(X, Y) \leq \delta_0 + \epsilon \quad \text{and} \\ &\forall Y \in W \exists X \in V : d_{\mathcal{P}(X)}(Y, X) \leq \delta_0 + \epsilon \end{aligned} \tag{2.2}$$

for an arbitrary $\epsilon > 0$. From the above claim $\delta \leq \delta_0$ follows by Proposition 2.3.2. Next we prove the claim. Choose some $\epsilon > 0$ and $X \in V$. Put $Y = W_0 \cup X$. By the closure property of W, since $W_0 \subseteq Y$, also $Y \in W$. Since $X \subseteq Y$ we have that for all $x \in X$ we can find $y \in Y$ such that $d_X(x, y) = 0$. On the other hand, for all $y \in Y$, either $y \in W_0$ or $y \in X$ by definition. So, either $d_X(y, x) \leq \delta_0 + \epsilon$ for some $x \in V_0 \subseteq X$ (since $d_{\mathcal{P}(X)}(V_0, W_0) \leq \delta_0 + \epsilon$ and $V_0 \subseteq X \in V$) or $d_X(y, x) = 0 \leq \delta_0 + \epsilon$ from $x = y \in X$. Hence (2.2) follows by Proposition 2.3.2. Symmetrically we derive

$$\forall Y \in W \exists X \in V : d_{\mathcal{P}(X)}(Y, X) \leq \delta_0 + \epsilon$$

from which our claim follows and we conclude $\delta = d_{\mathcal{P}(\mathcal{P}(X))}(V, W) \leq \delta_0$.

In order to show the converse, i.e. $\delta_0 \leq \delta$, we establish

$$\begin{aligned} &\forall x \in V_0 \exists y \in W_0 : d_X(x, y) \leq \delta + \epsilon \quad \text{and} \\ &\forall y \in W_0 \exists x \in V_0 : d_X(y, x) \leq \delta + \epsilon, \end{aligned} \tag{2.3}$$

for all $\epsilon > 0$. Since $d_{\mathcal{P}(\mathcal{P}(X))}(V, W) = \delta$, $V_0 \in V$, and $W_0 \in W$ we can find $Y \in W$ and $X \in V$ by Proposition 2.3.2 such that

$$d_{\mathcal{P}(X)}(V_0, Y) \leq \delta + \epsilon \quad \text{and} \quad d_{\mathcal{P}(X)}(X, W_0) \leq \delta + \epsilon.$$

From this we obtain the result (2.3) and its symmetric version, for $V_0 \subseteq Y$ and $W_0 \subseteq X$, respectively. Therefore $\delta_0 = d_{\mathcal{P}(X)}(V_0, W_0) \leq \delta$. □

In [AR89, RT93], generalizing the results of [BZ82], a method has been developed to justify recursive definitions of complete metric spaces as solutions of domain equations of the form $X \cong F(X)$, where $F : \mathbf{CMS} \to \mathbf{CMS}$ is a functor. A solution for the domain equation $X \cong F(X)$ exists and it is unique (up to isometries) if the functor F is *locally contracting*, that is, for every two complete metric spaces X and Y, the mapping

$$F_{X,Y} : Y^X \to F(Y)^{F(X)}$$

is contractive, where $F_{X,Y}(f) = F(f)$ for every non-expansive $f : X \to Y$. If, for all objects X, Y, $F_{X,Y}$ is non-expansive then the functor is called *locally non-expansive*. Composition of a locally non-expansive functor with a locally contractive one gives a locally contractive functor.

For example, the constructors $\mathcal{P}_{co}(-)$ and $\mathcal{P}_{cl}(-)$ can be extended to functors from **CMS** to **CMS** which are locally non-expansive, while the constructor $\frac{1}{2} \cdot (-)$ can be extended to a functor from **CMS** to **CMS** which is locally contractive. Also, for a fixed set S (understood as a discrete metric space), the constructors $S \times -$, $S^{(-)}$, and $S + (-)$ can be extended to functors from **CMS** to **CMS** which are locally non-expansive [AR89, RT93].

Part I

Basic dualities

Chapter 3

The weakest precondition calculus

The role of a *sequential program* is to produce a final result at the end of a terminating computation. Computations may possibly be non-deterministic and also fail to terminate. The main characteristic of sequential programs is that no interaction with its environment is possible. Programs written in classical programming languages like Pascal are examples of sequential programs. Different semantics for this type of programs (and their relationships) are our main interest in this first part of the thesis.

The *semantics* of a programming language $\mathcal{L}$ is a function which assigns to each program in $\mathcal{L}$ its meaning, that is, an element of a domain of meanings chosen for modeling the computations specified by the program. There are different approaches to the definitions of the semantic function and of the semantic domain.

The *operational* approach is intended to specify the meaning of a program in terms of the steps performed by an abstract machine when executing it. Formally, a transition relation on the configurations of an abstract machine is specified [HP79, Plo81b]: a transition from a configuration to another one represents one atomic step of a computation. Then the semantic function is defined in terms of the transition relation. A computation of a program may fail to terminate if it contains an infinite transition sequence. A computation deadlocks if there is a configuration reached by the computation from which no transition is possible. The operational view of a program on the one hand corresponds often to its intuitive meaning, but, on the other hand, it is not always abstract enough to be computationally useful since it might require a rather detailed and intricate analysis.

Another approach to semantics is the *denotational* one [SS71, MS76, Sto77, Gor79]: first provide an appropriate semantic domain according to the principle that program constructs denote values, and then define the semantic function in such way that the meaning of each syntactic construction of a program is given in terms of the meanings of its constituent parts. In particular fixed point techniques are needed to deal with recursion. For sequential programs this results in the relation between input and output values. Thus the most simple abstract denotational domain for sequential programs is that of all functions from a starting state space (the set of all admissible inputs values) to a final state space (the set of all possible output values). The semantics of a program is a function, which we call *state transformer.* In order to take into account non-termination of programs it is a natural step to consider state transformers employing complete partial orders with a bottom element—a fictitious state representing non-termination. Within this framework, non-determinism can be handled using powerdomains. The state transformer model reflects closely the operational view of a program, but abstracts from the intermediate configurations.

The *axiomatic* approach has different aims from the operational and the denotational ones: proving program correctness, analyzing program properties, and synthesizing correct programs from formal specifications [Dij76, Bac78, DS90, BW90b]. Informally, a sequential program is correct if it satisfies the intended relation between input values and output value. Program correctness is expressed by statements of the form $\{P\}S\{Q\}$, where S is a sequential program, P is a predicate on the set of input values (precondition) and Q is a predicate on the set of output values (postcondition) [Hoa69]. The precondition P describes the initial input values in which the program S is started, and the postcondition Q describes the set of the desirable output values. More abstractly, correctness statements can be defined with the weakest precondition and the weakest liberal precondition: programs can be identified with functions, called *predicate transformers*, from predicates on the set of all possible output values to predicates on the set of all admissible input values. The weakest (liberal) precondition calculus was introduced by Dijkstra [Dij76] as a mathematical tool for reasoning about the partial and total correctness of programs, and it has been further developed in [Gri81, DS90, Hes92b]. This predicate transformer model is called axiomatic because it relies only on algebraic properties of predicates (described for example in [Hal62]).

In this chapter we start by introducing the syntax of a sequential language. Then we define three different state transformer semantic domains. Accordingly, three state transformer semantics for our language are introduced and related. We define two predicate transformer semantics, one by taking into account the possibility of non-termination, and another one by not doing so. State transformer semantics and predicate transformer semantics will be proved to be equivalent. We conclude the chapter with a formal treatment of a backtrack operator in the weakest precondition calculus.

3.1 The sequential language $\mathcal{L}_0$

We begin by introducing a simple sequential language $\mathcal{L}_0$ which is inspired by Dijkstra's language of guarded commands [Dij76]. The language constructors are assignment, conditional, non-deterministic choice and sequential composition. The language allows for recursion by means of procedure variables. Dijkstra's guarded commands, conditionals and recursive combinators can be expressed in terms of the basic constructors of $\mathcal{L}_0$.

All the constructors of the language are well-known. The free occurrence of guards as a conditional is already present in Hoare [Hoa78]. The non-deterministic choice is studied, for example, by De Bakker in [Bak80]. More generally, the language $\mathcal{L}_0$ is a slight variation of Hesselink's calculus of commands [Hes88].

To define the language, we need as basic blocks the sets $(v \in) \mathit{IVar}$ of (individual) variables, $(e \in) \mathit{Exp}$ of expressions, $(b \in) \mathit{BExp}$ of Boolean expressions, and $(x \in) \mathit{PVar}$ of procedure variables, respectively. For a fixed set of values Val, the set of states $(s, t \in) \mathtt{St}$ is given by $\mathtt{St} = \mathit{IVar} \to \mathit{Val}$. As usual, for every state $s \in \mathtt{St}$, individual variable $v \in \mathit{IVar}$ and value $z \in \mathit{Val}$, $s[z/v]$ denotes the state which evaluates to $s(v')$ for every $v' \neq v$ and evaluates to z otherwise. Also, we postulate valuations

$$\mathcal{E}_{\mathcal{V}} : \mathit{Exp} \to (\mathtt{St} \to \mathit{Val}) \quad \text{and} \quad \mathcal{B}_{\mathcal{V}} : \mathit{BExp} \to \mathcal{P}(\mathtt{St}).$$

These functions provide, in a rather abstract way, the semantics of expressions and Boolean expressions. Clearly $\mathcal{E}_{\mathcal{V}}(e)(s) = z$ means that the expression e in a state s has value z, and, similarly, $s \in \mathcal{B}_{\mathcal{V}}(b)$ means that the Boolean expression b is true in a state s. Notice that for simplicity we assume that the evaluation of an expression and of a Boolean expression is deterministic and always terminates.

The language below has assignment ':=', conditional '$b\!\to$', sequential composition ';', choice '$\Box$', and recursion through procedure variables. Its syntax is defined as follows.

3.1.1. Definition. (i) The set $(S \in) \mathit{Stat}_0$ of *statements* is given by

$$S \ ::= \ v := e \mid b\!\to \mid x \mid S \,;\, S \mid S \Box S\,.$$

(ii) The set $(d \in) \mathit{Decl}_0$ of *declarations* is defined by $\mathit{Decl}_0 = \mathit{PVar} \to \mathit{Stat}_0$.

(iii) The language $\mathcal{L}_0$ is given by $\mathit{Decl}_0 \times \mathit{Stat}_0$.

The computational intuition behind assignments is as usual. The conditional '$b\!\to$' deadlocks in a state in which the Boolean expression 'b' does not evaluate to true and acts as a skip otherwise. We assume deadlock is not signaled. The sequential composition executes the first component and then it executes the second component. The choice executes one of its components (the choice as to which component is taken may be made by an implementation or, for non-sequential languages, may be forced by some external factor). The intended meaning of a procedure variable is body replacement.

We do not give an operational semantics for the language $\mathcal{L}_0$, since in this thesis we will not deal with the connection between the operational and denotational semantics (which, of course, is an important topic [Plo81a, BR92, BV96]). We concentrate on state transformer and predicate transformer models, and we shall rely on our computational intuition when formulating the semantic function.

3.2 State transformer models

In the state transformer approach programs are denoted by functions that relate an input state s to the outcomes of all the computations of the program when started in s. There are two important aspects to be considered. There may be input states s for which the program deadlocks or fails to terminate. In the first case, since no outcome is present, the input s is related to the empty set. This is in accordance with the fact that if a program at input s can either deadlock or produce some outputs then there is no reason to signal deadlock as a result of a computation. In the second case we need to introduce a special value —usually $\perp$— to which a non-terminating computation is mapped.

Some difficulties arise when we consider non-deterministic programs. Suppose we have a procedure variable $x \in PVar$ declared as $d(x) = v := 0 \,;\, x$, and let us consider the programs

- $P_1 = \langle d, v \ := \ 1\rangle$
- $P_2 = \langle d, x\rangle$
- $P_3 = \langle d, v \ := \ 1 \,\square\, x\rangle$.

While program P_1 always terminates when activated, an execution of the program P_2 gets stuck in a loop. An execution of the program P_3 consists of either executing the program P_1 or the program P_2. Which of these three programs should be considered equivalent by a state transformer semantics?

One view is to consider equivalent those programs which have computations that may fail to terminate since nothing can be guaranteed for them. Hence the program P_3 should be identified with the program P_2 and it should differ from the program P_1.

Another view is to identify those programs that have the same sets of outcomes, if any. Then the program P_1 should be identified with the program P_3, and both should be different from the program P_2.

Finally, another view is to consider what actually happens: all three programs are different. Below we give three state transformer domains corresponding to these three views.

Smyth state transformers

Let X be the set of inputs and Y be the set of all possible outcomes of a class of programs we consider. Computations that are possibly non-terminating are identified (since nothing can be guaranteed of any of them) and mapped to $Y_\perp = Y \cup \{\perp\}$. Computations that deadlock are mapped to the empty set.

3.2.1. Definition. The set of *Smyth state transformers* from a set X to a set Y is defined by

$$ST_S(X, Y) \quad = \quad X \to (\mathcal{P}(Y) \cup \{Y_\perp\}).$$

In general, Smyth state transformers are ordered by the pointwise extension of the superset order, that is, for $\sigma, \tau \in ST_S(X, Y)$

$$\sigma \leq \tau \quad \text{if and only if} \quad \forall x \in X \colon \sigma(x) \supseteq \tau(x).$$

The above order can be justified as follows: the smaller the set of outcomes of a program the more can be guaranteed of it. Smyth state transformers form a poset with a least element given by the function mapping every $x \in X$ to $Y_\perp$ (corresponding to the program which always fails to terminate, and for which nothing at all can be guaranteed).

Not all Smyth state transformers are 'reasonable' denotations of programs. In particular, we may wish to consider only programs which are finitely non-deterministic:

$$ST_S^{\mathit{fin}}(X, Y) \quad = \quad X \to (\mathcal{P}_{\mathit{fin}}(Y) \cup \{Y_\perp\}),$$

where $\mathcal{P}_{\mathit{fin}}(Y)$ consists of the finite subsets of Y.

3.2.2. Lemma. *For every set X and Y, both $ST_S(X, Y)$ and $ST_S^{\mathit{fin}}(X, Y)$ are complete partial orders.*

Proof: Since the function $\lambda x \in X. Y_\perp$ is in both $ST_S(X, Y)$ and $ST_S^{\mathit{fin}}(X, Y)$, it is their least element. Assume now $\mathcal{V}$ is a directed set of functions in $ST_S(X, Y)$. It is easy to see that

$$\lambda x \in X. \bigcap\{\sigma(x) \mid \sigma \in \mathcal{V}\} \tag{3.1}$$

is the least upper bound of $\mathcal{V}$ in $ST_S(X, Y)$. If every $\sigma \in \mathcal{V}$ is in $ST_S^{\mathit{fin}}(X, Y)$ then $\sigma(x)$ is either a finite set or $\{Y_\perp\}$. Thus also

$$\bigcap\{\sigma(x) \mid \sigma \in \mathcal{V}\}$$

is a finite set or $\{Y_\perp\}$ for every $x \in X$. It follows that (3.1) is the least upper bound of $\mathcal{V}$ also in $ST_S^{\mathit{fin}}(X, Y)$. $\square$

An alternative way to prove that $ST^{fin}_S(X, Y)$ is a complete partial order is to define it as the set of all functions from X to $\mathcal{S}(Y_\perp)^\top$, the Smyth powerdomain with emptyset (added as a top element) of the flat cpo $Y_\perp$.

There are two basic operators for Smyth state transformers which can be used as the semantical counterpart of the syntactical operators of $\mathcal{L}_0$.

3.2.3. Definition. Let X, Y and Z be three sets. Define, for every $x \in X$, the *union function* $\Box : ST_S(X, Y) \times ST_S(X, Y) \to ST_S(X, Y)$ by

$$(\sigma_1 \Box \sigma_2)(x) = \sigma_1(x) \cup \sigma_2(x),$$

and the *composition function* $; : ST_S(X, Y) \times ST_S(Y, Z) \to ST_S(X, Z)$ by

$$(\sigma_1 \,;\, \sigma_2)(x) \;=\; \begin{cases} Y_\perp & \text{if } \perp \in \sigma_1(x) \text{ or} \\ & \exists y \in \sigma_1(x)\colon \perp \in \sigma_2(y) \\ \bigcup\{\sigma_2(y) \mid y \in \sigma_1(x)\} & \text{otherwise.} \end{cases}$$

These functions are monotone in both their arguments. Moreover, if σ_1 and σ_2 are in $ST^{fin}_S(X, Y)$, then also $\sigma_1 \Box \sigma_2$ is in $ST^{fin}_S(X, Y)$. Similarly, because the finite union of finite sets is a finite set, if $\sigma_1 \in ST^{fin}_S(X, Y)$ and $\sigma_2 \in ST^{fin}_S(Y, Z)$ then $\sigma_1 \,;\, \sigma_2 \in ST^{fin}_S(X, Z)$.

Once we have defined the semantical operators which will denote the syntactic operators ';' and '$\Box$' of the language $\mathcal{L}_0$, we have almost all ingredients to define a state transformer semantics for $\mathcal{L}_0$ using $ST_S(\mathtt{St}, \mathtt{St})$ as semantic domain: we have only to define the semantics for the atomic commands '$v := e$' and '$b\!\rightarrow$', and for the procedure variables 'x'.

3.2.4. Definition. The semantic function $St_S[\![\cdot]\!]$ is defined as the least function in $\mathcal{L}_0 \to ST_S(\mathtt{St}, \mathtt{St})$ such that, for all $s \in S$,

$$\begin{array}{lcl} St_S[\![\langle d, v := e\rangle]\!](s) & = & \{s[\mathcal{E}_\mathcal{V}(e)(s)/v]\}, \\ St_S[\![\langle d, b\!\rightarrow\rangle]\!](s) & = & \begin{cases} \{s\} & \text{if } s \in \mathcal{B}_\mathcal{V}(b) \\ \emptyset & \text{otherwise,} \end{cases} \\ St_S[\![\langle d, x\rangle]\!](s) & = & St_S[\![\langle d, d(x)\rangle]\!](s), \\ St_S[\![\langle d, S_1 \,;\, S_2\rangle]\!](s) & = & (St_S[\![\langle d, S_1\rangle]\!] \,;\, St_S[\![\langle d, S_2\rangle]\!])(s), \\ St_S[\![\langle d, S_1 \Box S_2\rangle]\!](s) & = & (St_S[\![\langle d, S_1\rangle]\!] \Box St_S[\![\langle d, S_2\rangle]\!])(s). \end{array}$$

The well-definedness of the above semantics can be justified as follows. The semantics $St_S[\![\cdot]\!]$ can be obtained as the least fixed point of a higher order transformation.

3.2.5. Lemma. *Let* $F \in Sem_S = \mathcal{L}_0 \to ST_S(\mathtt{St}, \mathtt{St})$ *and define* $\Psi_S : Sem_S \to Sem_S$ *inductively, for all* $s \in \mathtt{St}$, *by*

$$\begin{array}{lcl}
\Psi_S(F)(\langle d, v := e\rangle)(s) & = & \{s[\mathcal{E}_\mathcal{V}(e)(s)/v]\}, \\
\Psi_S(F)(\langle d, b{\rightarrow}\rangle)(s) & = & \left\{ \begin{array}{ll} \{s\} & \textit{if } s \in \mathcal{B}_\mathcal{V}(b) \\ \emptyset & \textit{otherwise} \end{array} \right. \\
\Psi_S(F)(\langle d, x\rangle)(s) & = & F(\langle d, d(x)\rangle)(s), \\
\Psi_S(F)(\langle d, S_1\,;\,S_2\rangle)(s) & = & (\Psi_S(F)(\langle d, S_1\rangle)\,;\,\Psi_S(F)(\langle d, S_2\rangle))(s), \\
\Psi_S(F)(\langle d, S_1 \,\Box\, S_2\rangle)(s) & = & (\Psi_S(F)(\langle d, S_1\rangle) \Box \Psi_S(F)(\langle d, S_2\rangle))(s),
\end{array}$$

Then Ψ_S *is well-defined, monotone, and the function* $St_S[\![\cdot]\!]$ *defined in Definition 3.2.4 is the least fixed point of* Ψ_S.

Proof: Well-definedness of Ψ_S is readily checked. To prove monotonicity of Ψ_S assume $F_1 \le F_2$ in Sem_S. We show that $\Psi_S(F_1)(\langle d, S\rangle) \le \Psi_S(F_2)(\langle d, S\rangle)$ for any program $\langle d, S\rangle$ by induction on the structure of S. The base cases are immediate, and for the cases when $S \equiv S_1 \,\Box\, S_2$ or $S \equiv S_1\,;\,S_2$ we use induction and the fact that both the union function '$\Box$' and the composition function ';' are monotone in each argument.

Finally, since $ST_S(\mathtt{St}, \mathtt{St})$ is a cpo, Sem_S is also a cpo. Thus, by Proposition 2.2.3 the function Ψ_S has a least fixed point, which, from Definition 3.2.4, is $St_S[\![\cdot]\!]$. $\Box$

By structural induction on the statement S, and because $ST_S^{fin}(\mathtt{St}, \mathtt{St})$ is closed under the union function '$\Box$' and the composition function ';', it follows that $St_S[\![\langle d, S\rangle]\!] \in ST_S^{fin}(\mathtt{St}, \mathtt{St})$ for every program $\langle d, S\rangle$ in $\mathcal{L}_0$.

Hoare state transformers

Next we consider a domain of state transformers which can be used for identifying programs only on the basis of their sets of outcomes, if any. The main difference with the Smyth state transformers is that now we do not wish to record non-termination. Deadlocking computations are mapped to the empty set, as before.

3.2.6. Definition. The set of *Hoare state transformers* from a set X to a set Y is defined by

$$ST_H(X, Y) \;=\; X \to \mathcal{P}(Y).$$

Alternatively, Hoare state transformers can be defined as the cpo of all functions from X to $(\mathcal{H}(Y_\perp))_\perp$, the Hoare powerdomain with emptyset (added as a bottom element) of the flat cpo $Y_\perp$. We prefer our definition above since its conceptually simpler (no extra bottom elements $\perp$ have to be added to Y).

Since Hoare state transformers do not record non-termination, infinite sets of outcomes are possible also for programs with a finite non-deterministic behaviour [Bak80]. Consider for example the program $\langle d, x\rangle$ in $\mathcal{L}_0$ where the program variable x is declared as

$$d(x) = (v := v + 1 \,;\, x) \,\Box\, v := v.$$

According to the intended meaning, if we start the above program in a state where $v = 0$ then we expect that the program either fails to terminate or delivers a state in which the variable v has an arbitrary natural number as resulting outcome.

The set $ST_H(X, Y)$ is ordered by the pointwise extension of the subset inclusion, the natural order in $\mathcal{P}(Y)$. Thus, for σ and τ in $ST_H(X, Y)$,

$$\sigma \leq \tau \quad \text{if and only if} \quad \forall x \in X \colon \sigma(x) \subseteq \tau(x).$$

The set $ST_H(X, Y)$ ordered as above forms a complete partial order with least element given by the function $\lambda x \in X.\emptyset$. The least upper bound of a directed set $\{\sigma_i \mid i \in I\}$ of state transformers in $ST_H(X, Y)$ is calculated pointwise, that is,

$$(\bigvee\{\sigma_i \mid i \in I\})(x) = \bigcup\{\sigma_i(x) \mid i \in I\},$$

for all $x \in X$.

It is important to note that $ST_H(X, Y)$ is isomorphic to $\mathcal{P}(X \times Y)$, the set of all relations on X and Y. This explains why the Hoare state transformer semantics is often called *relational semantics* [Plo81a].

Every state transformer in $ST_H(X, Y)$ is a state transformer in $ST_S(X, Y)$. Hence we can define a union function and a composition function exactly in the same way as for the Smyth state transformers.

3.2.7. Definition. Let X, Y and Z be three sets. Define, for every $x \in X$ the *union function* $\Box : ST_H(X, Y) \times ST_H(X, Y) \to ST_H(X, Y)$ by

$$(\sigma_1 \Box \sigma_2)(x) = \sigma_1(x) \cup \sigma_2(x),$$

and the *composition function* $; : ST_H(X, Y) \times ST_H(Y, Z) \to ST_H(X, Z)$ by

$$(\sigma_1 \,;\, \sigma_2)(x) \quad = \quad \bigcup\{\sigma_2(y) \mid y \in \sigma_1(x)\}$$

for every $x \in X$.

The above '$\Box$' and ';' are well-defined and continuous in each argument. We are now in a position to define the Hoare state transformer semantics for $\mathcal{L}_0$.

3.2.8. Definition. The semantic function $St_H[\![\cdot]\!]$ is defined as the least function in $\mathcal{L}_0 \to ST_H(\mathtt{St}, \mathtt{St})$ such that,

$$\begin{array}{lcl}
St_H[\![\langle d, v := e\rangle]\!] & = & St_S[\![\langle d, v := e\rangle]\!],\\
St_H[\![\langle d, b{\to}\rangle]\!] & = & St_S[\![\langle d, b{\to}\rangle]\!],\\
St_H[\![\langle d, x\rangle]\!] & = & St_H[\![\langle d, d(x)\rangle]\!],\\
St_H[\![\langle d, S_1 \,;\, S_2\rangle]\!] & = & St_H[\![\langle d, S_1\rangle]\!] \,;\, St_H[\![\langle d, S_2\rangle]\!],\\
St_H[\![\langle d, S_1 \,\Box\, S_2\rangle]\!] & = & St_H[\![\langle d, S_1\rangle]\!] \Box St_H[\![\langle d, S_2\rangle]\!].
\end{array}$$

The well-definedness of the above semantics can be proved in a similar way as for the semantics $St_S[\![\cdot]\!]$.

Egli-Milner state transformers

Finally we turn to the possibility of identifying programs on the basis of what actually happens. Computations are mapped to the subset of all their possible outcomes, including $\perp$ to denote the possibility of non-termination. Note that we differ from the Smyth state transformers because we do not necessarily identify computations which fail to terminate. As always, deadlocking computations are mapped to the empty set.

3.2.9. Definition. The set of *Egli-Milner state transformers* from a set X to a set Y is defined by

$$ST_E(X, Y) \;=\; X \to \mathcal{P}(Y \cup \{\perp\}).$$

The set $ST_E(X, Y)$ can be turned into a cpo by the following order. For $\sigma, \tau \in ST_E(X, Y)$,

$$\sigma \leq \tau \quad \text{if and only if} \quad \forall x \in X\colon \; \begin{array}{l}(\perp \notin \sigma(x) \;\&\; \sigma(x) = \tau(x)) \text{ or}\\ (\perp \in \sigma(x) \;\&\; \sigma(x) \setminus \{\perp\} \subseteq \tau(x)).\end{array}$$

This ordering has been introduced for the semantics of non-deterministic programs by Egli [Egl75], and it has been studied in detail by De Bakker [Bak80]. It is often referred to as the Egli-Milner ordering because Milner has defined it in an essentially equivalent formulation (as reported by Plotkin [Plo76]). The Egli-Milner ordering is an approximation ordering: the computation represented by τ is 'better' than the one represented by σ if, for any input x, $\tau(x)$ can be obtained form $\sigma(x)$ by replacing the partialness in $\sigma(x)$ (represented by the presence of $\perp$ in $\sigma(x)$) by some set of outcomes.

Not all Egli-Milner state transformers correspond to denotations of programs that are finitely non-deterministic. We could restrict them by considering only a finite set of outcomes. However, if a computation fails to terminate then an infinite set of outcomes is also possible (essentially for the same reason as for the Hoare state transformers). Therefore, we take $ST_E^{fin}(X, Y)$ to be the set of all functions from the set X to all subsets of $Y \cup \{\perp\}$ which are either finite or contain $\perp$.

3.2.10. Lemma. *For every set X and Y, both $ST_E(X, Y)$ and $ST_E^{fin}(X, Y)$ are complete partial orders.*

Proof: If $\mathcal{V}$ is a directed set in $ST_E(X, Y)$ then

$$\bigvee \mathcal{V} \quad = \quad \lambda x \in X. \begin{cases} \bigcup\{\sigma(x) \mid \sigma \in \mathcal{V}\} & \text{if } \forall \sigma \in \mathcal{V}\colon \perp \in \sigma(x) \\ \bigcup\{\sigma(x) \setminus \{\perp\} \mid \sigma \in \mathcal{V}\} & \text{otherwise.} \end{cases} \tag{3.2}$$

Assume now that $\sigma \in ST_E^{fin}(X, Y)$ for every $\sigma \in \mathcal{V}$, and let $x \in X$. In order to show that $\bigvee \mathcal{V}$ is the least upper bound of $\mathcal{V}$ in $ST_E^{fin}(X, Y)$ we need to prove that the set $(\bigvee \mathcal{V})(x)$ is finite whenever $\perp \notin (\bigvee \mathcal{V})(x)$.

Assume $\perp \notin (\bigvee \mathcal{V})(x)$. Then by (3.2), there exists $\sigma_0 \in \mathcal{V}$ with $\perp \notin \sigma_0(x)$. Since $\mathcal{V}$ is a directed set, for every $\sigma_1 \in \mathcal{V}$, there exists $\sigma_2 \in \mathcal{V}$ which is an upper bound of both σ_0 and σ_1. By definition of the Egli-Milner order and because $\perp \notin \sigma_0(x)$ it must be the case that $\sigma_2(x) = \sigma_0(x)$. Hence

$$\bigcup\{\sigma(x) \mid \sigma \in \mathcal{V}\} = \sigma_0(x).$$

By (3.2) and because $\sigma_0(x)$ is a finite subset of Y, $(\bigvee \mathcal{V})(x)$ is also a finite subset of Y.

Finally, the function $\lambda x \in X.\{\perp\}$ is clearly the least element for both $ST_E(X, Y)$ and $ST_E^{fin}(X, Y)$. Hence they both are cpo's. $\Box$

As for the finitary Smyth state transformers, an alternative way to prove that $ST_E^{fin}(X, Y)$ is a complete partial order is to define it as the set of all functions from X to $\mathcal{E}(Y_\perp) \oplus (\mathbf{1})_\perp$, the Plotkin powerdomain with emptyset (added by means of a coalesced sum) of the flat cpo $Y_\perp$.

Next we give the semantical counterparts of the syntactic operators in $\mathcal{L}_0$.

3.2.11. Definition. Let X, Y and Z be three sets. Define, for every $x \in X$, the *union function* $\Box : ST_E(X, Y) \times ST_E(X, Y) \to ST_E(X, Y)$ by

$$(\sigma_1 \Box \sigma_2)(x) = \sigma_1(x) \cup \sigma_2(x),$$

and the *composition function* $; : ST_E(X, Y) \times ST_E(Y, Z) \to ST_E(X, Z)$ by

$$(\sigma_1 \; ; \sigma_2)(x) = \bigcup\{\sigma_2(y) \mid y \in \sigma_1(x) \setminus \{\perp\}\} \cup \{\perp \mid \perp \in \sigma_1(x)\}.$$

Both these functions are monotone in their arguments. Moreover, the set $ST_E^{fin}(X, Y)$ is closed under the union operation, and, if $\sigma_1 \in ST_E^{fin}(X, Y)$ and $\sigma_2 \in ST_E^{fin}(Y, Z)$ then $\sigma_1 \; ; \sigma_2 \in ST_E^{fin}(X, Z)$. We are now ready for the definition of the Egli-Milner state transformer semantics of $\mathcal{L}_0$.

3.2.12. Definition. The semantic function $St_E[\![\cdot]\!]$ is defined as the least function in $\mathcal{L}_0 \to ST_E(\mathtt{St}, \mathtt{St})$ such that,

$$\begin{array}{lcl} St_E[\![\langle d, v := e\rangle]\!] & = & St_S[\![\langle d, v := e\rangle]\!], \\ St_E[\![\langle d, b\!\to\rangle]\!] & = & St_S[\![\langle d, b\!\to\rangle]\!], \\ St_E[\![\langle d, x\rangle]\!] & = & St_E[\![\langle d, d(x)\rangle]\!], \\ St_E[\![\langle d, S_1 \,;\, S_2\rangle]\!] & = & St_E[\![\langle d, S_1\rangle]\!] \,;\, St_E[\![\langle d, S_2\rangle]\!], \\ St_E[\![\langle d, S_1 \,\Box\, S_2\rangle]\!] & = & St_E[\![\langle d, S_1\rangle]\!] \Box St_E[\![\langle d, S_2\rangle]\!]. \end{array}$$

We omit the proof of the well-definedness of the above semantics since it can be obtained in a similar way as for the semantics $St_S[\![\cdot]\!]$.

Relating the three state transformer models

So far we introduced three state transformer semantics for $\mathcal{L}_0$. Next we discuss how these semantics are related.

For fixed sets X and Y, define the functions $E_H : ST_E(X, Y) \to ST_H(X, Y)$ and $E_S : ST_E(X, Y) \to ST_S(X, Y)$ respectively by

$$E_H(\sigma)(x) = \sigma(x) \setminus \{\perp\} \quad \text{and} \quad E_S(\sigma)(x) = \begin{cases} Y_\perp & \text{if } \perp \in \sigma(x) \\ \sigma(x) & \text{otherwise} \end{cases}$$

for every $\sigma \in ST_E(X, Y)$ and $x \in X$. Then both E_H and E_S are strict, continuous, and onto, as can be easily verified. Moreover, if $\sigma \in ST_E^{fin}(X, Y)$ then $E_S(\sigma) \in ST_S^{fin}(X, Y)$.

3.2.13. Lemma. *For $\sigma_0, \sigma_1 \in ST_E(X, Y)$ and $\sigma_2 \in ST_E(Y, Z)$*

$$\begin{array}{c} E_S(\sigma_0 \Box \sigma_1) = E_S(\sigma_0) \Box E_S(\sigma_1) \quad \textit{and} \quad E_H(\sigma_0 \Box \sigma_1) = E_H(\sigma_0) \Box E_H(\sigma_1), \\ E_S(\sigma_0 \,;\, \sigma_1) = E_S(\sigma_0) \,;\, E_S(\sigma_1) \quad \textit{and} \quad E_H(\sigma_0 \,;\, \sigma_1) = E_H(\sigma_0) \,;\, E_H(\sigma_1). \end{array}$$

Proof: Immediate from the definitions of E_S and E_H, and of the union and composition functions on the Egli-Milner, the Smyth and the Hoare state transformers. $\Box$

Both the semantics based on the Smyth and Hoare state transformers are projections, under E_S and E_H respectively, of the semantics based on the Egli-Milner state transformers.

3.2.14. Theorem. *For all $\langle d, S\rangle \in \mathcal{L}_0$, $E_S(St_E[\![\langle d, S\rangle]\!]) = St_S[\![\langle d, S\rangle]\!]$ and $E_H(St_E[\![\langle d, S\rangle]\!]) = St_H[\![\langle d, S\rangle]\!]$.*

Proof: We prove that $E_S(St_E[\![\langle d, S\rangle]\!]) = St_S[\![\langle d, S\rangle]\!]$. The other equality $E_H(St_E[\![\langle d, S\rangle]\!]) = St_H[\![\langle d, S\rangle]\!]$ can be proved in a similar way.

Let Sem_E denote the set $\mathcal{L}_0 \to ST_E(\mathtt{St}, \mathtt{St})$, and define a monotone function $\Psi_E : Sem_E \to Sem_E$ such that $St_E[\![\cdot]\!]$ is the least fixed point of Ψ_E (the definition of Ψ_E can be obtained adapting the definition of Ψ_S given in Lemma 3.2.5).

By structural induction on S, following the definition of Ψ_E, and using Lemma 3.2.13 it is straightforward to prove that the following diagram commutes:

$$\begin{tikzcd}
Sem_E \arrow[r, "\Psi_E"] \arrow[d, "\lambda F.E_S \circ F"'] \arrow[dr, phantom, "*"] & Sem_E \arrow[d, "\lambda F.E_S \circ F"] \\
Sem_S \arrow[r, "\Psi_S"'] & Sem_S.
\end{tikzcd}$$

Since E_S is strict and continuous and Sem_E is a cpo, we can use Proposition 2.2.5: the least fixed point of Ψ_S coincides with the projection under $\lambda F \in Sem_E.E_S \circ F$ of the least fixed point of Ψ_E, showing that

$$E_S(St_E[\![\langle d, S\rangle]\!]) \quad = \quad St_S[\![\langle d, S\rangle]\!],$$

for all $\langle d, S\rangle \in \mathcal{L}_0$. □

3.3 Predicate transformer models

In this section we introduce predicate transformer models for sequential programs. We will proceed as follows. First we introduce informally predicate transformers for partial and total correctness. Then we give a partial correctness semantics and a total correctness semantics to $\mathcal{L}_0$. Subsequently, we show that for every state transformer there is an associated predicate transformer, and conversely, every predicate transformer corresponds uniquely to a state transformer. These relationships form the basic dualities we study in this thesis. The duality between the predicate transformers for total correctness and the finitary Smyth state transformers is well-known: it appears already in [Wan77, Bac81], and it is formally studied by Plotkin [Plo79]. Various generalizations of this duality have been studied in [Bes83, AP86, BK94b]. The connection between predicate transformers for partial correctness and the Hoare state transformers is presented in [Plo81a].

Predicate transformers for partial and total correctness

Let X be a set. Intensionally, a *predicate* on X is a function which maps each element of X to either *true* or *false*. We will use the extensional characterization of a predicate as the set of all points of X for which, intensionally, the predicate is true. This extensional view leads us to define the set of predicates on X as $\mathcal{P}(X)$, the collection of all subsets of X. We will usually denote predicates by P and Q. Predicates are ordered by subset inclusion when not stated otherwise.

3.3.1. Definition. A *predicate transformer* is a total function—typically denoted by π, ρ — from predicates on Y to predicates on X, that is

$$PT(Y,X) \;=\; \mathcal{P}(Y) \to \mathcal{P}(X).$$

Predicate transformers are ordered by pointwise extension of the subset order on X, that is, for $\pi, \rho \in PT(Y,X)$,

$$\pi \leq \rho \quad \text{if and only if} \quad \forall P \subseteq Y \colon \pi(P) \subseteq \rho(P).$$

The poset of predicate transformers $PT(Y,X)$ inherits much of the structure of $\mathcal{P}(X)$: as $PT(Y,X)$ is the pointwise extension of the complete Boolean algebra $\mathcal{P}(X)$, it will also be a complete Boolean algebra. Meets and joins are defined pointwise by

$$(\bigwedge_I \pi_i)(P) = \bigcap_I \pi_i(P) \quad \text{and} \quad (\bigvee_I \pi_i)(P) = \bigcup_I \pi_i(P),$$

for every set I, predicate transformers $\pi_i \in PT(Y,X)$ $(i \in I)$, and $P \subseteq Y$. Also the complement $\neg\pi$ of a predicate transformer $\pi \in PT(Y,X)$ is defined pointwise by

$$(\neg\pi)(P) \;=\; X \setminus \pi(P),$$

for every $P \subseteq Y$.

Predicate transformers in $PT(Y,X)$ can be used for the interpretation of a program which starts from a state in X and eventually terminates in some states that are elements of Y. We consider two different semantic models:

- *The total correctness model*: for a predicate P on Y and $\pi \in PT(Y,X)$, the predicate $\pi(P)$ holds precisely for those inputs $x \in X$ for which each computation of the program represented by π terminates in a final state $y \in Y$ satisfying the predicate P;

- *The partial correctness model*: for a predicate P on Y and $\pi \in PT(Y,X)$, the predicate $\pi(P)$ holds precisely for those inputs $x \in X$ for which each computation of the program represented by π either fails to terminate or terminates in a final state $y \in Y$ satisfying the predicate P;

In the total correctness model $\pi(Y)$ holds precisely for those inputs $x \in X$ for which each computation of the program represented by π terminates, whereas, according to the partial correctness model $\pi(Y) = X$.

Not every predicate transformer represents a 'reasonable' program. For example, a predicate transformer representing a program is required to preserve non-empty intersections: every computation of a program S at input x terminates in a final state $y \in Y$ satisfying the predicate $\bigcap_I P_i$ if and only if every computation of a program S at input x terminates in a final state $y \in Y$ satisfying P_i for all $i \in I$.

3.3.2. Definition. Let X and Y be two sets. We define

(i) the domain of *total correctness predicate transformers* $PT_T(Y, X)$ to be the set of all predicate transformers in $\mathcal{P}(Y) \to \mathcal{P}(X)$ that preserve non-empty intersections;

(ii) the domain of *partial correctness predicate transformers* $PT_P(Y, X)$ to be the set of all total correctness predicate transformers $\pi \in PT_T(Y, X)$ such that $\pi(Y) = X$.

Both the total and partial correctness predicate transformers are closed under arbitrary meets (defined pointwise) and functional composition. The closure under arbitrary meets turns $PT_T(Y, X)$ into a complete lattice.

We are now ready for the definition of two predicate transformer semantics for $\mathcal{L}_0$. We define them as the greatest and the least fixed point of a monotone function on the domain of all possible predicate transformer semantics for $\mathcal{L}_0$.

3.3.3. Lemma. *Let* $F \in Sem_T = \mathcal{L}_0 \to PT_T(\mathtt{St}, \mathtt{St})$ *and define* $\Psi_T : Sem_T \to Sem_T$ *inductively, for all* $P \subseteq \mathtt{St}$, *by*

$$\begin{array}{lcl}
\Psi_T(F)(\langle d, v := e\rangle)(P) & = & \{s \mid s[\mathcal{E}_{\mathcal{V}}(e)(s)/v] \in P\}, \\
\Psi_T(F)(\langle d, b{\rightarrow}\rangle)(P) & = & \{s \mid s \in \mathcal{B}_{\mathcal{V}}(b) \;\Rightarrow\; s \in P\}, \\
\Psi_T(F)(\langle d, x\rangle)(P) & = & F(\langle d, d(x)\rangle)(P), \\
\Psi_T(F)(\langle d, S_1\, ;\, S_2\rangle)(P) & = & \Psi_T(F)(\langle d, S_1\rangle)(\Psi_T(F)(\langle d, S_2\rangle)(P)), \\
\Psi_T(F)(\langle d, S_1 \,\Box\, S_2\rangle)(P) & = & \Psi_T(F)(\langle d, S_1\rangle)(P) \cap \Psi_T(F)(\langle d, S_2\rangle)(P).
\end{array}$$

Then Ψ_T *is well-defined and monotone.*

Proof: Both well-definedness and monotonicity are immediately proved using induction on the structure of $S \in \mathcal{L}_0$. □

As a consequence of Proposition 2.2.1, Ψ_T has both least and greatest fixed points. We denote them by $Wp_0[\![\cdot]\!]$ and $Wlp_0[\![\cdot]\!]$, respectively. The names Wp_0 and Wlp_0 stands for '*weakest precondition*' and '*weakest liberal precondition*' , respectively (the subscripts indicate the language to which they are referred to).

Dijkstra's weakest precondition calculus [Dij76] can be expressed by the semantics $Wp_0[\![\cdot]\!]$ if we allow 'enough' Boolean expressions in *BExp*. For example, the meaning of Dijkstra's guarded command $b{\rightarrow}S$ corresponds to $Wp_0[\![\langle d, b{\rightarrow}\, ; S\rangle]\!]$; the meaning of Dijkstra's conditional command

if $b_1{\rightarrow}S_1 \,\Box\, b_2{\rightarrow}S_2$ **fi**

is equivalent to $Wp_0[\![\langle d, x\rangle]\!]$ where the procedure variable x is declared by

$$d(x) = ((b_1\!\rightarrow\, ; \,S_1) \,\Box\, (b_2\!\rightarrow\, ; \,S_2)) \,\Box\, (b_3\!\rightarrow\, ; \,x)$$

and $\mathcal{B}\nu(b_3) = \mathtt{St} \setminus (\mathcal{B}\nu(b_1) \cup \mathcal{B}\nu(b_2)$. Finally, Dijkstra's iteration command

$$\mathbf{do}\; b_1\!\rightarrow\! S_1 \,\Box\, b_2\!\rightarrow\! S_2 \;\mathbf{od}$$

corresponds to $Wp_0[\![\langle d, x\rangle]\!]$ where the procedure variable x is declared by

$$d(x) = (((b_1\!\rightarrow\, ; \,S_1) \,; \,x) \,\Box\, ((b_2\!\rightarrow\, ; \,S_2) \,; \,x)) \,\Box\, b_3\!\rightarrow,$$

and $\mathcal{B}\nu(b_3) = \mathtt{St} \setminus (\mathcal{B}\nu(b_1) \cap \mathcal{B}\nu(b_2))$.

Another form of conditional command '$\{b\}$' for $b \in \mathit{BExp}$, is often considered [Hes88]. The computational intuition behind the command '$\{b\}$' is that it is undefined in a state in which the Boolean expression 'b' does not evaluate to true and acts as a skip otherwise. Identifying undefined with failure of termination (nothing can be guaranteed for an undefined statement), we obtain that the meaning of '$\{b\}$' is equivalent to the predicate transformer $Wp_0[\![\langle d, x\rangle]\!]$ where x is a procedure variable declared as $d(x) = b\!\rightarrow\!\Box(b'\!\rightarrow;x)$ and $\mathcal{B}\nu(b') = \mathtt{St}\setminus\mathcal{B}\nu(b)$.

By definition, the $Wp_0[\![\cdot]\!]$ semantics is about the total correctness of $\mathcal{L}_0$. Next we show that $Wlp_0[\![\cdot]\!]$ is concerned with the partial correctness of $\mathcal{L}_0$.

3.3.4. Lemma. *For every $\langle d, S\rangle \in \mathcal{L}_0$, $Wlp_0[\![\langle d, S\rangle]\!](\mathtt{St}) = \mathtt{St}$.*

Proof: We prove, by induction on α, that $\Psi_T^{[\alpha]}(\langle d, S\rangle)(\mathtt{St}) = \mathtt{St}$ for all ordinals α.

For $\alpha = 0$, it is straightforward to see (by structural induction on S) that $\Psi_T^{[0]}(\langle d, S\rangle)(\mathtt{St}) = \mathtt{St}$. Note that if $S \equiv x$, for $x \in \mathit{PVar}$, then

$$\Psi_T^{[0]}(\langle d, x\rangle)(\mathtt{St}) = F^{\top}(\langle d, d(x)\rangle)(\mathtt{St})$$

where $F^{\top}$ is the top element of Sem_T, that is, the function mapping every program $\langle d, S\rangle \in \mathcal{L}_0$ and every $P \subseteq \mathtt{St}$ to $\mathtt{St}$. Hence $F^{\top}(\langle d, d(x)\rangle)(\mathtt{St}) = \mathtt{St}$.

Next we assume for an ordinal α that for all ordinals $\beta < \alpha$,

$$\Psi_T^{[\beta]}(\langle d, S\rangle)(\mathtt{St}) = \mathtt{St},$$

and we prove that also $\Psi_T^{[\alpha]}(\langle d, S\rangle)(\mathtt{St}) = \mathtt{St}$. Recall that

$$\Psi_T^{[\alpha]}(\langle d, S\rangle)(\mathtt{St}) = \Psi_T(\bigwedge\{\Psi_T^{[\beta]} \mid \beta < \alpha\})(\langle d, S\rangle)(\mathtt{St}).$$

By structural induction on S we verify that the above right-hand side equals $\mathtt{St}$. The only interesting case is when $S \equiv x$ for $x \in \mathit{PVar}$:

$$\begin{aligned} &\Psi_T(\bigwedge\{\Psi_T^{[\beta]} \mid \beta < \alpha\})(\langle d, x\rangle)(\mathtt{St}) \\ &\quad = (\bigwedge\{\Psi_T^{[\beta]} \mid \beta < \alpha\})(\langle d, d(x)\rangle)(\mathtt{St}) \end{aligned}$$

$$\begin{aligned} &= \bigcap\{\Psi_T^{[\beta]}(\langle d, d(x)\rangle)(\mathtt{St}) \mid \beta < \alpha\} \quad \text{[meets are pointwise]}\\ &= \bigcap\{\mathtt{St} \mid \beta < \alpha\} \quad \text{[induction hypothesis]}\\ &= \mathtt{St}. \end{aligned}$$

We can conclude that $\Psi_T^{[\alpha]}(\langle d, S\rangle)(\mathtt{St}) = \mathtt{St}$ for every ordinal α. Since $Wlp_0[\![\cdot]\!]$ is defined as the greatest fixed point of Ψ_T, by Proposition 2.2.4 there exists an ordinal λ such that $Wlp_0[\![\cdot]\!] = \Psi_T^{[\lambda]}$. Therefore $Wlp_0[\![\langle d, S\rangle]\!](\mathtt{St}) = \mathtt{St}$ for every $\langle d, S\rangle \in \mathcal{L}_0$. □

Intuitively, the $Wp_0[\![\cdot]\!]$ and the $Wlp_0[\![\cdot]\!]$ semantics of $\mathcal{L}_0$ agree with the informal characterization of the total and partial correctness models. To make these correspondences precise we will give duality theorems which relate the state transformer models with these predicate transformer models.

The total correctness model

Smyth state transformers capture the operational meaning of programs for the total correctness semantic model. To determine their associated predicate transformers we define the function $\omega : ST_S(X, Y) \to PT_T(Y, X)$ by

$$\omega(\sigma)(P) \;=\; \{x \in X \mid \sigma(x) \subseteq P\}, \tag{3.3}$$

for $\sigma \in ST_S(X, Y)$ and $P \subseteq Y$. Well-definedness of ω is easily verified. If $\sigma(x) = Y_\perp$ then $x \notin \omega(\sigma)(P)$ for all predicates P of Y. Accordingly, if σ is the denotation of a program then $x \in \omega(\sigma)(P)$ precisely for those inputs $x \in X$ for which each computation of the program represented by σ terminates in a final state $y \in Y$ satisfying the predicate P.

We are now in a position to show that $ST_S(X, Y)$ and $PT_T(Y, X)$ are order-isomorphic, and that the two semantics $St_S[\![\cdot]\!]$ (based on the Smyth state transformers) and $Wp_0[\![\cdot]\!]$ (based on the total correctness predicate transformers) are isomorphic. To define an inverse for the function ω above we need the following lemma. It is a variation of the stability lemma in [Plo79, AP86].

3.3.5. Lemma. *Let π be a predicate transformer in $PT_T(Y, X)$ and $x \in X$ with $x \in \pi(Y)$. Then there is a set $q(x, \pi)$ such that*

$$x \in \pi(P) \quad \textit{if and only if} \quad q(x, \pi) \subseteq P,$$

for every $P \subseteq Y$.

Proof: Define $q(x, \pi) = \bigcap\{Q \in \mathcal{P}(Y) \mid x \in \pi(Q)\}$. If $x \in \pi(P)$ then clearly $q(x, \pi) \subseteq P$. For the converse we use the fact that total correctness predicate transformers preserve non-empty intersections. Since $x \in \pi(Y)$, the set $\{Q \in \mathcal{P}(Y) \mid x \in \pi(Q)\}$ is non-empty. Hence

$$\pi(q(x, \pi)) = \bigcap\{\pi(Q) \mid x \in \pi(Q)\},$$

from which it follows that $x \in \pi(q(x,\pi))$. Because $q(x,\pi) \subseteq P$ and π is monotone (preserving non-empty intersections),

$$\pi(q(x,\pi)) \subseteq \pi(P).$$

Thus $x \in \pi(P)$. □

For any partial correctness predicate transformer π the above lemma shows that $q(x,\pi)$ exists and that it is uniquely determined. This set can be used to obtain a state transformer from a predicate transformer. Indeed, we can now define $\omega^{-1} : PT_T(Y,X) \to ST_S(X,Y)$ by

$$\omega^{-1}(\pi)(x) \quad = \quad \begin{cases} q(x,\pi) & \text{if } x \in \pi(Y) \\ Y_\perp & \text{otherwise,} \end{cases} \tag{3.4}$$

for every $\pi \in PT_T(Y,X)$ and $x \in X$.

3.3.6. Theorem. *The function $\omega : ST_S(X,Y) \to PT_T(Y,X)$ is an order isomorphism with inverse ω^{-1}.*

Proof: We first prove that both ω and ω^{-1} are monotone. Let $\sigma_1 \le \sigma_2$ in $ST_S(X,Y)$ and let $P \subseteq Y$. If $x \in \omega(\sigma_1)(P)$ then $\sigma_1(x) \subseteq P$. But $\sigma_2(x) \subseteq \sigma_1(x)$, hence also $\sigma_2(x) \subseteq P$. It follows that $x \in \omega(\sigma_2)(P)$. Hence $\omega(\sigma_1) \le \omega(\sigma_2)$ in $PT_T(Y,X)$.

Assume now that $\pi_1 \le \pi_2$ in $PT_T(Y,X)$ and take $x \in X$. The only interesting case is when $\omega^{-1}(\pi_1)(x) \ne Y_\perp$. In this case $x \in \pi_1(Y)$. Since $\pi_1(Y) \subseteq \pi_2(Y)$, $x \in \pi_2(Y)$. Hence $\omega^{-1}(\pi_2)(x) = q(x,\pi_2)$. But $q(x,\pi_2) \subseteq q(x,\pi_1)$ because $\pi_1 \le \pi_2$. Thus $\omega^{-1}(\pi_2)(x) \subseteq \omega^{-1}(\pi_1)(x)$.

Next we prove that both ω and ω^{-1} are isomorphisms. For π in $PT_T(Y,X)$ and $P \subseteq Y$ we have

$$\begin{aligned} \omega((\omega^{-1}(\pi))(P) &= \{x \in X \mid \omega^{-1}(\pi)(x) \subseteq P\} \\ &= \{x \in X \mid x \in \pi(Y) \ \&\ q(x,\pi) \subseteq P\} \\ &= \{x \in X \mid x \in \pi(Y) \ \&\ x \in \pi(P)\} \quad \text{[Lemma 3.3.5]} \\ &= \pi(P). \quad [\pi \text{ is monotone}] \end{aligned}$$

Conversely, let σ in $ST_S(X,Y)$ and x in X. If $\sigma(x) = Y_\perp$ then $x \notin \omega(\sigma)(Y)$. Hence $\omega^{-1}(\omega(\sigma))(x) = Y_\perp = \sigma(x)$. Otherwise $\omega^{-1}(\omega(\sigma))(x) = q(x,\omega(\sigma))$. By definition of ω, $x \in \omega(\sigma)(P)$ if and only if $\sigma(x) \subseteq P$ for all $P \subseteq Y$. Hence, by Lemma 3.3.5, $q(x,\omega(\sigma)) = \sigma(x)$, from which we conclude $\omega^{-1}(\omega(\sigma))(x) = \sigma(x)$. □

Assume $\sigma \in ST_S^{fin}(X,Y)$, and let $\mathcal{V}$ be a directed set of subsets of Y. Then

$$\sigma(x) \subseteq \bigcup \mathcal{V} \quad \Rightarrow \quad \exists P \in \mathcal{V} \colon \sigma(x) \subseteq P \tag{3.5}$$

because $\mathcal{V}$ is directed and $\sigma(x)$ is either a finite set or $Y_\perp$. Hence

$$\omega(\sigma)(\bigcup \mathcal{V}) = \bigcup\{\omega(\sigma)(P) \mid P \in \mathcal{V}\},$$

that is, $\omega(\sigma)$ is continuous. Conversely, if π is a continuous predicate transformer in $PT_T(Y, X)$ then $\omega^{-1}(\pi) \in ST_S^{\mathit{fin}}(X, Y)$ because the set $q(x, \pi)$ is finite. This can be proved using the property that every set is the directed union of all its finite subsets. Hence

$$\begin{array}{rcll} q(x,\pi) & = & \bigcup\{P \subseteq q(x,\pi) \mid P \text{ finite}\} & \\ & \Leftrightarrow & x \in \pi(\bigcup\{P \subseteq q(x,\pi) \mid P \text{ finite}\}) & \text{[Lemma 3.3.5]} \\ & \Leftrightarrow & x \in \bigcup\{\pi(P) \mid P \subseteq_{\mathit{fin}} q(x,\pi)\} & [\pi \text{ is continuous}] \\ & \Leftrightarrow & \exists P \subseteq_{\mathit{fin}} q(x,\pi) \colon q(x,\pi) \subseteq P. & \text{[Lemma 3.3.5]} \end{array}$$

Therefore the isomorphism of Theorem 3.3.6 restricts to an isomorphism between $ST_S^{\mathit{fin}}(X, Y)$ and the continuous predicate transformers in $PT_T(Y, X)$.

3.3.7. Lemma. *Let $\sigma_0 \in ST_S(X, Y)$ and $\sigma_1, \sigma_2 \in ST_S(Y, Z)$. Then*

$$\begin{array}{rcl} \omega(\sigma_1 \Box \sigma_2)(P) & = & \omega(\sigma_1)(P) \cap \omega(\sigma_2)(P), \quad \textit{and} \\ \omega(\sigma_0 \,;\, \sigma_1)(P) & = & \omega(\sigma_0)(\omega(\sigma_1)(P)), \end{array}$$

for all $P \subseteq Z$.

Proof: For $P \subseteq Z$ we have

$$\begin{array}{rcl} \omega(\sigma_1 \Box \sigma_2)(P) & = & \{x \in X \mid (\sigma_1 \Box \sigma_2)(x) \subseteq P\} \\ & = & \{x \in X \mid \sigma_1(x) \cup \sigma_2(x) \subseteq P\} \\ & = & \{x \in X \mid \sigma_1(x) \subseteq P \;\&\; \sigma_2(x) \subseteq P\} \\ & = & \{x \in X \mid \sigma_1(x) \subseteq P\} \cap \{x \in X \mid \sigma_2(x) \subseteq P\} \\ & = & \omega(\sigma_1)(P) \cap \omega(\sigma_2)(P); \end{array}$$

and also

$$\begin{array}{rcl} \omega(\sigma_0 \,;\, \sigma_1)(P) & = & \{x \in X \mid (\sigma_0 \,;\, \sigma_1)(x) \subseteq P\} \\ & = & \{x \in X \mid \bigcup\{\sigma_1(y) \mid y \in \sigma_0(x)\} \subseteq P\} \\ & = & \{x \in X \mid \perp \notin \sigma_0(x) \;\&\; \forall y \in \sigma_0(x) \colon \sigma_1(y) \subseteq P\} \\ & = & \{x \in X \mid \sigma_0(x) \subseteq \{y \mid \sigma_1(y) \subseteq P\}\} \\ & = & \{x \in X \mid \sigma_0(x) \subseteq \omega(\sigma_1)(P)\} \\ & = & \omega(\sigma_0)(\omega(\sigma_1)(P)). \end{array}$$

□

By Theorem 3.3.6 and the above lemma it is immediate that if $\pi_0 \in PT_T(Y, X)$ and $\pi_1, \pi_2 \in PT_T(Z, Y)$ then

$$\begin{aligned} \omega^{-1}(\pi_1 \wedge \pi_2) &= \omega^{-1}(\pi_1) \,\Box\, \omega^{-1}(\pi_2) \\ \omega^{-1}(\pi_0 \circ \pi_1) &= \omega^{-1}(\pi_0) \,;\, \omega^{-1}(\pi_1). \end{aligned}$$

Below we demonstrate the equivalence between the $Wp_0[\![\cdot]\!]$ semantics and the $St_S[\![\cdot]\!]$ semantics of $\mathcal{L}_0$.

3.3.8. Theorem. *For all $\langle d, S\rangle \in \mathcal{L}_0$ we have*

$$\omega(St_S[\![\langle d, S\rangle]\!]) = Wp_0[\![\langle d, S\rangle]\!] \quad \textit{and} \quad \omega^{-1}(Wp_0[\![\langle d, S\rangle]\!]) = St_S[\![\langle d, S\rangle]\!].$$

Proof: We begin by proving that $\omega(St_S[\![\cdot]\!])$ is a fixed point of Ψ_T. We proceed by structural induction on the statement S. If $S \equiv v := e$ then, for $P \subseteq \mathtt{St}$,

$$\begin{aligned} \omega(St_S[\![\langle d, v := e\rangle]\!])(P) &= \{s \in \mathtt{St} \mid St_S[\![\langle d, v := e\rangle]\!](s) \subseteq P\} \\ &= \{s \in \mathtt{St} \mid s[\mathcal{E}\nu(e)(s)/v] \in P\} \\ &= \Psi_T(\omega(St_S[\![\cdot]\!]))(\langle d, v := e\rangle)(P). \end{aligned}$$

If $S \equiv b\!\rightarrow$ then, for $P \subseteq \mathtt{St}$,

$$\begin{aligned} \omega(St_S[\![\langle d, b\!\rightarrow\rangle]\!])(P) &= \{s \in \mathtt{St} \mid St_S[\![\langle d, b\!\rightarrow\rangle]\!](s) \subseteq P\} \\ &= \{s \in \mathtt{St} \mid s \in \mathcal{B}\nu(b) \;\Rightarrow\; s \in P\} \\ &= \Psi_T(\omega(St_S[\![\cdot]\!]))(\langle d, b\!\rightarrow\rangle)(P). \end{aligned}$$

If $S \equiv x$ then

$$\omega(St_S[\![\langle d, x\rangle]\!]) = \omega(St_S[\![\langle d, d(x)\rangle]\!]) = \Psi_T(\omega(St_S[\![\cdot]\!]))(\langle d, x\rangle).$$

Assume now $S \equiv S_1 \,;\, S_2$. Then, for $P \subseteq \mathtt{St}$,

$$\begin{aligned} &\Psi_T(\omega(St_S[\![\cdot]\!]))(\langle d, S_1 \,;\, S_2\rangle)(P) \\ &\quad= \Psi_T(\omega(St_S[\![\cdot]\!]))(\langle d, S_1\rangle)(\Psi_T(\omega(St_S[\![\cdot]\!]))(\langle d, S_2\rangle)(P)) \\ &\quad= \omega(St_S[\![\langle d, S_1\rangle]\!])(\omega(St_S[\![\langle d, S_2\rangle]\!])(P)) \quad \text{[induction hypothesis]} \\ &\quad= \omega(St_S[\![\langle d, S_1\rangle]\!] \,;\, St_S[\![\langle d, S_2\rangle]\!])(P) \quad \text{[Lemma 3.3.7]} \\ &\quad= \omega(St_S[\![\langle d, S_1 \,;\, S_2\rangle]\!])(P). \end{aligned}$$

In case $S \equiv S_1 \,\Box\, S_2$ we proceed similarly. Therefore $St_S[\![\cdot]\!]$ is a fixed point of Ψ_T. Since $Wp_0[\![\cdot]\!]$ is the least fixed point of Ψ_T,

$$Wp_0[\![\langle d, S\rangle]\!] \leq \omega(St[\![\langle d, S\rangle]\!]), \tag{3.6}$$

for all $\langle d, S\rangle \in \mathcal{L}_0$. Following essentially the same pattern, we can prove that $\omega^{-1}(Wp_0[\![\cdot]\!])$ is a fixed point of the semantic transformation Ψ_S defined in Lemma 3.2.5. Hence

$$St[\![\langle d, S\rangle]\!] \leq \omega^{-1}(Wp_0[\![\langle d, S\rangle]\!]). \tag{3.7}$$

Because ω and ω^{-1} form an order isomorphism, we can conclude that the inequalities in (3.6) and (3.7) are in fact equalities. $\Box$

Since for all $\langle d, S\rangle \in \mathcal{L}_0$, $St_S[\![\langle d, S\rangle]\!]$ is in $ST_S^{fin}(\mathtt{St}, \mathtt{St})$, and the latter domain is isomorphic to the set of continuous predicate transformers in $PT_T(\mathtt{St}, \mathtt{St})$, the following corollary is immediate from Theorem 3.3.8.

3.3.9. Corollary. *For $\langle d, S\rangle \in \mathcal{L}_0$, the predicate transformer $Wp_0[\![\langle d, S\rangle]\!]$ is continuous.* □

The partial correctness model

We relate the set of Hoare state transformers to the set of partial correctness predicate transformers by restricting and co-restricting the isomorphism of Theorem 3.3.6.

The set of Hoare state transformers $ST_H(X, Y)$ is a subset of $ST_S(X, Y)$. If we apply the function ω to a Hoare state transformer $\sigma \in ST_H(X, Y)$ then

$$\omega(\sigma)(Y) = \{x \in X \mid \sigma(x) \subseteq Y\} = X.$$

Thus $\omega(\sigma)$ is a partial correctness predicate transformer in $PT_P(Y, X)$. Conversely, if π is a partial correctness predicate transformer in $PT_P(Y, X)$ then, by applying ω^{-1} to π we obtain a Hoare state transformer because $x \in \pi(Y)$ for all $x \in X$. Therefore, by Theorem 3.3.6 we have the following isomorphism.

3.3.10. Theorem. *The function $\omega : ST_H(X, Y) \to PT_P(Y, X)$ is an isomorphism with inverse ω^{-1}.* □

Note that the above isomorphism is not an order isomorphism. If $\sigma_1 \leq \sigma_2$ in $ST_H(X, Y)$ then, for all $P \subseteq Y$,

$$\omega(\sigma_1)(P) \supseteq \omega(\sigma_2)(P)$$

because $\sigma_1(x) \subseteq \sigma_2(x)$ for all $x \in X$. Similarly, for $\pi_1, \pi_2 \in PT_P(Y, X)$, if $\pi_1(P) \subseteq \pi_2(P)$ for all $P \subseteq Y$ then $\omega^{-1}(\pi_1) \geq \omega^{-1}(\pi_2)$ in $ST_H(X, Y)$.

3.3.11. Theorem. *For all $\langle d, S\rangle \in \mathcal{L}_0$ we have*

$$\omega(St_H[\![\langle d, S\rangle]\!]) = Wlp_0[\![\langle d, S\rangle]\!] \quad \textit{and} \quad \omega^{-1}(Wlp_0[\![\langle d, S\rangle]\!]) = St_H[\![\langle d, S\rangle]\!].$$

Proof: In a way similar to the proof of Theorem 3.3.8, we first note that $\omega(St_H[\![\langle d, S\rangle]\!])$ is a fixed point of Ψ_T. Hence

$$\omega(St_H[\![\langle d, S\rangle]\!])(P) \subseteq Wlp_0[\![\langle d, S\rangle]\!](P), \tag{3.8}$$

for all $\langle d, S\rangle \in \mathcal{L}_0$, $P \subseteq \mathtt{St}$. Similarly, $St_H[\![\langle d, S\rangle]\!](x) \subseteq \omega^{-1}(Wlp_0[\![\langle d, S\rangle]\!])(x)$ for all $x \in X$. Since ω and ω^{-1} are monotone with respect to the opposite of the Hoare order, it follows that the above inclusions are, in fact, equalities. □

Total and partial correctness, together

Egli-Milner state transformers denote programs on the basis of what 'actually' happens. In the predicate transformer model this is done by describing both the total and the partial correctness of a program [DS90]. The relationship between the two domains is described informally by Nelson [Nel89], it is briefly mentioned by De Roever [Roe76] and De Bakker [Bak80], and it has been proved in its full generality in [BK93b, BK94b].

First we need to characterize those pairs of predicate transformers in the total and partial correctness models which denote the semantics of the same computation. To this end, assume π_1 and π_2 denote the semantics of the same program in the total and partial correctness model, respectively. Intuitively it holds that, for every predicate P on the output state space Y,

$$\pi_1(P) \quad = \quad \pi_1(Y) \cap \pi_2(P) \tag{3.9}$$

because, $\pi_1(P)$ holds for an input state x if and only if every computation of the program denoted by π_1 at input x terminates (and hence $x \in \pi_1(Y)$) in a final state satisfying the predicate P (and hence $x \in \pi_2(P)$).

3.3.12. Definition. Let X and Y be two sets. The domain of *Nelson predicate transformers* $PT_N(Y, X)$ consists of pairs (π_1, π_2) such that
(i) $\pi_1 \in PT_T(Y, X)$,
(ii) $\pi_2 \in PT_P(Y, X)$, and
(iii) $\pi_1(P) = \pi_1(Y) \cap \pi_2(P)$ for all $P \subseteq Y$.

We show that the Nelson predicate transformers are in a bijective correspondence with the Egli-Milner state transformers. Define $\eta : ST_E(X, Y) \to PT_N(Y, X)$ by

$$\eta(\sigma) \quad = \quad \langle \omega(E_S(\sigma)), \omega(E_H(\sigma))\rangle, \tag{3.10}$$

for all $\sigma \in ST_E(X, Y)$. Well-definedness of η is proved in the following lemma.

3.3.13. Lemma. *For every $\sigma \in ST_E(X, Y)$, $\eta(\sigma) \in PT_N(Y, X)$.*

Proof: Since $E_S(\sigma) \in ST_S(X, Y)$, by Theorem 3.3.6, $\omega(E_S(\sigma))$ is a total correctness predicate transformer in $PT_T(Y, X)$. Similarly, $\omega(E_H(\sigma))$ is a partial correctness predicate transformer in $PT_P(Y, X)$ because $E_H(\sigma) \in ST_H(X, Y)$.

It remains to prove (3.9). For $x \in X$ and $P \subseteq Y$,

$$\begin{aligned} x \in \omega(E_S(\sigma))(P) \quad & \Leftrightarrow \quad E_S(\sigma)(x) \subseteq P \\ & \Leftrightarrow \quad \sigma(x) \subseteq P \\ & \Leftrightarrow \quad \perp \notin \sigma(x) \;\&\; \sigma(x) \setminus \{\perp\} \subseteq P \\ & \Leftrightarrow \quad E_S(\sigma)(x) \subseteq Y \;\&\; E_H(\sigma) \subseteq P \\ & \Leftrightarrow \quad x \in \omega(E_S(\sigma))(Y) \cap \omega(E_H(\sigma))(P). \end{aligned}$$

□

A Nelson predicate transformer $\langle \pi_1, \pi_2 \rangle \in PT_N(Y, X)$ determines uniquely an Egli-Milner state transformer $\eta^{-1}(\langle \pi_1, \pi_2 \rangle)$ by putting, for $x \in X$,

$$\eta^{-1}(\langle \pi_1, \pi_2 \rangle)(x) \quad = \quad \omega^{-1}(\pi_2)(x) \cup \{\perp \mid x \notin \pi_1(Y)\}.$$

According to the intuition behind the pair $\langle \pi_1, \pi_2 \rangle$, we use the predicate transformer π_1 to determine non-terminating computations, whereas we use the predicate transformer π_2 to calculate their final outcomes.

3.3.14. Theorem. *The function $\eta : ST_E(X, Y) \to PT_N(Y, X)$ is a bijection with inverse η^{-1}.*

Proof: Let $\sigma \in ST_E(X, Y)$ and $x \in X$. We have

$$\begin{aligned}
&\eta^{-1}(\eta(\sigma))(x) \\
&\quad = \eta^{-1}(\langle \omega(E_S(\sigma)), \omega(E_H(\sigma)) \rangle)(x) \quad \text{[definition } \eta] \\
&\quad = \omega^{-1}(\omega(E_H(\sigma)))(x) \cup \{\perp \mid x \notin \omega(E_S(\sigma))(Y)\} \quad \text{[definition } \eta^{-1}] \\
&\quad = E_H(\sigma)(x) \cup \{\perp \mid E_S(\sigma)(x) = Y_\perp\} \quad \text{[Theorem 3.3.10 and definition } \omega] \\
&\quad = (\sigma(x) \setminus \{\perp\}) \cup \{\perp \mid \perp \in \sigma(x)\} \quad \text{[definition } E_H \text{ and } E_S] \\
&\quad = \sigma(x).
\end{aligned}$$

Conversely, for $\langle \pi_1, \pi_2 \rangle \in PT_N(Y, X)$, $P \subseteq Y$, and $x \in X$,

$$\begin{aligned}
&x \in \omega(E_S(\eta^{-1}(\langle \pi_1, \pi_2 \rangle)))(P) \\
&\quad \Leftrightarrow \ E_S(\eta^{-1}(\langle \pi_1, \pi_2 \rangle))(x) \subseteq P \quad \text{[definition } \omega] \\
&\quad \Leftrightarrow \ \perp \notin \eta^{-1}(\langle \pi_1, \pi_2 \rangle)(x) \ \& \ \eta^{-1}(\langle \pi_1, \pi_2 \rangle)(x) \subseteq P \quad \text{[definition } E_S] \\
&\quad \Leftrightarrow \ x \in \pi_1(Y) \ \& \ \omega^{-1}(\pi_2)(x) \subseteq P \quad \text{[definition } \eta^{-1}] \\
&\quad \Leftrightarrow \ x \in \pi_1(Y) \ \& \ x \in \pi_2(P) \quad \text{[Lemma 3.3.5]} \\
&\quad \Leftrightarrow \ x \in \pi_1(P). \quad \text{[Equation (3.9)]}
\end{aligned}$$

Hence

$$\begin{aligned}
&\eta(\eta^{-1}(\langle \pi_1, \pi_2 \rangle)) \\
&\quad = \langle \omega(E_S(\eta^{-1}(\langle \pi_1, \pi_2 \rangle))), \omega(E_H(\eta^{-1}(\langle \pi_1, \pi_2 \rangle))) \rangle \quad \text{[definition } \eta] \\
&\quad = \langle \pi_1, \omega(\eta^{-1}(\langle \pi_1, \pi_2 \rangle) \setminus \{\perp\}) \rangle \quad \text{[above calculation and definition } E_H] \\
&\quad = \langle \pi_1, \omega(\omega^{-1}(\pi_2)) \rangle \quad \text{[definition } \eta^{-1}] \\
&\quad = \langle \pi_1, \pi_2 \rangle. \quad \text{[Theorem 3.3.10]} \qquad \square
\end{aligned}$$

The set of Nelson predicate transformers $PT_N(Y,X)$ can now be turned into a partial order by the order induced by η^{-1} on $PT_N(Y,X)$: for $\langle\pi_1,\pi_2\rangle$ and $\langle\pi_3,\pi_4\rangle$ in $PT_N(Y,X)$, define

$$\langle\pi_1,\pi_2\rangle \leq \langle\pi_3,\pi_4\rangle \quad \text{if and only if} \quad \eta^{-1}(\langle\pi_1,\pi_2\rangle) \leq \eta^{-1}(\langle\pi_3,\pi_4\rangle).$$

The order on $PT_N(Y,X)$ satisfies the following equation.

3.3.15. Lemma. *For all $\langle\pi_1,\pi_2\rangle$ and $\langle\pi_3,\pi_4\rangle$ in $PT_N(Y,X)$,*

$$\langle\pi_1,\pi_2\rangle \leq \langle\pi_3,\pi_4\rangle \;\Leftrightarrow\; \forall P\subseteq Y\colon \pi_1(P)\subseteq\pi_3(P) \;\&\; \pi_2(P)\supseteq\pi_4(P).$$

Proof: Let us use σ as shorthand for $\eta^{-1}(\langle\pi_1,\pi_2\rangle)$ and τ as shorthand for $\eta^{-1}(\langle\pi_3,\pi_4\rangle)$. Assume first $\sigma\leq\tau$ in $ST_E(X,Y)$ and let $P\subseteq Y$.

If $x\in\pi_1(P)$ then $\bot\notin\sigma(x)$. Since $\sigma\leq\tau$, $\sigma(x)=\tau(x)$. Because $x\in\pi_1(P)=\omega(E_S(\sigma))(P)$ it follows that $x\in\pi_3(P)=\omega(E_S(\tau))(P)$. Therefore $\pi_1(P)\subseteq\pi_3(P)$.

If $x\in\pi_4(P)$ we have to consider two cases depending on the presence of $\bot$ in $\sigma(x)$. In case $\bot\notin\sigma(x)$, $\sigma\leq\tau$ implies $\sigma(x)=\tau(x)$. Hence $x\in\pi_4(P)=\omega(E_H(\tau))(P)$ implies $x\in\omega(E_H(\sigma))(P)=\pi_2(P)$. In the other case $\bot\in\sigma(x)$. Since $\sigma\leq\tau$ then $\sigma(x)\setminus\{\bot\}\subseteq\tau(x)$. Thus $\sigma(x)\setminus\{\bot\}\subseteq\tau(x)\setminus\{\bot\}$, that is, $E_H(\sigma)(x)\subseteq E_H(\tau)(x)$. Hence $x\in\pi_4(P)=\omega(E_H(\tau))(P)$ implies $x\in\omega(E_H(\sigma))(P)=\pi_2(P)$. Therefore $\pi_2(P)\supseteq\pi_4(P)$.

For the converse, assume that $\pi_1(P)\subseteq\pi_3(P)$ and $\pi_2(P)\supseteq\pi_4(P)$ for all $P\subseteq Y$. First note that for every $x\in X$,

$$\omega^{-1}(\pi_2)(x)\subseteq\omega^{-1}(\pi_4)(x) \tag{3.11}$$

because $\pi_4(P)\subseteq\pi_2(P)$ for all $P\subseteq Y$. Next we distinguish two cases.

If $\bot\notin\sigma(x)$ then by definition of η^{-1} $x\in\pi_1(Y)$ and $\sigma(x)=\omega^{-1}(\pi_2)(x)$. Since $\pi_1(Y)\subseteq\pi_3(Y)$, $x\in\pi_3(Y)$. Thus $\bot\notin\tau(x)$ and $\tau(x)=\omega^{-1}(\pi_4)(x)$. By (3.11) it follows $\sigma(x)\subseteq\tau(x)$. We still need to prove the reverse inclusion. Because $\langle\pi_1,\pi_2\rangle$ is a Nelson predicate transformer, $x\in\pi_1(Y)$ and, by Lemma 3.3.5, $x\in\pi_2(\omega^{-1}(\pi_2)(x))$, it follows that $x\in\pi_1(\omega^{-1}(\pi_2)(x))$. Hence $x\in\pi_3(\omega^{-1}(\pi_2)(x))$. Because also $\langle\pi_3,\pi_4\rangle$ is a Nelson predicate transformer, $x\in\pi_4(\omega^{-1}(\pi_2)(x))$. Thus, by Lemma 3.3.5, $\omega^{-1}(\pi_4)(x)=q(x,\pi_4)\subseteq\omega^{-1}(\pi_2)(x)$. Therefore $\tau(x)\subseteq\sigma(x)$.

If $\bot\in\sigma(x)$ then $\sigma(x)\setminus\{\bot\}=\omega^{-1}(\pi_2)(x)$ by definition of η^{-1}. Thus, by equation (3.11), $\sigma(x)\setminus\{\bot\}\subseteq\omega^{-1}(\pi_4)(x)$. Since $\omega^{-1}(\pi_4)(x)\subseteq\tau(x)$ by definition of η^{-1}, we obtain that $\sigma(x)\setminus\{\bot\}\subseteq\tau(x)$. □

The above characterization of the order between Nelson predicate transformers is used in [Roe76] to give an early treatment of recursion in the original weakest precondition calculus of Dijkstra [Dij76], based on continuity of the weakest preconditions. A more detailed treatment of the recursion is given in [Heh79] and [Bak80].

We conclude this section by showing that the Egli-Milner state transformer semantics of $\mathcal{L}_0$ corresponds to the pair of weakest precondition and weakest liberal precondition semantics. For $\langle d, S\rangle \in \mathcal{L}_0$ we have

$$\begin{aligned}
&\eta(St_E[\![\langle d, S\rangle]\!]) \\
&\quad = \langle \omega(E_S(St_E[\![\langle d, S\rangle]\!])), \omega(E_H(St_E[\![\langle d, S\rangle]\!]))\rangle \\
&\quad = \langle \omega(St_S[\![\langle d, S\rangle]\!]), \omega(St_H[\![\langle d, S\rangle]\!]))\rangle \qquad \text{[Theorem 3.2.14]} \\
&\quad = \langle Wp_0[\![\langle d, S\rangle]\!], Wlp_0[\![\langle d, S\rangle]\!]\rangle. \qquad \text{[Theorems 3.3.8 and 3.3.11]}
\end{aligned}$$

As a consequence of the above, we obtain that the weakest precondition semantics $Wp_0[\![\langle d, S\rangle]\!]$ and the weakest liberal precondition semantics $Wlp_0[\![\langle d, S\rangle]\!]$ of a program $\langle d, S\rangle \in \mathcal{L}_0$ satisfy the pairing condition (3.9).

3.4 Can a backtrack operator be added to $\mathcal{L}_0$?

In this section we study the incorporation of a backtrack operator into our language $\mathcal{L}_0$. The backtrack operator is a binary operator '$\boxtimes$' which backtracks to the second component if the first component deadlocks. We define it in the domain of Egli-Milner state transformers to derive its weakest precondition semantics. Maybe surprisingly, the backtrack operator is not monotone with respect to the order of the total correctness predicate transformers. To repair the problem a new order can be defined which refines the ordinary order on predicate transformers and such that the backtrack operator becomes monotone. However, sequential composition is not monotone with respect to this new order. In order to justify the well-definedness of a weakest precondition semantics for $\mathcal{L}_0$ extended with a backtrack operator we prove that under certain conditions the least fixed point of a non-monotone function exists.

Our extension of $\mathcal{L}_0$ is a variation of the language studied in [Nel89]. In this article a weakest precondition semantics together with a weakest liberal precondition semantics for a language with a backtrack operator is given. Below we will concentrate only on a weakest precondition semantics.

3.4.1. Definition. (i) The set $(S \in)\, Stat_B$ of *statements* is given by

$$S \;::=\; v := e \mid b{\rightarrow} \mid x \mid S\,;S \mid S \,\Box\, S \mid S \boxtimes S\,.$$

(ii) The set $(d \in)\, Decl_B$ of *declarations* is defined by $Decl_B = PVar \rightarrow Stat_B$.

(iii) The language $\mathcal{L}_B$ is given by $Decl_B \times Stat_B$.

To guide the intuition about the backtrack operator '$\boxtimes$' we define the corresponding semantical operator in the domain of the Egli-Milner state transformers. For $\sigma_1, \sigma_2 \in ST_E(X, Y)$ define $\sigma_1 \boxtimes \sigma_2$ by

$$(\sigma_1 \boxtimes \sigma_2)(x) \;=\; \begin{cases} \sigma_2(x) & \text{if } \sigma_1(x) = \emptyset \\ \sigma_1(x) & \text{otherwise,} \end{cases}$$

for $x \in X$. A similar definition can be given for the Smyth state transformers and for the Hoare state transformers. It is a straightforward verification to see that

$$\boxtimes : ST_E(X, Y) \times ST_E(X, Y) \to ST_E(X, Y)$$

is a monotone function. However this is not true with respect to the order of the Smyth state transformers $ST_S(X, Y)$. Indeed if $y_1, y_2 \in Y$ then

$$\lambda x.\{y_1\} \leq \lambda x.\emptyset$$

in $ST_S(X, Y)$, but,

$$\begin{array}{rcl} \lambda x.\{y_1\} \boxtimes \lambda x.\{y_2\} & = & \lambda x.\{y_1\} \\ & \not\leq & \lambda x.\{y_2\} \\ & = & \lambda x.\emptyset \boxtimes \lambda x.\{y_2\} \end{array}$$

The above monotonicity problem is caused by the fact that the function $\lambda x.\emptyset$ is the top element of $ST_S(X, Y)$. In $ST_E(X, Y)$ this is not the case, and indeed the backtrack operator is monotone. We can try to define a new domain of state transformers between $ST_S(X, Y)$ and $ST_E(X, Y)$ by introducing a new order on the Smyth state transformers which preserves deadlock. The idea is that a state transformer which does not deadlock cannot be substituted by another which does, even if more can be guaranteed for it.

3.4.2. Definition. Define $ST_D(X, Y)$ to be the set of all functions from X to $\mathcal{P}(Y) \cup \{Y_\perp\}$ ordered as follows. For $\sigma, \tau \in ST_D(X, Y)$,

$$\begin{array}{lll} \sigma \leq \tau & \text{if and only if} & \forall x \in X : (\tau(x) \neq \emptyset \ \&\ \sigma(x) \supseteq \tau(x)) \text{ or} \\ & & \qquad (\tau(x) = \emptyset \ \&\ (\sigma(x) = \emptyset \text{ or } \sigma(x) = Y_\perp)). \end{array}$$

As for $ST_S(X, Y)$, the above domain $ST_D(X, Y)$ is a partial order with the function $\lambda x.\{Y_\perp\}$ as bottom element. However $ST_D(X, Y)$ need not to be a cpo. For example let $\mathbb{N}$ be the set of natural numbers, and consider in $ST_D(X, \mathbb{N})$ the following directed set

$$\lambda x.\mathbb{N} \leq \lambda x.\mathbb{N} \setminus \{0\} \leq \lambda x.\mathbb{N} \setminus \{0, 1\} \leq \dots .$$

It has no upper bound in $ST_D(X, \mathbb{N})$ (in $ST_S(X, \mathbb{N})$ it would have the function $\lambda x.\emptyset$ as a least upper bound).

It is now easy to see that the backtrack operator '$\boxtimes$' is monotonic with respect to the new domain $ST_D(X, Y)$. However the composition function ';', defined exactly as for $ST_S(X, Y)$, is not monotone anymore. For $y_1, y_2 \in Y$,

$$\lambda x.\{y_1, y_2\} \leq \lambda x.\{y_1\}$$

in $ST_D(X, Y)$. If we compose them with the function $\sigma \in ST_S(Y, Z)$ which maps y_2 to $\{z\} \subseteq Z$ and every other $y \in Y$ to $\emptyset$ we obtain

$$\begin{aligned}\lambda x.\{y_1, y_2\} \,;\, \sigma &= \lambda x.\{z\} \\ &\not\leq \lambda x.\emptyset \\ &= \lambda x.\{y_1\} \,;\, \sigma.\end{aligned}$$

Next we turn to a weakest precondition semantics for $\mathcal{L}_B$. First we use the isomorphism of Theorem 3.3.6 to derive the semantical backtrack operator in the domain of total correctness predicate transformers. For $\sigma_1, \sigma_2 \in ST_S(X, Y)$ let

$$\pi_1 = \omega(\sigma_1) \quad \text{and} \quad \pi_2 = \omega(\sigma_2).$$

Then $\sigma_1 = \omega^{-1}(\pi_1)$ and $\sigma_2 = \omega^{-1}(\pi_2)$. For $P \subseteq Y$,

$$\begin{aligned}&\omega(\sigma_1 \boxtimes \sigma_2)(P) \\ &\quad = \{x \in X \mid (\sigma_1 \boxtimes \sigma_2)(x) \subseteq P\} \\ &\quad = \{x \in X \mid \sigma_1(x) = \emptyset \;\&\; \sigma_2(x) \subseteq P\} \cup \\ &\quad\ \{x \in X \mid \sigma_1(x) \neq \emptyset \;\&\; \sigma_1(x) \subseteq P\} \\ &\quad = (\{x \in X \mid \sigma_1(x) \subseteq \emptyset\} \cap \{x \in X \mid \sigma_2(x) \subseteq P\}) \cup \\ &\quad\ (X \setminus \{x \in X \mid \sigma_1(x) \subseteq \emptyset\} \cap \{x \in X \mid \sigma_1(x) \subseteq P\}) \\ &\quad = (\omega(E_S(\sigma_1))(\emptyset) \cap \omega(E_S(\sigma_2))(P)) \cup \\ &\quad\ ((X \setminus \omega(E_S(\sigma_1))(\emptyset)) \cap \omega(E_S(\sigma_1))(P)) \\ &\quad = (\pi_1(\emptyset) \cap \pi_2(P)) \cup ((X \setminus \pi_1(\emptyset)) \cap \pi_1(P)) \\ &\quad = \pi_1(P) \cap (\pi_1(\emptyset) \;\Rightarrow\; \pi_2(P)),\end{aligned}$$

where $P \;\Rightarrow\; Q$ is a shorthand for $(P \cap Q) \cup (X \setminus P)$. The above justifies the following definition.

3.4.3. Definition. For $\pi_1, \pi_2 \in PT_T(Y, X)$ define $\pi_1 \boxtimes \pi_2 \in PT_T(Y, X)$ by

$$(\pi_1 \boxtimes \pi_2)(P) \;=\; \pi_1(P) \cap (\pi_1(\emptyset) \;\Rightarrow\; \pi_2(P)),$$

for all $P \subseteq Y$.

Since ω is an order-preserving isomorphism '$\boxtimes$' is not monotone with respect to the order in $PT_T(Y, X)$. Nevertheless we want to define the weakest precondition semantics of $\mathcal{L}_B$ in the same way as we did in Lemma 3.3.3 for the weakest precondition semantics of $\mathcal{L}_0$: as the least fixed point of a higher order transformation.

3.4.4. Definition. Let $F \in Sem_B = \mathcal{L}_B \to PT_T(\mathtt{St}, \mathtt{St})$ and define $\Psi_B : Sem_B \to Sem_B$ inductively by

$$
\begin{array}{lcl}
\Psi_B(F)(\langle d, v := e\rangle) & = & Wp_0[\![\langle d, v := e\rangle]\!], \\
\Psi_B(F)(\langle d, b{\rightarrow}\rangle) & = & Wp_0[\![\langle d, b{\rightarrow}\rangle]\!], \\
\Psi_B(F)(\langle d, x\rangle) & = & F(\langle d, d(x)\rangle), \\
\Psi_B(F)(\langle d, S_1 \,;\, S_2\rangle) & = & \Psi_B(F)(\langle d, S_1\rangle) \circ \Psi_B(F)(\langle d, S_2\rangle), \\
\Psi_B(F)(\langle d, S_1 \,\Box\, S_2\rangle) & = & \Psi_B(F)(\langle d, S_1\rangle) \wedge \Psi_B(F)(\langle d, S_2\rangle), \\
\Psi_B(F)(\langle d, S_1 \boxtimes S_2\rangle)(P) & = & \Psi_B(F)(\langle d, S_1\rangle) \boxtimes \Psi_B(F)(\langle d, S_2\rangle).
\end{array}
$$

Well-definedness of Ψ_B is straightforwardly checked, since it is based on the well-definedness of the corresponding semantical operators in $PT_B(\mathtt{St}, \mathtt{St})$. Since the semantical operator '$\boxtimes$' is not monotone, also Ψ_B is not monotone. At first sight it seems that we cannot define a weakest precondition semantics for $\mathcal{L}_B$ as the least fixed point of Ψ_B because the ordinary fix-point methods require Ψ_B to be at least monotone.

However, we show that, under certain conditions, the least fixed point of a non-monotonic function on a poset (which need not to be complete) exists and that it can be calculated by iteration.

3.4.5. Proposition. *Let P be a cpo and let Q be a poset such that there there is an onto and continuous function $h : P \to Q$. Assume also that, for every $y \in Q$ there is a top element in $h^{-1}(y)$, that is, there exists $z \in h^{-1}(y)$ such that $x \leq z$ for all $x \in h^{-1}(y)$. If $f : P \to P$ is a monotone function then every function $g : Q \to Q$ making the following diagram commute*

$$
\begin{tikzcd}
P \arrow[r, "f"] \arrow[d, "h"'] \arrow[dr, phantom, "*"] & P \arrow[d, "h"] \\
Q \arrow[r, "g"'] & Q
\end{tikzcd}
$$

has a least fixed point. Moreover, for every ordinal α, $g^{\langle\alpha\rangle}$ exists and equals $h(f^{\langle\alpha\rangle})$.

Proof: By Proposition 2.2.3 f has as least fixed point $f^{\langle\lambda\rangle}$, for some ordinal λ. We have:

$$h(f^{\langle\lambda\rangle}) = h(f^{\langle\lambda+1\rangle}) = h(f(f^{\langle\lambda\rangle})) = g(h(f^{\langle\lambda\rangle})).$$

So $h(f^{\langle\lambda\rangle})$ is a fixed point of g. Next we prove $h(f^{\langle\lambda\rangle})$ is also the least one.

Let $y \in Q$ be such that $g(y) = y$ and let z be the top element in $h^{-1}(y)$. We prove by induction on ordinals that $f^{\langle\alpha\rangle} \leq z$ for every ordinal α. In the proof below we need the fact that $f(z) \leq z$ which can justified by the following

$$h(f(z)) = g(h(z)) = g(y) = y.$$

If $\alpha = 0$ then $f^{\langle\alpha\rangle} = f(\bot) \leq f(z) \leq z$. Assume now that $f^{\langle\beta\rangle} \leq z$ for all ordinals $\beta < \alpha$. We have

$$\begin{aligned}(\forall\beta < \alpha : f^{\langle\beta\rangle} \leq z) &\Rightarrow \bigvee\{f^{\langle\beta\rangle} \mid \beta < \alpha\} \leq z \\ &\Rightarrow f(\bigvee\{f^{\langle\beta\rangle} \mid \beta < \alpha\}) \leq f(z) \quad [f \text{ is monotone}] \\ &\Rightarrow f^{\langle\alpha\rangle} \leq z. \quad [\text{definition of } f^{\langle\alpha\rangle} \text{ and } f(z) \leq z]\end{aligned}$$

It follows that $f^{\langle\lambda\rangle} \leq z$. Hence, by monotonicity of h,

$$h(f^{\langle\lambda\rangle}) \leq h(z) = y,$$

from which we can conclude that $h(f^{\langle\lambda\rangle})$ is the least fixed point of g.

It remains to prove that $g^{\langle\alpha\rangle} = h(f^{\langle\alpha\rangle})$ for every ordinal α. Since h is onto and monotone, it is also strict. Hence, for $\alpha = 0$,

$$h(f^{\langle\alpha\rangle}) = h(f(\bot)) = g(h(\bot)) = g(\bot) = g^{\langle\alpha\rangle}.$$

Using induction on ordinals we have for $\alpha > 0$

$$\begin{aligned}h(f^{\langle\alpha\rangle}) &= h(f(\bigvee\{f^{\langle\beta\rangle} \mid \beta < \alpha\})) \\ &= g(h(\bigvee\{f^{\langle\beta\rangle} \mid \beta < \alpha\})) \quad [\text{commutativity}] \\ &= g(\bigvee\{h(f^{\langle\beta\rangle}) \mid \beta < \alpha\}) \quad [h \text{ is continuous}] \\ &= g(\bigvee\{g^{\langle\beta\rangle} \mid \beta < \alpha\}) \quad [\text{induction hypothesis}] \\ &= g^{\langle\alpha\rangle}. \quad [\text{by definition}]\end{aligned}$$

□

In order to apply the above proposition consider the cpo $Sem_E = \mathcal{L}_0 \to ST_E(\mathtt{St},\mathtt{St})$, and define the transformation $\Phi : Sem_E \to Sem_B$ by

$$\Phi(F)(\langle d, S\rangle) = \omega(E_S(F(\langle d, S\rangle))).$$

Since $E_S : ST_E(\mathtt{St},\mathtt{St}) \to ST_B(\mathtt{St},\mathtt{St})$ is strict, onto and continuous, and $\omega : ST_B(\mathtt{St},\mathtt{St}) \to PT_T(\mathtt{St},\mathtt{St})$ is an order isomorphism, Φ is onto and continuous. Moreover, if $\sigma \in ST_B(\mathtt{St},\mathtt{St})$ then σ is also a function in $ST_E(\mathtt{St},\mathtt{St})$ and $E_S(\sigma) = \sigma$. Clearly σ is the top element of $E_S^{-1}(\sigma)$. Hence also $\Phi^{-1}(F)$ has a top element for every $F \in Sem_B$.

3.4.6. Theorem. *The function $\Psi_B : Sem_B \to Sem_B$ has a least fixed point which can be calculated by iteration from the bottom element of Sem_B.*

Proof: Define $\Psi : Sem_E \to Sem_E$ inductively by

$$\begin{array}{lcl}
\Psi_E(F)(\langle d, v := e\rangle) & = & St_E[\![\langle d, v := e\rangle]\!], \\
\Psi_E(F)(\langle d, b\to\rangle) & = & St_E[\![\langle d, b\to\rangle]\!], \\
\Psi_E(F)(\langle d, x\rangle) & = & F(\langle d, d(x)\rangle), \\
\Psi_E(F)(\langle d, S_1 \,;\, S_2\rangle) & = & \Psi_E(F)(\langle d, S_1\rangle) \,;\, \Psi_E(F)(\langle d, S_2\rangle), \\
\Psi_E(F)(\langle d, S_1 \,\Box\, S_2\rangle) & = & \Psi_E(F)(\langle d, S_1\rangle) \,\Box\, \Psi_E(F)(\langle d, S_2\rangle), \\
\Psi_E(F)(\langle d, S_1 \boxtimes S_2\rangle)(P) & = & \Psi_E(F)(\langle d, S_1\rangle) \boxtimes \Psi_E(F)(\langle d, S_2\rangle).
\end{array}$$

Well-definedness and monotonicity of Ψ_E can be straightforwardly checked. It is ultimately based on the monotonicity of the corresponding state transformer constructors. Moreover, by induction on the structure of S, and using Theorem 3.2.14, Theorem 3.3.8, and the definition of '$\boxtimes$' we have that

$$\Phi(\Psi_E(F))(\langle d, S\rangle) = \Psi_B(\Phi(F))(\langle d, S\rangle)$$

for all $\langle d, S\rangle \in \mathcal{L}_B$. Therefore by Proposition 3.4.5 Ψ_B has a least fixed point which can be calculated by iteration from the bottom element of Sem_B. $\Box$

The least fixed point of Ψ_B defines the weakest precondition semantics for $\mathcal{L}_B$.

3.5 Concluding notes

The predicate transformer semantics we presented in this chapter is formulated using higher-order transformations. Hence predicate transformers are regarded as basic objects in contrast to the more traditional view which regards predicates on states as basic objects. Accordingly, we treated recursion at the level of predicate transformers whereas for example Dijkstra and Scholten [DS90] treat recursion at the level of predicates.

Several semantic domains we introduced in this chapter are general enough to support both recursion and unbounded non-determinism. For example our Egli-Milner state transformer domain $ST_E(X, Y)$ is more general than the similar domain for countable non-determinism of Apt and Plotkin [AP86], while our predicate transformers domain $PT_T(Y, X)$ is equivalent to the domain of predicate transformers for unbounded non-determinism treated in [DG86, Hes89].

We have not used the capability of the domains to express unbounded non-determinism. In this chapter we only treated a language without specification constructs. An extension of the language $\mathcal{L}_0$ with this kind of constructs is treated in Chapter 4.

The results of this chapter can be extended to capture the semantics of more general programs than the sequential ones. In Chapter 7 we treat an example of a program which interacts with its environment by extending $\mathcal{L}_0$ with a parallel operator. The key step towards this goal is a refinement of our definition of predicates. In Chapter 5 affirmative predicates are introduced as open sets of a topological space, and in Chapter 6 we introduce two kinds of topological predicate transformers which generalize the total and the partial correctness predicate transformers. Dualities between state transformers and topological predicate transformers are also studied in Chapter 6.

Chapter 4

The refinement calculus

Predicate transformers were introduced in the previous chapter as a mathematical domain for the semantics of sequential programming languages. The goal is to use this domain for the support of systematic development of programs from their formal specifications [DS90]. However, the domain is not yet suited for a weakest precondition semantics of a language which includes certain specification constructs. For example, it would be nice if angelic non-determinism were allowed. This is useful, for example, in data abstraction via inverse commands [Bac89, GM91]. Another useful extension is to allow unbounded non-determinacy both for angelic and for demonic choice.

An extension of the domain which supports both unbounded angelic non-determinism and unbounded demonic non-determinism is given in the framework of the refinement calculus. The language of the refinement calculus as introduced by Back [Bac80] combines basic predicate transformers (which generalize assignments and conditionals), functional composition, and the lattice operations of infinite meets and infinite joins. The language is expressive enough to model both executable sequential programs and abstract specifications. The language of the refinement calculus has a predicate transformer semantics. The domain of this semantics consists of the monotonic predicate transformers. This semantics is based on a lattice theoretical interpretation [Wri90, Hes92a]: demonic choice is modeled by the meet of programs and angelic choice is modeled by the join. The lattice of predicate transformers is the basis of the refinement calculus and was first introduced in [Bac80], and successively developed in [Mor87, Mor90, BW90b, Wri90].

The execution of a statement in the refinement calculus can also be described as a game between two parties with the goal of trying or preventing, respectively, to reach a state in which a given predicate holds. This game-theoretical interpretation is inspired by the and/or programs of Harel [Har80] and it is developed for the refinement calculus by Back and Von Wright [BW96]. A game semantics for a language similar to the language of the refinement calculus is also given by Hesselink [Hes94].

In this chapter we give a short overview of the refinement calculus. Then we extend the language $\mathcal{L}_0$ to a language $\mathcal{L}_1$ with the specification constructs of the refinement calculus. A backward predicate transformer semantics is given. We also give a forward semantics for $\mathcal{L}_1$. It is based on a duality between predicate transformers and completely distributive lattices. The idea is to model commands of the calculus as functions mapping an input state to the collection of all predicates satisfiable by every output of the command. As in the previous chapter, we show that the backward semantics and the forward semantics are isomorphic. Based on the operational interpretation of the refinement calculus as a two-person game, Back and von Wright [BW96] also present a forward semantics of the refinement calculus. They also present a duality between predicate transformers and the two-step game domain. Although their duality result and forward semantics coincide with our duality and forward semantics, they have been found independently.

We conclude the chapter by giving an operational semantics for $\mathcal{L}_1$ using hyper transition systems. The hyper transition systems (a generalization of transition systems) specify the atomic steps of the computations. We show that the operational and the forward semantics coincide.

4.1 Specification and refinement

The *specification* of a sequential program consists usually of the declaration of a set specifying all possible states in which the program is allowed to work, a precondition (a predicate on the set of states) and a postcondition (also a predicate on the set of states). The postcondition specifies states in which the program has to terminate when started in a state satisfying the precondition [Gri81].

We need a calculus which includes at least a 'reasonable' programming language, a specification language and a definition of a satisfaction relation between programs and specifications. Moreover, we want to have refinement relations both for specifications and for programs.

If we take the language $\mathcal{L}_0$ defined in the previous chapter as programming language, and $\mathtt{St}$ as the set of states in which a program of $\mathcal{L}_0$ is allowed to work, then a pair (P, Q) of subsets of $\mathtt{St}$ can be seen as a specification (with P as precondition and Q as postcondition). A program $\langle d, S\rangle \in \mathcal{L}_0$ satisfies a specification (P, Q) if $P \subseteq Wp_0[\![\langle d, S\rangle]\!](Q)$: every computation of $\langle d, S\rangle$ starting in a state $x \in P$ is guaranteed to terminate in a state satisfying Q. By Theorem 3.3.6, we could equivalently say that $\langle d, S\rangle$ satisfies a specification (P, Q) if $St_S[\![\langle d, S\rangle]\!](x) \subseteq Q$ for every $x \in P$.

A specification can be refined by another one provided that any program satisfying the refined specification satisfies also the original one. Thus a specification (P, Q) is refined by a specification (P', Q') if $P' \subseteq P$ and $Q \subseteq Q'$. In this case, for a program $\langle d, S\rangle \in \mathcal{L}_0$, if $P \subseteq Wp_0[\![\langle d, S\rangle]\!](Q)$ then also

$P' \subseteq Wp_0[\![\langle d, S\rangle]\!](Q')$ by monotonicity of $Wp_0[\![\cdot]\!]$. Hence (P', Q') is satisfied by any program which satisfies (P, Q).

In the same way a program can be refined by another one provided that any specification satisfied by the original program is also satisfied by the refined one. For example, a program $\langle d, S\rangle \in \mathcal{L}_0$ is refined by a program $\langle d', S'\rangle \in \mathcal{L}_0$ if, for all $Q \subseteq \mathtt{St}$, $Wp_0[\![\langle d, S\rangle]\!](Q) \subseteq Wp_0[\![\langle d', S'\rangle]\!](Q)$. In this case, if (P, Q) is a specification which is satisfied by $\langle d, S\rangle$ then (P, Q) is also satisfied by $\langle d', S'\rangle$.

In the synthesis of programs from specifications it can be useful to have a single language for programs and specifications, and to have a single relation for expressing the refinement of specifications and programs. The refinement calculus uses a language describing monotonic predicate transformers as such a single language.

4.1.1. Definition. Let X and Y be two sets. Define $PT_M(Y, X)$ to be the set of all *monotonic predicate transformers* in $PT(Y, X)$ ordered as in $PT(Y, X)$, i.e., for $\pi_1, \pi_2 \in PT_M(Y, X)$,

$$\pi_1 \leq \pi_2 \quad \text{if and only if} \quad \forall P \subseteq Y \colon \pi_1(P) \subseteq \pi_2(P).$$

The order between monotonic predicate transformers is the refinement order: the predicate transformer $\pi_1 \in PT_M(Y, X)$ is said to be *refined by* $\pi_2 \in PT_M(Y, X)$ if $\pi_1 \leq \pi_2$ in $PT_M(Y, X)$.

4.1.2. Definition. A monotonic predicate transformer $\pi \in PT_M(Y, X)$ is said to be *totally correct* with respect to a precondition $P \subseteq X$ and a postcondition $Q \subseteq Y$ if $P \subseteq \pi(Q)$.

In other words, every computation of a program specified by the predicate transformer π at input $x \in P$ terminates in a final state satisfying the predicate Q. A monotonic predicate transformer π is said to be terminating if $\pi(Y) = X$. The restriction to monotonicity for π in the above definition can be justified as follows. Assume (P, Q) is a specification and let π be a predicate transformer denoting a class of programs which satisfies the above specification. If $Q \subseteq Q'$ then every computation of every program which for input $x \in P$ terminates in a state satisfying Q, terminates also in a state satisfying Q'. Hence $\pi(Q) \subseteq \pi(Q')$.

Refinement coincides with preservation of total correctness: π_1 refines π_2 if π_1 satisfies every total correctness specification that π_2 satisfies. Moreover this condition characterizes the refinement relation exactly.

4.1.3. Proposition. *For π_1 and π_2 in $PT_M(Y, X)$,*

$$\pi_1 \leq \pi_2 \quad \textit{if and only if} \quad \forall P \subseteq X \, \forall Q \subseteq Y \colon P \subseteq \pi_1(Q) \;\Rightarrow\; P \subseteq \pi_2(Q).$$

Proof: Assume $\pi_1(Q) \subseteq \pi_2(Q)$ for all $Q \subseteq Y$. Then $P \subseteq \pi_1(Q) \subseteq \pi_2(Q)$ implies $P \subseteq \pi_2(Q)$. For the converse, assume the above right hand side holds. Since $\pi_1(Q) \subseteq \pi_1(Q)$ for all $Q \subseteq Y$, $\pi_1(Q) \subseteq \pi_2(Q)$. Hence $\pi_1 \leq \pi_2$. □

Next we show how every monotonic predicate transformer can be described by some primitive monotone predicate transformers together with some constructors on predicate transformers. This gives then the language for the description of the monotonic predicate transformers in the refinement calculus. We first give three collections of primitive predicate transformers.

A subset $V \subseteq X$ can be lifted to the monotonic predicate transformer '$\{V\}$' $\in PT_M(X, X)$ by

$$\{V\}(P) \quad = \quad V \cap P,$$

and also to the monotonic predicate transformer '$V{\rightarrow}$' $\in PT_M(X, X)$ by

$$V{\rightarrow}(P) \quad = \quad \{x \in X \mid x \in V \;\Rightarrow\; x \in P\},$$

for all $P \subseteq X$. The predicate transformers '$\{V\}$' and '$V{\rightarrow}$' are called *assert command* and *guarded command*, respectively. They can be thought of as conditional tests. Note that the predicate transformer '$V{\rightarrow}$' always terminates whereas '$\{V\}$' does not for all inputs x with $x \notin V$.

Every function $f : X \rightarrow Y$ can be lifted to the monotonic predicate transformer '$\langle f \rangle$' $\in PT_M(Y, X)$ by

$$\langle f \rangle(P) \quad = \quad \{x \in X \mid f(x) \in P\},$$

for all $P \subseteq Y$. The predicate transformer $\langle f \rangle$ is called *update command* and can be thought of as a multiple assignment.

Next we look at the predicate transformer constructors: two monotonic predicate transformers $\pi_1 \in PT_M(Z, Y)$ and $\pi_2 \in PT_M(Y, X)$ can be composed by functional composition obtaining the monotonic predicate transformer $\pi_1 \circ \pi_2 \in PT_M(Z, X)$. Thus

$$(\pi_1 \circ \pi_2)(P) \quad = \quad \pi_1(\pi_2(P)),$$

for all $P \subseteq Y$. The above functional composition is also called *sequential composition.*

Finally, from an arbitrary set (possibly empty) of monotonic predicate transformers $\{\pi_i \in PT_M(Y, X) \mid i \in I\}$ two other monotonic predicate transformers can be obtained by applying the meet and the join of the lattice $PT_M(Y, X)$. Although $PT_M(Y, X)$ is not a complete Boolean algebra, it is a complete lattice with meets and joins defined pointwise, exactly as in $PT(Y, X)$. Hence we have

$$\begin{aligned} (\bigwedge\{\pi_i \mid i \in I\})(P) &= \bigcap\{\pi_i(P) \mid i \in I\}, \\ (\bigvee\{\pi_i \mid i \in I\})(P) &= \bigcup\{\pi_i(P) \mid i \in I\}, \end{aligned}$$

for all $P \subseteq Y$. The meet $\bigwedge$ is called *demonic choice* while the join $\bigvee$ is called *angelic choice.*

Besides preserving monotonicity, the above constructors are also monotonic as functions on the lattice of predicate transformers. In general, if z is a variable ranging over monotonic predicate transformers in $PT_M(Y,X)$ and $C(z)$ is a monotonic predicate transformer constructed from the above primitive monotonic predicate transformers, the lattice and functional constructors, and containing the variable z, then

$$\lambda z. C(z) : PT_M(Y,X) \to PT_M(Y,X)$$

is a monotonic function. This means that we may always replace a monotonic predicate transformer by a refined one in any context, because

$$\pi_1 \leq \pi_2 \quad \text{implies} \quad C(\pi_1) \leq C(\pi_2).$$

The following theorem, due to Von Wright [Wri90], shows that every monotonic predicate transformer can be obtained from the primitive predicate transformers, the lattice constructors and the functional composition.

4.1.4. Theorem. *Let $\pi \in PT_M(Y,X)$ and let V_x denote the set $\{x\} \subseteq X$ for $x \in X$. Then π coincides with the predicate transformer*

$$\bigvee\{\{V_x\} \circ \bigwedge\{\langle f : X \to Y\rangle \mid \forall x \in X \colon f(x) \in P\} \mid P \subseteq Y \;\&\; x \in \pi(P)\}.$$

Proof: The proof proceeds in three steps.

1. Let $Q \subseteq Y$. By definition of meets and of the update command,

$$\begin{aligned}&(\bigwedge\{\langle f : X \to Y\rangle \mid \forall x \in X \colon f(x) \in P\})(Q)\\ &= \bigcap\{\langle f : X \to Y\rangle(Q) \mid \forall x \in X \colon f(x) \in P\}\\ &= \bigcap\{\{x \mid f(x) \in Q\} \mid \forall x \in X \colon f(x) \in P\}.\end{aligned}$$

If $P \subseteq Q$ then the above set is clearly X. Otherwise, it is empty. To prove the latter statement let $y \in P \setminus Q$ (which exists because $P \not\subseteq Q$). Consider $f : X \to Y$ such that $f(x) \in P$ for all $x \in X$ (which exists because P is nonempty). If $f(z) \in Q$ for some $z \in X$ define $f_z : X \to Y$ by

$$f_z(x) \quad = \quad \begin{cases} y & \text{if } x = z \\ f(x) & \text{otherwise,} \end{cases}$$

for every $x \in X$. Then $f_z(x) \in P$ for all $x \in X$ but $f_z(z) \notin Q$. It follows that, if $P \not\subseteq Q$

$$\bigcap\{\{x \mid f(x) \in Q\} \mid \forall x \in X \colon f(x) \in P\} = \emptyset.$$

2. Let $\rho_{x,P} = \{V_x\} \circ \bigwedge\{\langle f : X \to Y\rangle \mid \forall x \in X : f(x) \in P\}$. By step 1., and the definitions of the predicate transformer $\{V_x\}$ and of the functional composition,

$$\rho_{x,P}(Q) \quad = \quad \begin{cases} \{x\} & \text{if } P \subseteq Q \\ \emptyset & \text{otherwise,} \end{cases} \tag{4.1}$$

for all $Q \subseteq Y$.

3. For $x \in X$ and $P \subseteq Y$ let $\rho_{x,P}$ be defined as above. For $Q \subseteq Y$,

$$\begin{aligned} &(\bigvee\{\rho_{x,P} \mid P \subseteq Y \ \&\ x \in \pi(P)\})(Q) \\ &= \bigcup\{\rho_{x,P}(Q) \mid P \subseteq Y \ \&\ x \in \pi(P)\} \\ &= \bigcup\{\rho_{x,P}(Q) \mid P \subseteq Q \ \&\ x \in \pi(P)\} \quad [\rho_{x,P}(Q) = \emptyset \text{ if } P \not\subseteq Q] \\ &= \bigcup\{\{x\} \mid P \subseteq Q \ \&\ x \in \pi(P)\} \quad [\text{by (4.1)}] \\ &= \bigcup\{\pi(P) \mid P \subseteq Q\} \\ &= \pi(Q). \quad [\pi \text{ is monotone}] \end{aligned}$$

□

4.2 The language $\mathcal{L}_1$ and its predicate transformer semantics

We now extend the programming language $\mathcal{L}_0$ to a language $\mathcal{L}_1$ with the specification constructs of the refinement calculus. The main difference with the language of the refinement calculus is that we have procedure variables in the language.

4.2.1. Definition. Let $\mathtt{St}$ be a set of states and let $PVar$ be a set of procedure variables.

(i) The class $(S \in)\, Stat_1$ of *statements* is given by

$$S \quad ::= \quad V\!\to \mid \{V\} \mid \langle f\rangle \mid x \mid \bigvee_I S_i \mid \bigwedge_I S_i \mid S\,;S\,,$$

where $V \subseteq \mathtt{St}$, $f : \mathtt{St} \to \mathtt{St}$, $x \in PVar$, and I is an arbitrary set.

(ii) A declaration is a function $d \in Decl_1 = PVar \to Stat_1$.

(iii) A *command* in the language $\mathcal{L}_1$ is a pair $\langle d, S\rangle$, where d is a declaration in $Decl_1$ and S a statement in $Stat_1$.

The language $\mathcal{L}_1$ is a proper class since the index I in the $\bigvee$ and $\bigwedge$ constructs can be any set. One way of circumventing the use of proper classes is to impose a limit (which can be an arbitrary cardinal) on the size of the index sets I that are used in the $\bigvee$ and $\bigwedge$ constructs. We can then form an inductive hierarchy of syntactic terms indexed by the ordinals. By fixing a regular cardinal κ which is larger than the cardinalities of the set of states, of the set of procedure variables, and of the limit imposed on the index sets of $\bigvee$ and $\bigwedge$, then it is straightforward to show that the cardinality of $\mathcal{L}_1$ is bounded by κ. For more details on this kind of arguments, see [MRS95].

The language $\mathcal{L}_0$ of Definition 3.1.1 can be mapped into $\mathcal{L}_1$ via the translation function $(\cdot)^\dagger{:}Stat_0 \to Stat_1$ defined inductively by

$$\begin{array}{lcl}
(v\ :=\ e)^\dagger & = & \langle \lambda s \in \mathtt{St}.s[\mathcal{E}_{\mathcal{V}}(e)(s)/v]\rangle, \\
(b{\rightarrow})^\dagger & = & \mathcal{B}_{\mathcal{V}}(b){\rightarrow}, \\
(x)^\dagger & = & x, \\
(S_1\ ;\ S_2)^\dagger & = & (S_1)^\dagger\ ;\ (S_2)^\dagger, \\
(S_1 \,\square\, S_2)^\dagger & = & (S_1)^\dagger \wedge (S_2)^\dagger.
\end{array}$$

where $v \in IVar$, $e \in Exp$, $b \in BExp$, and $x \in PVar$. The mapping $(\cdot)^\dagger$ can be extended to programs in $\mathcal{L}_0$ by

$$(\langle d, S\rangle)^\dagger = \langle d^\dagger, (S)^\dagger\rangle, \quad \text{where, for all } x \in PVar, \quad d^\dagger(x) = (d(x))^\dagger.$$

Notice that $d^\dagger(x) \in Stat_1$ for every $x \in PVar$ and $d \in Decl_0$.

The semantics of $\mathcal{L}_1$ can be given by associating to every command in $\mathcal{L}_1$ a predicate transformer in $PT_M(\mathtt{St},\mathtt{St})$.

4.2.2. Definition. Put $(\xi \in)\, PTEnv = PVar \to PT_M(\mathtt{St},\mathtt{St})$.

(i) The map $Pt : Stat_1 \to (PTEnv \to PT_M(\mathtt{St},\mathtt{St}))$ is given inductively by

$$\begin{array}{lcl}
Pt(V{\rightarrow})(\xi) & = & V{\rightarrow}, \\
Pt(\{V\})(\xi) & = & \{V\}, \\
Pt(\langle f\rangle)(\xi) & = & \langle f\rangle, \\
Pt(x)(\xi) & = & \xi(x) \\
Pt(\bigvee_I S_i)(\xi) & = & \bigvee\{Pt(S_i)(\xi) \mid i \in I\}, \\
Pt(\bigwedge_I S_i)(P) & = & \bigwedge\{Pt(S_i)(\xi) \mid i \in I\}, \\
Pt(S_1\ ;\ S_2)(\xi) & = & Pt(S_1)(\xi) \circ Pt(S_2)(\xi).
\end{array}$$

(ii) For every declaration $d : Decl_1$ define $\Xi_d : PTEnv \to PTEnv$ by
$\Xi_d(\xi)(x) \;=\; Pt(d(x))(\xi)$.

(iii) The semantics $Wp_1[\![\cdot]\!] : \mathcal{L}_1 \to PT_M(\mathtt{St},\mathtt{St})$ is given by
$Wp_1[\![\langle d, S\rangle]\!] \;=\; Pt(S)(\xi_d)$,
where ξ_d is the least fixed point of Ξ_d.

Monotonicity of Ξ_d can be checked as follows. It is based on the monotonicity of the corresponding predicate transformer constructors. Since $PT_M(\mathtt{St},\mathtt{St})$ is a complete lattice, $PTEnv$ is a complete lattice too. Therefore the function Ξ_d has a least fixed point by Proposition 2.2.1 (and also by Proposition 2.2.3), say ξ_d. Note that this means that ξ_d is the least environment such that $\xi_d(x) = Pt(d(x))(\xi_d)$.

The semantics $Wp_1[\![\cdot]\!]$ is a fixed point semantics in the sense that the meaning of a procedure variable is equal to the meaning of its declaration:

$$\begin{aligned} Wp_1[\![\langle d, x\rangle]\!] &= Pt(x)(\xi_d) \quad \text{[by definition of } Pt[\![\cdot]\!]] \\ &= \xi_d(x) \quad \text{[by definition of } Pt(\cdot)] \\ &= \Xi_d(\xi_d)(x) \quad [\xi_d \text{ is a fixed point of } \Xi_d] \\ &= Pt(d(x))(\xi_d) \quad \text{[by definition of } \Xi_d] \\ &= Wp_1[\![\langle d, d(x)\rangle]\!]. \quad \text{[by definition of } Wp_1[\![\cdot]\!]] \end{aligned}$$

Furthermore, $Wp_1[\![\cdot]\!]$ is the least among all 'reasonable' fixed point semantics of $\mathcal{L}_1$. This is shown in the next lemma.

4.2.3. Lemma. *Let $F : \mathcal{L}_1 \to PT_M(\mathtt{St}, \mathtt{St})$ be a function such that*

$$\begin{aligned} F(\langle d, V{\rightarrow}\rangle) &= V{\rightarrow}, \\ F(\langle d, \{V\}\rangle) &= \{V\}, \\ F(\langle d, \langle f\rangle\rangle) &= \langle f\rangle, \\ F(\langle d, x) &= F(\langle d, d(x)\rangle) \\ F(\langle d, \textstyle\bigvee_I S_i\rangle) &= \textstyle\bigvee\{F(\langle d, S_i\rangle) \mid i \in I\}, \\ F(\langle d, \textstyle\bigwedge_I S_i\rangle)(P) &= \textstyle\bigwedge\{F(\langle d, S_i\rangle) \mid i \in I\}, \\ F(\langle d, S_1\,;\, S_2\rangle) &= F(\langle d, S_1\rangle) \circ F(\langle d, S_2\rangle). \end{aligned}$$

Then $WP_1[\![\langle d, S\rangle]\!] \leq F(\langle d, S\rangle)$ for all $\langle d, S\rangle \in \mathcal{L}_1$.

Proof: For an arbitrary but fixed declaration $d \in Decl_1$ define the environment $\xi \in PTEnv$ by $\xi(x) = F(\langle d, d(x)\rangle)$. By induction on the structure of S it is easy to see that

$$F(\langle d, S\rangle) = Pt(S)(\xi). \tag{4.2}$$

For example, if $S \equiv x$ then

$$F(\langle d, x\rangle) = F(\langle d, d(x)\rangle) = \xi(x) = Pt(x)(\xi).$$

Next we prove that the environment ξ is a fixed point of Ξ_d: for every $x \in PVar$,

$$\begin{aligned} \Xi_d(\xi)(x) &= Pt(d(x))(\xi) \\ &= F(d(x)) \quad \text{[Equation (4.2)]} \\ &= \xi(x). \quad [\text{Definition of } \xi] \end{aligned}$$

Using again induction on the structure of the statement S we can finally prove that $Wp_1[\![\langle d, S\rangle]\!] \leq F(\langle d, S\rangle)$ for every $\langle d, S\rangle \in \mathcal{L}_1$. We treat here only the case of procedure variables.

Since ξ_d is the least fixed point of Ξ_d we have $\xi_d(x) \leq \xi(x)$ for every $x \in PVar$. Therefore

$$\begin{array}{rcl} Wp_1[\![\langle d, x\rangle]\!] & = & Pt(x)(\xi_d) \quad [\text{Definition of } Wp_1[\![\cdot]\!]] \\ & = & \xi_d(x) \quad [\text{Definition of } Pt(\cdot)] \\ & \leq & \xi(x) \quad [\xi_d \text{ is the least fixed point of } \Xi_d] \\ & = & F(\langle d, d(x)\rangle) \quad [\text{Definition of } \xi] \\ & = & F(\langle d, x\rangle). \end{array}$$

□

A similar argument to the one used in the above proof can be used to prove that the $Wp_1[\![\cdot]\!]$ semantics is a total correctness semantics which extends conservatively the weakest precondition semantics of $\mathcal{L}_0$.

4.2.4. Theorem. *For every $\langle d, S\rangle \in \mathcal{L}_0$, $Wp_0[\![\langle d, S\rangle]\!] = Wp_1[\![(\langle d, S\rangle)^\dagger]\!]$.*

Proof: By induction on the structure of S, it is easy to see that $Wp_1[\![(\langle d, S\rangle)^\dagger]\!]$ is a total correctness predicate transformer in $PT_T(\mathtt{St}, \mathtt{St})$ for all $\langle d, S\rangle \in \mathcal{L}_0$. Moreover $Wp_1[\![(\cdot)^\dagger]\!]$ is a fixed point of the function Ψ_T defined in Lemma 3.3.3. Since $Wp_0[\![\cdot]\!]$ is the least fixed point of Ψ_T,

$$Wp_0[\![\langle d, S\rangle]\!] \leq Wp_1[\![(\langle d, S\rangle)^\dagger]\!]$$

for all $\langle d, S\rangle \in \mathcal{L}_0$.

Conversely, first note by induction on the structure of S that, for all $\langle d, S\rangle \in \mathcal{L}_0$, $Wp_0[\![\langle d, S\rangle]\!] = Pt((S)^\dagger)(\xi)$, where $\xi(x) = Wp_0[\![\langle d, d(x)\rangle]\!]$. It follows that ξ is a fixed point of Ξ_d and hence

$$Wp_1[\![(\langle d, S\rangle)^\dagger]\!] = Pt((S)^\dagger)(\xi_d) \leq Pt((S)^\dagger)(\xi) = Wp_0[\![\langle d, S\rangle]\!],$$

for all $\langle d, S\rangle \in \mathcal{L}_0$. □

It is natural to define a *refinement relation* on commands of $\mathcal{L}_1$ by putting, for $\langle d, S_1\rangle, \langle d, S_2\rangle$ in $\mathcal{L}_1$:

$$\langle d, S_1\rangle \precsim \langle d, S_2\rangle \quad \text{if and only if} \quad Wp_1[\![(\langle d, S_1\rangle)]\!] \leq Wp_1[\![(\langle d, S_2\rangle)]\!].$$

In this case we say that $\langle d, S_1\rangle$ is refined by $\langle d, S_2\rangle$, since every total correctness property satisfied by $\langle d, S_1\rangle$ is satisfied also by $\langle d, S_2\rangle$. Hence the $Wp_1[\![\cdot]\!]$ semantics identifies specification commands on the basis of the satisfied total correctness properties.

4.3 A state transformer semantics for $\mathcal{L}_1$

We now look for a forward denotational semantics for the specification language $\mathcal{L}_1$. We want a semantic domain of state transformers which is isomorphic to the domain of monotonic predicate transformers. Because of the possibility of arbitrary meets and joins of commands in $\mathcal{L}_1$ the simpler domains introduced in the previous chapter (or variations thereof) will not work. We take as domain the free completely distributive lattice over X.

4.3.1. Definition. Let X be a set. Define the *free completely distributive lattice* over X, denoted by $CDL(X)$, to be the collection of all lower closed subsets of the complete lattice $L = (\mathcal{P}(X), \supseteq)$. Elements of $CDL(X)$ are ordered by subset inclusion.

Clearly $CDL(X)$ is a partial order with $\emptyset$ as least element (which will be used for denoting a non-terminating computation), and the set of all subsets of X as top element (which will be used for denoting deadlocking computations). Since $CDL(X)$ is closed under arbitrary unions and arbitrary intersections, it is a complete sub-lattice of $\mathcal{P}(\mathcal{P}(X))$. Hence $CDL(X)$ is a completely distributive lattice. In Chapter 9 we will discuss some lattice theoretical properties of $CDL(X)$, proving, for example, in Theorem 9.1.3 that $CDL(X)$ is indeed the free completely distributive lattice over X.

Using the above definition we can define the semantic domain $ST(X, Y)$.

4.3.2. Definition. The domain of *state transformers for specification* from a set X to Y is the set $X \to CDL(Y)$ ordered by the pointwise extension of the order of $CDL(Y)$. It is denoted by $ST(X, Y)$ with σ, τ as typical elements.

Before proving that the above domain of state transformers is equivalent the domain of the predicate transformers, we give some motivation for the definition. A function σ in $ST(X, Y)$ denotes the specification of a class of commands. It assigns to every input state $x \in X$ the collection $\sigma(x)$ of all predicates on the output space Y which must be satisfied by every computation started in x of every command specified by σ. This implies that every computation started in x of every command specified by σ must terminate (hence no special symbol $\perp$ to record non-termination is required). The set $\sigma(x)$ is maximal in the sense that it is upper closed because if every computation of a command specified by σ at input x terminates and satisfies a predicate $P \in \sigma(x)$, then it satisfies also predicates Q with $Q \supseteq P$.

If there is a computation starting in x that fails to terminate then $\sigma(x) = \emptyset$. If every computation of a command started at x deadlocks, then no output in Y is obtained and hence every predicate in Y is satisfied. Hence $\sigma(x) = \{P \mid P \subseteq Y\}$ specifies commands which starting from input x always deadlock.

The relationship between state transformers and predicate transformers is the content of the following theorem.

4.3.3. Theorem. *Let X and Y be two sets. There is an order-isomorphism between*

$$X \to \mathcal{P}(\mathcal{P}(Y)) \quad \textit{and} \quad \mathcal{P}(Y) \to \mathcal{P}(X).$$

The isomorphism is given by the functions

$$\hat{\omega}(\sigma)(P) = \{x \in X \mid P \in \sigma(x)\} \quad \textit{and} \quad \hat{\omega}^{-1}(\pi)(x) = \{P \subseteq Y \mid x \in \pi(P)\},$$

for $\sigma : X \to \mathcal{P}(\mathcal{P}(Y))$, $\pi : \mathcal{P}(Y) \to \mathcal{P}(X)$, $x \in X$, and $P \subseteq Y$. Furthermore, it restricts and co-restricts to an order-isomorphism between $ST(X,Y)$ and $PT_M(Y,X)$.

Proof: The function $\hat{\omega}^{-1}$ is a right inverse of $\hat{\omega}$ because, for $x \in X$,

$$\hat{\omega}^{-1}(\hat{\omega}(\sigma))(x) = \{P \mid x \in \hat{\omega}(\sigma)(P)\} = \{P \mid P \in \sigma(x)\} = \sigma(x).$$

Similarly, $\hat{\omega}^{-1}$ is a left inverse of $\hat{\omega}$ because, for $P \subseteq Y$

$$\hat{\omega}(\hat{\omega}^{-1}(\pi))(P) = \{x \mid P \in \hat{\omega}^{-1}(\pi)(x)\} = \{x \mid x \in \pi(P)\} = \pi(P).$$

Next we show that the isomorphism is order preserving. Assume $\sigma_1(x) \subseteq \sigma_2(x)$ for every $x \in X$. Then $P \in \sigma_1(x)$ implies $P \in \sigma_2(x)$ for all $P \subseteq Y$ and therefore $\hat{\omega}(\sigma_1)(P) \subseteq \hat{\omega}(\sigma_2)(P)$.

Conversely, if $\pi_1(P) \subseteq \pi_2(P)$ for all $P \subseteq Y$ then $x \in \pi_1(P)$ implies $x \in \pi_2(P)$ and therefore $\hat{\omega}^{-1}(\pi_1)(x) \subseteq \hat{\omega}^{-1}(\pi_2)(x)$ for all $x \in X$.

Finally we show that the isomorphism restricts and co-restricts to an order-isomorphism between $ST(X,Y)$ and $PT_M(Y,X)$. Let $\sigma \in ST(X,Y)$. If $P \subseteq Q \subseteq Y$ then

$$\hat{\omega}(\sigma)(P) = \{x \mid P \in \sigma(x)\} \subseteq \{x \mid Q \in \sigma(x)\} = \hat{\omega}(\sigma)(Q).$$

Hence $\hat{\omega}(\sigma)$ is monotone. For the converse, let π be a monotonic predicate transformer in $PT_M(Y,X)$. For every $x \in X$, if $P \in \hat{\omega}^{-1}(\pi)(x)$ and $P \subseteq Q$ then $x \in \pi(Q)$ because $x \in \pi(P)$ and π is monotone. Thus $Q \in \hat{\omega}^{-1}(\pi)(x)$. Therefore $\hat{\omega}^{-1}(\pi) \in PT_M(Y,X)$. $\square$

The predicate $\hat{\omega}(\sigma)(P)$ can be thought of as the weakest precondition associated with the function σ and the postcondition P. Indeed $x \in \hat{\omega}(\sigma)(P)$ exactly when every computation of a program specified by σ for input x terminates in a state satisfying P.

Next we give some constructors on $ST(X,Y)$. Since $CDL(X)$ is a completely distributive lattice, also $ST(X,Y)$ is completely distributive: meets and joins are defined pointwise. Indeed, if $\{\sigma_i \mid i \in I\}$ is an arbitrary set of functions in $ST(X,Y)$ then, for $x \in X$,

$$\begin{aligned} (\bigwedge\{\sigma_i \mid i \in I\})(x) &= \bigcap\{\sigma_i(x) \mid i \in I\}, \\ (\bigvee\{\sigma_i \mid i \in I\})(x) &= \bigcup\{\sigma_i(x) \mid i \in I\}. \end{aligned}$$

A function $\sigma_1 \in ST(X, Y)$ can be composed with $\sigma_2 \in ST(Y, Z)$ as follows. For $x \in X$,

$$(\sigma_1 ; \sigma_2)(x) \;=\; \bigcup\{\bigcap\{\sigma_2(y) \mid y \in P\} \mid P \in \sigma_1(x)\}. \tag{4.3}$$

Well-definedness of these three operations can be easily verified. The ';' operation can intuitively be explained as follows.

Assume every computation specified by σ_1 started at input x terminates and satisfies a predicate P in $\sigma_1(x)$. Next assume that every computation started at $y \in P$ terminates satisfying a predicate Q_y in $\sigma_2(y)$. Then every computation of the combined commands started at x terminates and is guaranteed to satisfy every Q_y for $y \in P$.

4.3.4. Lemma. *Let $\pi_1 \in PT_M(Y, X)$, and $\pi_2 \in PT_M(Z, Y)$. Let also $\{\pi_i \mid i \in I\}$ be a set of monotonic predicate transformers in $PT_M(Y, X)$. Then*
(i) $\hat{\omega}^{-1}(\bigwedge_I \pi_i) = \bigwedge_I \hat{\omega}^{-1}(\pi_i)$,
(ii) $\hat{\omega}^{-1}(\bigvee_I \pi_i) = \bigvee_I \hat{\omega}^{-1}(\pi_i)$,
(iii) $\hat{\omega}^{-1}(\pi_1 \circ \pi_2) = \hat{\omega}^{-1}(\pi_1) ; \hat{\omega}^{-1}(\pi_2)$.

Proof: We start by proving the first item. For every $x \in X$ we have:

$$\begin{aligned}
&\hat{\omega}^{-1}(\textstyle\bigwedge_I \pi_i)(x)\\
&\quad= \{P \subseteq Y \mid x \in (\textstyle\bigwedge_I \pi_i)(P)\}\\
&\quad= \{P \subseteq Y \mid x \in \textstyle\bigcap_I \pi_i(P)\}\\
&\quad= \textstyle\bigcap_I \{P \subseteq Y \mid x \in \pi_i(P)\}\\
&\quad= \textstyle\bigcap_I \hat{\omega}^{-1}(\pi_i)(x).
\end{aligned}$$

The second item can be proved in a similar way. It remains to prove the last item. For $x \in X$,

$$\begin{aligned}
&\hat{\omega}^{-1}(\pi_1 \circ \pi_2)(x)\\
&\quad= \{P \subseteq Z \mid x \in (\pi_1 \circ \pi_2)(P)\}\\
&\quad= \{P \subseteq Z \mid x \in \pi_1(\pi_2(P))\}\\
&\quad\overset{*}{=} \{P \subseteq Z \mid \exists Q \subseteq Y \colon x \in \pi_1(Q) \;\&\; \forall y \in Q \colon y \in \pi_2(P)\}\\
&\quad= \{P \subseteq Z \mid \exists Q \in \hat{\omega}^{-1}(\pi_1)(x) \colon \forall y \in Q \colon P \in \hat{\omega}^{-1}(\pi_2)(y)\}\\
&\quad= \textstyle\bigcup\{\bigcap\{\hat{\omega}^{-1}(\pi_2)(y) \mid y \in Q\} \mid Q \in \hat{\omega}^{-1}(\pi_1)(x)\}\\
&\quad= (\hat{\omega}^{-1}(\pi_1) ; \hat{\omega}^{-1}(\pi_2))(x),
\end{aligned}$$

where $\overset{*}{\subseteq}$ trivially holds if we take $Q = \pi_2(P)$. Conversely, let $P \subseteq Z$ such that there exists $Q \subseteq Y$ with $x \in \pi_1(Q)$ and $y \in \pi_2(P)$ for all $y \in Q$. Then $Q \subseteq \pi_2(P)$. Hence, by monotonicity of π_1, we have $x \in \pi_1(Q) \subseteq \pi_1(\pi_2(P))$ implies $x \in \pi_1(\pi_2(P))$. It follows that $P \in \{V \subseteq Z \mid x \in \pi_1(\pi_2(V))\}$. $\square$

By Theorem 4.3.3 and the above lemma it is immediate that

(i) $\hat{\omega}(\bigwedge_I \sigma_i) = \bigwedge_I \hat{\omega}(\sigma_i)$,
(ii) $\hat{\omega}(\bigvee_I \sigma_i) = \bigvee_I \hat{\omega}(\sigma_i)$,
(iii) $\hat{\omega}(\sigma_1 \,;\, \sigma_2) = \hat{\omega}(\sigma_1) \circ \hat{\omega}(\sigma_2)$.

We can now give a forward denotational semantics for $\mathcal{L}_1$. We proceed as for the predicate transformer semantics by using environments to record the meanings of procedure variables.

4.3.5. Definition. Put $(\zeta \in)\, STEnv = PVar \to ST(\mathtt{St}, \mathtt{St})$.

(i) The map $St : Stat_1 \to (STEnv \to ST(\mathtt{St}, \mathtt{St}))$ is given inductively by

$$\begin{array}{lcl}
St(V{\rightarrow})(\zeta)(s) & = & \{P \subseteq \mathtt{St} \mid s \in V \;\Rightarrow\; s \in P\}, \\
St(\{V\})(\zeta)(s) & = & \{P \subseteq \mathtt{St} \mid s \in V \cap P\}, \\
St(\langle f \rangle)(\zeta)(s) & = & \{P \subseteq \mathtt{St} \mid f(s) \in P\}, \\
St(x)(\zeta) & = & \zeta(x), \\
St(\bigvee_I S_i)(\zeta) & = & \bigvee\{St(S_i)(\zeta) \mid i \in I\}, \\
St(\bigwedge_I S_i)(\zeta) & = & \bigwedge\{St(S_i)(\zeta) \mid i \in I\}, \\
St(S_1 \,;\, S_2)(\zeta) & = & St(S_1)(\zeta) \,;\, St(S_2)(\zeta).
\end{array}$$

(ii) For every declaration $d : Decl_1$ define $\mathrm{H}_d : STEnv \to STEnv$ by
$$\mathrm{H}_d(\zeta)(x) \;=\; St(d(x))(\zeta).$$

(iii) The semantics $St[\![\cdot]\!] : \mathcal{L}_1 \to ST(\mathtt{St}, \mathtt{St})$ is given by
$$St[\![\langle d, S\rangle]\!] \;=\; St(S)(\zeta_d),$$
where ζ_d is the least fixed point of H_d.

The transformation $\mathrm{H}_d : STEnv \to STEnv$ is monotone. Since $STEnv$ is a complete lattice, H_d has a least fixed point. Hence the semantics $St[\![\cdot]\!]$ is well-defined. Below we prove that it is isomorphic to the predicate transformer semantics $Wp_1[\![\cdot]\!]$.

4.3.6. Theorem. *For every* $\langle d, S\rangle \in \mathcal{L}_1$,

$$\hat{\omega}(St[\![\langle d, S\rangle]\!]) = Wp_1[\![\langle d, S\rangle]\!] \quad \textit{and} \quad \hat{\omega}^{-1}(Wp_1[\![\langle d, S\rangle]\!]) = St[\![\langle d, S\rangle]\!].$$

Proof: By Theorem 4.3.3 $\lambda x.\hat{\omega}(\zeta(x)) \in PTEnv$ for all $\zeta \in STEnv$. Next we prove by structural induction on S, and using Lemma 4.3.4, that

$$\hat{\omega}(St(S)(\zeta)) \;=\; Pt(S)(\lambda x.\hat{\omega}(\zeta(x))). \tag{4.4}$$

We treat only two cases. If $S \equiv x$ then

$$\hat{\omega}(St(x)(\zeta)) = \hat{\omega}(\zeta(x)) = Pt(x)(\lambda x.\hat{\omega}(\zeta(x))).$$

If $S \equiv S_1 \,;\, S_2$ then

$$\hat{\omega}(St(S_1 \,;\, S_2)(\zeta))$$

$$= \hat\omega(St(S_1)(\zeta)\ ;\ St(S_2)(\zeta)) \quad [\text{Definition of } St(\cdot)(\zeta)]$$
$$= \hat\omega(St(S_1)(\zeta)) \circ \hat\omega(St(S_1)(\zeta)) \quad [\text{Lemma 4.3.4}]$$
$$= Pt(S_1)(\lambda x.\hat\omega(\zeta(x))) \circ Pt(S_2)(\lambda x.\hat\omega(\zeta(x))) \quad [\text{induction hypothesis}]$$
$$= Pt(S_1\ ;\ S_2)(\lambda x.\hat\omega(\zeta(x)). \quad [\text{Definition of } Pt(\cdot)(\lambda x.\hat\omega(\zeta(x)))]$$

Next we characterize the least fixed point of Ξ_d, for a declaration $d : PVar \to GStat_1$, in terms of the least fixed point of H_d using the isomorphism $\hat\omega$. First we see that for every $\zeta \in STEnv$ and declaration d,

$$\hat\omega(\lambda x.\mathrm{H}_d(\zeta)(x))$$
$$= \hat\omega(\lambda x.St(d(x))(\zeta)) \quad [\text{Definition } \mathrm{H}_d]$$
$$= \lambda x.Pt(d(x))(\lambda x.\hat\omega(\zeta(x)) \quad [\text{by 4.4}]$$
$$= \Xi_d(\lambda x.\hat\omega(\zeta(x)). \quad [\text{Definition } \Xi_d]$$

Hence, by Proposition 2.2.5 the least fixed point of Ξ_d is $\hat\omega(\lambda x.\zeta_d(x))$, where ζ_d is the least fixed point of H_d.

We finally prove that the state transformer semantics and the predicate transformer semantics of $\mathcal{L}_1$ are isomorphic. For all $\langle d, S\rangle \in \mathcal{L}_1$,

$$\hat\omega(St[\![\langle d, S\rangle]\!])$$
$$= \hat\omega(St(S)(\zeta_d)) \quad [\text{Definition of } St[\![\cdot]\!]]$$
$$= Pt(S)(\lambda x.\hat\omega(\zeta_d(x))) \quad [\text{by 4.4}]$$
$$= Wp_1[\![\langle d, S\rangle]\!]. \quad [\text{Def. of } Wp_1[\![\cdot]\!],\ \lambda x.\hat\omega(\zeta_d(x)) \text{ least fixed point of } \Xi_d]$$

By Theorem 4.3.3 and the above, $\hat\omega^{-1}(Wp_1[\![\langle d, S\rangle]\!]) = St[\![\langle d, S\rangle]\!]$. □

As an immediate consequence of the above theorem and Lemma 4.2.3 we have the following corollary.

4.3.7. Corollary. *The function $St[\![\cdot]\!]$ is the least among all the functions $F : \mathcal{L}_1 \to ST(\mathtt{St}, \mathtt{St})$ such that*

$$\begin{array}{lcl}
F(\langle d, V\!\rightarrow\rangle)(s) & = & \{P \subseteq \mathtt{St} \mid s \in V \ \Rightarrow\ s \in P\},\\
F(\langle d, \{V\}\rangle)(s) & = & \{P \subseteq \mathtt{St} \mid s \in V \cap P\},\\
F(\langle d, \langle f\rangle\rangle)(s) & = & \{P \subseteq \mathtt{St} \mid f(s) \in P\},\\
F(\langle d, x\rangle)(s) & = & F(\langle d, d(x)\rangle)(s),\\
F(\langle d, \bigvee_I S_i\rangle)(s) & = & \bigcup\{F(\langle d, S_i\rangle)(s) \mid i \in I\},\\
F(\langle d, \bigwedge_I S_i\rangle)(s) & = & \bigcap\{F(\langle d, S_i\rangle) \mid i \in I\},\\
F(\langle d, S_1\ ;\ S_2\rangle)(s) & = & (F(\langle d, S_1\rangle)\ ;\ F(\langle d, S_2\rangle))(s),
\end{array}$$

for every $s \in \mathtt{St}$. □

4.4 An operational semantics for $\mathcal{L}_1$

In this section we give an operational semantics for $\mathcal{L}_1$, and prove it equivalent to the forward semantics. The operational semantics is based on hyper transition systems, which are a generalization of standard transition systems.

Transition systems and hyper transition systems

Before we introduce hyper transition systems, we first discuss transition systems. They are a useful mathematical structure to describe the atomic steps of a computation of a program [Plo81b].

4.4.1. Definition. A *transition system* with deadlock is a tuple $\langle X, \delta, \longrightarrow \rangle$ where X is the set of all proper configurations for a program, $\delta \notin X$ denotes a deadlock configuration, and $\longrightarrow \subseteq X \times (X \cup \{\delta\})$ is a transition relation.

The idea is that configurations represent states of a computation, whereas a transition $x \longrightarrow y$ (read 'x goes to y') indicates a possible atomic step which a computation can do, changing the configuration x into the configuration y. If $x \longrightarrow \delta$ then the computation in the configuration x may deadlock. If there is no $y \in X \cup \{\delta\}$ such that $x \longrightarrow y$ then the computation is undefined in the configuration x.

Let us now be a bit more precise about what we mean by 'computation'. Let $T = \langle X, \delta, \longrightarrow \rangle$ be a transition system and $x \in X$. Define a *finite computation* of T starting at x to be a finite sequence $(x_n)_{n \leq k}$ in $X \cup \{\delta\}$ such that

(i) $x = x_0$,
(ii) $x_n \longrightarrow x_{n+1}$ for all $n < k$, and
(iii) for all $y \in X \cup \{\delta\}$ there is no transition $x_k \longrightarrow y$ in T.

If $(x_n)_{n \leq k}$ is a finite computation of T starting at x_0 then we say that it terminates in the configuration x_k. Notice that x_k may also be equal to δ. Not every computation of a program need to be finite. An *infinite computation* of T starting at x is a countable sequence $(x_n)_{n \in \mathbb{N}}$ in $X \cup \{\delta\}$ such that

(i) $x = x_0$, and
(ii) $x_n \longrightarrow x_{n+1}$ for all $n \in \mathbb{N}$.

In general, a *computation* of a transition system T is a finite or infinite computation of T. In other words, a computation of T is a transition sequence of T that cannot be extended.

The next step is to introduce hyper transition systems. Hyper transition systems occur under the name of AND/OR graphs or hyper-graphs in logic programming and artificial intelligence [Nil82].

4.4.2. Definition. A *hyper transition system* is a pair $H = \langle X, \longrightarrow\!\!\!\!\in \rangle$ where X is the collection of all possible configurations in which a computation is allowed to work, and $\longrightarrow\!\!\!\!\in \subseteq X \times \mathcal{P}(X)$ is a transition relation which specifies the atomic steps of a computation.

A hyper transition system specifies a set of computations by specifying their atomic steps. The idea is that a computation specified by an hyper transition system $H = \langle X, \mathrel{-\!\!\!-\!\!\!\in} \rangle$ can change a configuration x into a configuration y if the configuration y satisfies all predicates $W \subseteq X$ such that $x \mathrel{-\!\!\!-\!\!\!\in} W$ (read 'x goes into W'). More formally, the set of all computations specified by a hyper transition system H can be modeled by the following transition system $TS(H)$.

4.4.3. Definition. For a hyper transition system $H = \langle X, \mathrel{-\!\!\!-\!\!\!\in} \rangle$ define the induced transition system $TS(H) = \langle X, \delta, \longrightarrow \rangle$ by

$$\begin{aligned} x \longrightarrow \delta \quad &\Leftrightarrow \quad \bigcap \{ W \mid x \mathrel{-\!\!\!-\!\!\!\in} W \} = \emptyset, \\ x \longrightarrow y \quad &\Leftrightarrow \quad (\exists\, W{:}x \mathrel{-\!\!\!-\!\!\!\in} W) \;\&\; y \in \bigcap \{ W \mid x \mathrel{-\!\!\!-\!\!\!\in} W \}, \end{aligned}$$

for all $x, y \in X$.

A computation of $TS(H)$ (or, equivalently, a computation that satisfies the specification of the hyper transition system H) in a configuration x has three possibilities:

1. it goes to a deadlock configuration because there is no configuration $y \in X$ satisfying all predicates $W \subseteq X$ such that $x \mathrel{-\!\!\!-\!\!\!\in} W$;
2. it is undefined because there is no predicate $W \subseteq X$ such that $x \mathrel{-\!\!\!-\!\!\!\in} W$;
3. it goes to a configuration y satisfying all predicates $W \subseteq X$ such that $x \mathrel{-\!\!\!-\!\!\!\in} W$.

Observe that, by definition, exactly one of the above three possibilities is possible. Indeed, for every $x \in X$, if $x \mathrel{-\!\!\!-\!\!\!\in} W$ then either $x \longrightarrow \delta$ or there exists $y \in W$ such that $x \longrightarrow y$. Conversely, there exists $W \subseteq X$ such that $x \mathrel{-\!\!\!-\!\!\!\in} W$ only if either $x \longrightarrow \delta$ or $x \longrightarrow y$ (and in this case $y \in W$). It follows that a computation specified by a hyper transition system H is undefined in a configuration x if and only if there is no $W \subseteq X$ such that $x \mathrel{-\!\!\!-\!\!\!\in} W$.

As an example of a hyper transition system consider $H = \langle \mathbb{N}, \mathrel{-\!\!\!-\!\!\!\in} \rangle$, where $\mathbb{N}$ is the set of natural numbers and $\mathrel{-\!\!\!-\!\!\!\in}$ is defined by

$$n \mathrel{-\!\!\!-\!\!\!\in} W \quad \Leftrightarrow \quad \forall m > 0 \colon m < n \;\Rightarrow\; m \in W.$$

The configuration '0' is the only configuration in H such that there is no $W \subseteq X$ with $x \mathrel{-\!\!\!-\!\!\!\in} W$. Two of the many computations specified by H are

$$10 \longrightarrow 9 \longrightarrow 4 \longrightarrow 2 \longrightarrow 1 \longrightarrow 0 \quad \text{and} \quad 10 \longrightarrow 7 \longrightarrow 1 \longrightarrow 0\,.$$

It is not hard to see that every computation specified by H is finite and terminates in the configuration '0'.

Under the above interpretation of hyper transition systems it is natural to require that the transition relation $\relbar\joinrel\in$ is *upper closed on the right hand side*, that is,

$$x \relbar\joinrel\in V \;\&\; V \subseteq W \quad \text{implies} \quad x \relbar\joinrel\in W. \tag{4.5}$$

Essentially, the above closure property is due to the fact that $V \subseteq W$ if and only if $V = V \cap W$. No extra information is added by upper closing to the right the transition relation of a hyper transition system.

Observe that hyper transition systems specify computations at the level of the properties that an atomic step has to satisfy, whereas transition systems specify computations at the level of the configurations that an atomic step may reach. Because of this difference a hyper transition system $H = \langle X, \relbar\joinrel\in \rangle$ can model two different kinds of non-determinism: one at the level of the computations specified and one at the level of the specification. The non-determinism of the computations specified by H in a configuration x depends on all the sets $W \subseteq X$ such that $x \relbar\joinrel\in W$: the bigger these sets, the more computations are specified. The non-determinism of the specification depends on the number of transitions starting from the same configuration: the more a specification is non-deterministic the less is the number of computations that it specifies.

Consider the following two examples.

(i) Let $X = \{0, 1, 2\}$ be a set of configurations and consider the hyper transition system $H_1 = \langle X, \relbar\joinrel\in_1 \rangle$ with $0 \relbar\joinrel\in_1 V$ if both 0 and 1 are in V. Then H_1 specifies two computations: they are undefined in a configuration different from 0, but in the configuration 0 one computation does not change configuration, whereas the other one changes 0 to 1. In other words, the transition relation of the induced transition system $TS(H_1)$ is defined by $0 \longrightarrow 0$ and $0 \longrightarrow 1$.

(ii) Let now $H_2 = \langle X, \relbar\joinrel\in_2 \rangle$ be a hyper transition system with $0 \relbar\joinrel\in_2 V$ if either both 0 and 1 are in V or both 0 and 2 are in V. Then only one of the computations of H_1 is specified by H_2, namely the one which does not change the configuration 0. Indeed, the only transition in $TS(H_2)$ is $0 \longrightarrow 0$.

In the next subsection we will see that the non-determinism of the specification is related to angelic non-determinism, and the non-determinism of the computations is related to the demonic non-determinism.

First we compare hyper transition systems to transition systems. We have already seen that a hyper transition system H induces a transition system $TS(H)$ representing all the computations specified by H. However, different hyper transition systems can specify the same sets of computations. Let $X = \{0, 1\}$ and consider two hyper transition systems $H_1 = \langle X, \relbar\joinrel\in_1 \rangle$ and $H_2 = \langle X, \relbar\joinrel\in_2 \rangle$ with

$$\begin{array}{ll} 0 \relbar\joinrel\in_1 V & \text{if } 0 \in V \text{ or } 1 \in V\text{; and} \\ 0 \relbar\joinrel\in_2 V & V \subseteq X. \end{array}$$

Then $TS(H_1) = TS(H_2) = \langle X, \delta, \longrightarrow \rangle$ with $0 \longrightarrow \delta$.

Conversely, every transition system T induces a canonical hyper transition system $HTS(T)$ which specifies exactly all computations of T.

4.4.4. Definition. For a transition system $T = \langle X, \delta, \longrightarrow \rangle$ define the hyper transition system $HTS(T) = \langle X, \longrightarrow\!\!\in \rangle$ by putting $x \longrightarrow\!\!\in W$ if and only if

$$x \longrightarrow \delta \quad \text{or} \quad (\exists y \in X : x \longrightarrow y) \;\&\; (\forall y \in X : x \longrightarrow y \;\Rightarrow\; y \in W)$$

for every $x \in X$ and $W \subseteq X$.

The computations specified by $HTS(T)$ coincide with the computations of T. This is a consequence of the following lemma.

4.4.5. Lemma. *Let $T = \langle X, \delta, \longrightarrow \rangle$ be a transition system with deadlock. Then $TS(HTS(T)) = T$.*

Proof: Let $TS(HTS(T)) = \langle X, \delta, \longrightarrow' \rangle$ and let $x \in X$. If $x \longrightarrow \delta$ then $x \longrightarrow\!\!\in \emptyset$, by Definition 4.4.4. Hence, by Definition 4.4.3 $x \longrightarrow' \delta$.

Conversely, if $x \longrightarrow' \delta$ then

$$\bigcap \{ W \subseteq X \mid x \longrightarrow\!\!\in W \} = \emptyset.$$

By Definition 4.4.4 this is the case only if $x \longrightarrow \delta$.

Let now $x, y \in X$. If $x \longrightarrow y$ then, by Definition 4.4.4,

$$x \longrightarrow\!\!\in \{ y \in X \mid x \longrightarrow y \}.$$

By Definition 4.4.3 it follows that $x \longrightarrow' y$.

Conversely, if $x \longrightarrow' y$ then, by Definition 4.4.3 there exists $W \subseteq X$ such that $x \longrightarrow\!\!\in W$ and for all $z \in X$ such that $x \longrightarrow' z$, $z \in W$. Hence, by Definition 4.4.4, $x \longrightarrow y$. $\square$

Which are the hyper transition systems that are in one-to-one correspondence with transition systems? In order to characterize them, notice that, for every transition system $T = \langle X, \delta, \longrightarrow \rangle$ the transition relation $\longrightarrow\!\!\in$ of the hyper transition system $HTS(T)$ is upper closed on the right hand side and it satisfies the following property:

$$\exists W \subseteq X\colon x \longrightarrow\!\!\in W \quad \Rightarrow \quad x \longrightarrow\!\!\in \bigcap \{ V \subseteq X \mid x \longrightarrow\!\!\in V \} \tag{4.6}$$

for every $x \in X$.

4.4.6. Lemma. *Let $H = \langle X, \longrightarrow\!\!\in \rangle$ be a hyper transition system satisfying Equation (4.6) and such that the relation $\longrightarrow\!\!\in$ is upper closed on the right hand side. Then $HTS(TS(H)) = H$.*

Proof: Let $HTS(TS(H)) = \langle X, \mathrel{-\!\!\in'}\rangle$, $x \in X$ and $W \subseteq X$. By Definition 4.4.4, if $x \mathrel{-\!\!\in'} W$ then there are two cases: either $x \longrightarrow \delta$ or there exists $y \in X$ such that $x \longrightarrow y$ and $\{y \in X \mid x \longrightarrow y\} \subseteq W$.

In the first case, by Definition 4.4.3, $\bigcap\{W \mid x \mathrel{-\!\!\in} W\} = \emptyset$. Hence there exists $W \subseteq X$ such that $x \mathrel{-\!\!\in} W$ and, by Equation (4.6) $x \mathrel{-\!\!\in} \emptyset$. Since the relation $\mathrel{-\!\!\in}$ is upper closed to the right hand side, $x \mathrel{-\!\!\in} W$.

In the other case, by Definition 4.4.3, there exists $W \subseteq X$ such that $x \mathrel{-\!\!\in} W$ and

$$\bigcap\{W \subseteq X \mid x \mathrel{-\!\!\in} W\} = \{y \in X \mid x \longrightarrow y\}.$$

By Equation (4.6) and the upper closure on the right hand side of the relation $\mathrel{-\!\!\in}$ it follows that $x \mathrel{-\!\!\in} W$.

Conversely, assume $x \mathrel{-\!\!\in} W$. Then, by Equation (4.6), $x \mathrel{-\!\!\in} \bigcap\{W \subseteq X \mid x \mathrel{-\!\!\in} W\}$. Let W_0 denote the set on the right hand side. By Definition 4.4.3, if $W_0 = \emptyset$ then $x \longrightarrow \delta$, otherwise $x \longrightarrow y$ for all $y \in W_0$. In both cases, by Definition 4.4.4, $x \mathrel{-\!\!\in'} W_0$. Since $W_0 \subseteq W$ and the relation $\mathrel{-\!\!\in'}$ is upper closed on the right hand side, $x \mathrel{-\!\!\in'} W$. $\square$

Essentially, what makes a hyper transition system more expressive than an ordinary transition system is the possibility of describing two different kinds of non-determinism in a single framework. However, this does not imply that transition systems are not expressive enough to specify computations. One argument for the introduction of hyper transition systems is that they allow for the specification of a computation in terms of the properties that the atomic steps of the computation have to satisfy.

A hyper transition system for $\mathcal{L}_1$

In this subsection we define a hyper transition system for the language $\mathcal{L}_1$. We consider configurations to be either states in $\mathtt{St}$, representing the final outcomes of the computations, or pairs $\langle S, s\rangle$ where $s \in \mathtt{St}$ is a possible initial or intermediate state of a computation and $S \in Stat_1$ is the specification of the remainder of the computation to be executed.

4.4.7. Definition. Let $(c \in)\ Conf_1 = (Stat_1 \times \mathtt{St}) \cup \mathtt{St}$ be the collection of configurations and define, for every declaration $d : PVar \rightarrow Stat_1$ the hyper transition system $\langle Conf_1, \mathrel{-\!\!\in_d}\rangle$ by taking $\mathrel{-\!\!\in_d}$ to be the least relation between configurations in $Conf_1$ and subsets of configurations of $Conf_1$ satisfying the following axioms and the rules:

$$\langle V\!\rightarrow, s\rangle \mathrel{-\!\!\in_d} W \quad \text{if } s \in V \text{ implies } s \in W$$

$$\langle \{V\}, s\rangle \mathrel{-\!\!\in_d} W \quad \text{if } s \in V \cap W$$

$$\langle \langle f\rangle, s\rangle \mathrel{-\!\!\in_d} W \quad \text{if } f(s) \in W$$

$$\langle x, s\rangle \relbar\!\!\in_d W \qquad \text{if } \langle d(x), s\rangle \in W \text{, for } x \in PVar$$

$$\frac{\langle S_i, s\rangle \relbar\!\!\in_d W}{\langle \bigvee_I S_i, s\rangle \relbar\!\!\in_d W} \qquad \text{if } i \in I$$

$$\frac{\{\langle S_i, s\rangle \relbar\!\!\in_d W_i \mid i \in I\}}{\langle \bigwedge_I S_i, s\rangle \relbar\!\!\in_d \bigcup\{W_i \mid i \in I\}}$$

$$\frac{\langle S_1, s\rangle \relbar\!\!\in_d W}{\langle S_1 \,;\, S_2, s\rangle \relbar\!\!\in_d \{\langle S_2, t\rangle \mid t \in W \cap \mathtt{St}\} \cup \{\langle S_1' \,;\, S_2, t\rangle \mid \langle S_1', t\rangle \in W\}}.$$

An explanation is in order here. According to our interpretation of hyper transition systems, the command $\langle d, V\!\rightarrow\rangle$ specifies a computation that when started at input $s \in V$ terminates in one step with the state s as the only outcome because $\langle V\!\rightarrow, s\rangle \relbar\!\!\in_d \{s\}$. However, if the computation is started at input $s \notin V$ then it must deadlock because $\langle V\!\rightarrow, s\rangle \longrightarrow \emptyset$.

The command $\langle d, \{V\}\rangle$ is similar except that the computations specified by $\langle d, \{V\}\rangle$ are undefined at input $s \notin V$ because no transition is possible from the configuration $\langle\{V\}, s\rangle$.

The command $\langle d, \langle f\rangle\rangle$ specifies a computation that at input s terminates in one step, with as only output the state $f(s)$ (because $\langle\langle f\rangle, s\rangle \relbar\!\!\in_d \{f(s)\}$).

The command $\langle d, x\rangle$ specifies a computation that at input s goes to the configuration $\langle d(x), s\rangle$ (because $\langle x, s\rangle \relbar\!\!\in_d \{\langle d(x), s\rangle\}$).

The command $\langle d, \bigvee_I S_i\rangle$ specifies those computations which are specified by all $\langle d, S_i\rangle$ for $i \in I$. It increases the non-determinism of the specification and hence restricts the non-determinism of the computations. For example, if $\langle S_1, s\rangle \relbar\!\!\in_d \{c_1, c_2\}$ and $\langle S_2, s\rangle \relbar\!\!\in_d \{c_1, c_3\}$ then $\langle S_1 \vee S_2, s\rangle \relbar\!\!\in_d \{c_1, c_2\}$ and $\langle S_1 \vee S_2, s\rangle \relbar\!\!\in_d \{c_1, c_3\}$. Hence $\langle d, S_1 \vee S_2\rangle$ specifies the computation which at input s reaches the configuration c_1. The computations specified by $\langle d, \bigvee_I S_i\rangle$ are undefined at input s only if the computations specified by all $\langle d, S_i\rangle$ for $i \in I$ are undefined at input s. The computations specified by $\langle d, \bigvee_I S_i\rangle$ must deadlock at input s if there is one $\langle d, S_k\rangle$ for $k \in I$ which specifies a computation which must deadlock.

The command $\langle d, \bigwedge_I S_i\rangle$ increases the non-determinism at the level of the specified computations. It specifies computations which behave as any of the computations specified by $\langle d, S_i\rangle$ for $i \in I$. For example, if $\langle S_1, s\rangle \longrightarrow_d \{c_1\}$ and $\langle S_2, s\rangle \relbar\!\!\in_d \{c_2\}$ then $\langle S_1 \wedge S_2, s\rangle \relbar\!\!\in_d \{c_1, c_2\}$. Thus $\langle d, S_1 \wedge S_2\rangle$ specifies, among others, the computation which at input s may choose to go either in the configuration c_1 or in the configuration c_2. Dual to the command $\langle d, \bigvee_I S_i\rangle$, the computations specified by $\langle d, \bigwedge_I S_i\rangle$ are undefined at input s if there is one $\langle d, S_k\rangle$ for $k \in I$ which specifies a computation undefined at input s. Also, the computations specified by $\langle d, \bigwedge_I S_i\rangle$ must deadlock at input s only if the computations specified by all $\langle d, S_i\rangle$ for $i \in I$ must deadlock at input s.

Finally, the command $\langle d, S_1 \,;\, S_2\rangle$ specifies computations that at input s may either deadlock, or go to a configuration $\langle S_2, s'\rangle$ if S_1 specifies a computation which at input s terminates in a state s', or goes to a configuration $\langle S\ ;\ S_2, s'\rangle$ if S_1 specifies a computation which at input s may go in a state s' with $\langle d, S\rangle$ the command specifying the remainder of the computation to be executed.

In order to prove properties of the hyper transition system $\langle \mathit{Conf}_1, \mathrel{-\!\!-\!\!\in}_d\rangle$ we will often use induction on the structure of S. Indeed we can define inductively an assignment of ordinals to statements in Stat_1 by

$$
\begin{array}{lcl}
wgt_1(V\!\rightarrow) & = & 1,\\
wgt_1(\{V\}) & = & 1,\\
wgt_1(\langle f\rangle) & = & 1,\\
wgt_1(x) & = & 1,\\
wgt_1(S_1\ ;\ S_2) & = & wgt_1(S_1) + wgt_1(S_1) + 1,\\
wgt_1(\bigvee_I S_i) & = & \sup\{wgt_1(S_i) \mid i \in I\} + 1,\\
wgt_1(\bigwedge_I S_i) & = & \sup\{wgt_1(S_i) \mid i \in I\} + 1.
\end{array}
$$

Since the index I in the statements $\bigvee_I S_i$ and $\bigwedge_I S_i$ is a set, the above function is well-defined.

The first property we prove of the hyper transition system $\langle \mathit{Conf}_1, \mathrel{-\!\!-\!\!\in}_d\rangle$ is the upper closure on the right hand side of the transition relation $\mathrel{-\!\!-\!\!\in}_d$.

4.4.8. Lemma. *For all commands $\langle d, S\rangle$ of $\mathcal{L}_1$ and states $s \in \mathtt{St}$,*

$$\langle S, s\rangle \mathrel{-\!\!-\!\!\in}_d W_1 \ \&\ W_1 \subseteq W_2 \quad\Rightarrow\quad \langle S, s\rangle \mathrel{-\!\!-\!\!\in}_d W_2.$$

Proof: We prove the lemma by induction on $wgt_1(S)$. Since base cases are obvious we concentrate on the other sub-cases.

$[S_1\ ;\ S_2]$ Let $\langle S_1\ ;\ S_2, s\rangle \mathrel{-\!\!-\!\!\in}_d W_1$ and $W_2 \supseteq W_1$. Define

$$\overline{W_1} = \{s' \mid \langle S_2, s'\rangle \in W_1\} \cup \{\langle S_1', s'\rangle \mid \langle S_1'\ ;\ S_2, s'\rangle \in W_1\}.$$

Similarly define also $\overline{W_2}$. Then $\langle S_1, s\rangle \mathrel{-\!\!-\!\!\in}_d \overline{W_1}$ and $\overline{W_1} \subseteq \overline{W_2}$. Hence, by induction, $\langle S_1, s\rangle \mathrel{-\!\!-\!\!\in}_d \overline{W_2}$. The latter implies $\langle S_1\ ;\ S_2, s\rangle \mathrel{-\!\!-\!\!\in}_d W_2$.

$[\bigvee_I S_i]$ If $\langle \bigvee_I S_i, s\rangle \mathrel{-\!\!-\!\!\in}_d W_1$ then there exists $k \in I$ such that $\langle S_k, s\rangle \mathrel{-\!\!-\!\!\in}_d W_1$. By induction, if $W_2 \supseteq W_1$ then $\langle S_k, s\rangle \mathrel{-\!\!-\!\!\in}_d W_2$. Therefore $\langle \bigvee_I S_i, s\rangle \mathrel{-\!\!-\!\!\in}_d W_2$.

$[\bigwedge_I S_i]$ Assume $\langle \bigwedge_I S_i, s\rangle \mathrel{-\!\!-\!\!\in}_d W_1$. By induction all transitions starting from $\langle S_i, s\rangle$ for $i \in I$ are upper closed on the right. Hence, by definition of $\mathrel{-\!\!-\!\!\in}_d$, $\langle S_i, s\rangle \mathrel{-\!\!-\!\!\in}_d W_1$ for all $i \in I$. If we take $W_2 \supseteq W_1$ then, by induction, $\langle S_i, s\rangle \mathrel{-\!\!-\!\!\in}_d W_2$ for all $i \in I$. Hence $\langle \bigwedge_I S_i, s\rangle \mathrel{-\!\!-\!\!\in}_d W_2$. $\square$

Since the transition relation $\mathrel{-\!\!\in}_d$ is upper closed on the right hand side we have that $\langle \bigwedge_I S_i, s\rangle \mathrel{-\!\!\in}_d W$ if and only if $\langle S_i, s\rangle \mathrel{-\!\!\in}_d W$ for all $i \in I$. Dually, by Definition 4.4.7 $\mathrel{-\!\!\in}_d$, $\langle \bigvee_I S_i, s\rangle \mathrel{-\!\!\in}_d W$ if and only if there exists $k \in I$ such that $\langle S_k, s\rangle \mathrel{-\!\!\in}_d W$.

Recall that the language $\mathcal{L}_0$ can be mapped into the language $\mathcal{L}_1$ via the function $(\cdot)^\dagger$. For $d \in \mathit{Decl}_0$, the restriction of the hyper transition system $\langle \mathit{Conf}_1, \mathrel{-\!\!\in}_{d^\dagger}\rangle$ to a hyper transition system H (with configurations stemming either from state $s \in \mathtt{St}$ or to pair $\langle (S)^\dagger, s\rangle$ with $S \in \mathit{Stat}_0$) induces a transition system $\mathit{TS}(H)$ which is equivalent to H. This is a consequence of Lemma 4.4.8 and of the result below.

4.4.9. Lemma. *For every $\langle d, S\rangle \in \mathcal{L}_0$ and $s \in \mathtt{St}$ if there exists $W \subseteq \mathit{Conf}_1$ such that $\langle (S)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W$ then $\langle (S)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} \bigcap\{W \mid \langle (S)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W\}$.*

Proof: By induction on the structure of $S \in \mathit{Stat}_0$. We consider only one subcase. Assume $\langle (S_1 \sqcap S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W$. Since $(S_1 \sqcap S_2)^\dagger = (S_1)^\dagger \wedge (S_2)^\dagger$, by definition of $\mathrel{-\!\!\in}_{d^\dagger}$,

$$\langle (S_1)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W \quad \text{and} \quad \langle (S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W.$$

Hence, by induction hypothesis,

$$\begin{array}{l}\langle (S_1)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} \bigcap\{W \mid \langle (S_1 \sqcap S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W\} \quad \text{and}\\ \langle (S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} \bigcap\{W \mid \langle (S_1 \sqcap S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W\}.\end{array}$$

We can conclude that $\langle (S_1 \sqcap S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} \bigcap\{W \mid \langle (S_1 \sqcap S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W\}$ because

$$\begin{array}{l}\bigcap\{W \mid \langle (S_1 \sqcap S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W\}\\ \quad = \bigcap\{W \mid \langle (S_1)^\dagger \wedge (S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W\}\\ \quad = \bigcap\{W \mid \langle (S_1)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W \;\&\; \langle (S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W\}\\ \quad = \bigcap\{W \mid \langle (S_1)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W\} \cap \bigcap\{W \mid \langle (S_2)^\dagger, s\rangle \mathrel{-\!\!\in}_{d^\dagger} W\}.\end{array}$$ □

Operational semantics

Next we want to use the hyper transition system $(\mathit{Conf}_1, \mathrel{-\!\!\in}_d)$ to define an operational semantics $\mathit{Op}[\![\cdot]\!]$ for the language $\mathcal{L}_1$. Since we are interested only in the input-output behaviour of the language $\mathcal{L}_1$ we need to abstract from the intermediate configurations recorded by the transition relation of the hyper transition system. Therefore we need to take a kind of transitive closure of the transition relation.

4.4.10. Definition. Let $\langle X, \longrightarrow\!\!\in \rangle$ be a hyper transition system. For every ordinal $\lambda \geq 0$ define the relation $\overset{\lambda}{\longrightarrow\!\!\in}$ on $X \times \mathcal{P}(X)$ inductively by

$$\begin{array}{lcll}
x \overset{0}{\longrightarrow\!\!\in} W & \equiv & x \in W, & \\
x \overset{\lambda+1}{\longrightarrow\!\!\in} W & \equiv & \exists V \subseteq X \colon x \longrightarrow\!\!\in V \ \&\ \forall y \in V \exists \alpha \leq \lambda \colon y \overset{\alpha}{\longrightarrow\!\!\in} W, & \\
x \overset{\lambda}{\longrightarrow\!\!\in} W & \equiv & \exists \alpha < \lambda \colon x \overset{\alpha}{\longrightarrow\!\!\in} W & \text{where } \lambda \text{ is a limit ordinal}
\end{array}$$

for $x \in X$ and $W \subseteq X$.

By induction on λ it is easy to see that, for every ordinal $\lambda \geq 0$, the relation $\overset{\lambda}{\longrightarrow\!\!\in}$ is upper closed on the right hand side if the relation $\longrightarrow\!\!\in$ is upper closed on the right hand side.

The ordinal used to label the transition relation $x \overset{\lambda}{\longrightarrow\!\!\in} W$ is not equal to the number of atomic steps which a computation specified by a hyper transition system starting in a configuration x need to execute in order to satisfy the predicate W. Rather, this label is related to the size of the set on the right hand side of the transition relation.

The relation $x \overset{\lambda}{\longrightarrow\!\!\in} W$ for an infinite ordinal can be defined in terms of the successor ordinal below λ. This technical property will be useful in most of the proofs by induction below.

4.4.11. Lemma. *Let $\langle X, \longrightarrow\!\!\in \rangle$ be a hyper transition system. For every limit ordinal λ, $x \overset{\lambda}{\longrightarrow\!\!\in} W$ if and only if either $x \overset{0}{\longrightarrow\!\!\in} W$ or there exists an ordinal $\alpha < \lambda$ such that $x \overset{\alpha+1}{\longrightarrow\!\!\in} W$.*

Proof: Let λ be a limit ordinal. If $x \overset{0}{\longrightarrow\!\!\in} W$ then $x \overset{\lambda}{\longrightarrow\!\!\in} W$ because $0 < \lambda$. Also, if there exists a $\alpha < \lambda$ such that $x \overset{\alpha+1}{\longrightarrow\!\!\in} W$ then $\alpha + 1 < \lambda$. Hence $x \overset{\lambda}{\longrightarrow\!\!\in} W$.

The converse follows immediately by showing, by induction on λ, that if $x \overset{\lambda}{\longrightarrow\!\!\in} W$ and λ is a limit ordinal then either $x \overset{0}{\longrightarrow\!\!\in} W$ or there exists a $\alpha < \lambda$ such that $x \overset{\alpha+1}{\longrightarrow\!\!\in} W$. □

We can now define a semantics $Op[\![\cdot]\!]$ for the language $\mathcal{L}_1$ in terms of the hyper transition system $(Conf_1, \longrightarrow\!\!\in_d)$.

4.4.12. Definition. (i) Put $Sem_1 = Decl_1 \times Conf_1 \to \mathcal{P}(\mathcal{P}(\mathtt{St}))$ and define $Op \in Sem_1$, for $d \in Decl_1$ and $c \in Conf_1$, by

$$Op(d, c) \;=\; \{P \subseteq \mathtt{St} \mid \exists \lambda \colon c \overset{\lambda}{\longrightarrow\!\!\in}_d P\}.$$

(ii) The operational semantics $Op[\![\cdot]\!] : \mathcal{L}_1 \to (\mathtt{St} \to \mathcal{P}(\mathcal{P}(\mathtt{St}))$ is given by

$$Op[\![\langle d, S\rangle]\!](s) \;=\; Op(d, \langle S, s\rangle).$$

The idea behind the above operational semantics is that of total correctness: if a predicate P on the output space of a program is in $Op[\![\langle d, S\rangle]\!](s)$ then every computation started at input s and specified by the command $\langle d, S\rangle$ of $\mathcal{L}_1$ terminates either in a state $t \in P$ or in the deadlock configuration δ.

4.4.13. Theorem. *Let* $T = \langle Conf_1, \delta, \relbar\!\relbar\!\!\in_d \rangle$ *be the transition system induced by the hyper transition system associated to* $\mathcal{L}_1$ *according to Definition 4.4.3. For all* $\langle d, S\rangle \in \mathcal{L}_1$*,* $P \subseteq \mathtt{St}$ *and* $s \in \mathtt{St}$ *if* $P \in Op[\![\langle d, S\rangle]\!](s)$ *then every computation of* T *starting at* $\langle S, s\rangle$ *is finite and terminates either in the configuration* δ *or in a state* $t \in P$*.*

Proof: It is enough to prove by induction on the ordinal λ that if $\langle S, s\rangle \overset{\lambda}{\relbar\!\relbar\!\!\in}_d P$ then every computation of T starting at $\langle S, s\rangle$ is finite and terminates either in δ or in a state $t \in P$.

For $\lambda = 0$ the above statement is obviously true because there is no $P \subseteq \mathtt{St}$ such that $\langle S, s\rangle \overset{0}{\relbar\!\relbar\!\!\in}_d P$.

Assume the above statement holds for all ordinals $\alpha \leq \lambda$, and let $P \subseteq \mathtt{St}$ such that $\langle S, s\rangle \overset{\lambda+1}{\relbar\!\relbar\!\!\in}_d P$. Let also $(x_n)_n$ be a computation of T with $x_0 = \langle S, s\rangle$.

By definition of $\overset{\lambda+1}{\relbar\!\relbar\!\!\in}_d$ there exists $W \subseteq Conf_1$ such that

$$\langle S, s\rangle \relbar\!\relbar\!\!\in_d W \quad \& \quad \forall c \in W \exists \alpha \leq \lambda \colon c \overset{\alpha}{\relbar\!\relbar\!\!\in}_d P. \tag{4.7}$$

By definition 4.4.3, $\langle S, s\rangle \relbar\!\relbar\!\!\in_d W$ implies that the sequence $(x_n)_n$ contains at least two elements, x_0 and x_1 with $x_0 \relbar\!\relbar\!\!\in_d x_1$ in T. Moreover, either $x_1 = \delta$ or $x_1 \in W$. Since there is no transition in T starting from δ, if $x_1 = \delta$ the the computation $(x_n)_n$ terminates in δ. Otherwise, by (4.7), $x_1 \overset{\alpha}{\relbar\!\relbar\!\!\in}_d P$ for some $\alpha \leq \lambda$. Hence, by induction hypothesis, every computation of T starting at x_1 is finite and terminates either in δ or in state $t \in P$. Since $x_0 \relbar\!\relbar\!\!\in_d x_1$ in T, also the computation $(x_n)_n$ of T is finite and terminates either in δ or in state $x_k \in P$. □

We conjecture that also the converse of the above theorem holds, that is, if every computation specified by the hyper transition system associated with $\mathcal{L}_1$ and starting at $\langle S, s\rangle$ is finite and terminates in either δ or $t \in P$ then $P \in Op[\![\langle d, S\rangle]\!](s)$. A proof of this statement reduces to the proof of the existence of an ordinal λ such that $\langle S, s\rangle \overset{\lambda}{\relbar\!\relbar\!\!\in}_d P$. This will require a rather detailed analysis of the computations specified by a hyper transition system.

The function $Op(\cdot)$ can be characterized as the least solution of an operational fixed point equation.

4.4.14. Theorem. *The function* $Op(\cdot)$ *is the least function in* Sem_1 *such that, for* $d \in Decl_1$*,* $s \in \mathtt{St}$*, and* $S \in Stat_1$*,*

$$\begin{array}{lcl} Op(d, s) & = & \{P \subseteq \mathtt{St} \mid s \in P\} \\ Op(d, \langle S, s\rangle) & = & \bigcup\{\bigcap\{Op(d, c') \mid c' \in W\} \mid \langle S, s\rangle \relbar\!\relbar\!\!\in_d W\} \end{array}$$

Proof: The proof is divided in two parts. We first prove that $Op(\cdot)$ satisfies the above equations, and then we show that $Op(\cdot)$ is the least function which satisfies them.

For $s \in \mathtt{St}$, there is no $W \subseteq Conf_1$ such that $s \relbar\!\relbar\!\!\in_d W$. Hence

$$\begin{aligned} &Op(d, s) \\ &\quad = \{P \subseteq \mathtt{St} \mid \exists\lambda\colon s \xrightarrow{\lambda}\!\!\epsilon_d\, P\} \\ &\quad = \{P \subseteq \mathtt{St} \mid s \xrightarrow{0}\!\!\epsilon_d\, P\} \\ &\quad = \{P \subseteq \mathtt{St} \mid s \in P\}. \end{aligned}$$

For $\langle S, s\rangle \in \mathit{Conf}_1$, $P \in Op(d, \langle S, s\rangle)$ if and only if there exists $\lambda \geq 0$ such that $\langle S, s\rangle \xrightarrow{\lambda}\!\!\epsilon_d\, P$. Since $P \subseteq \mathtt{St}$, $\langle S, s\rangle \notin P$. Hence $\lambda > 0$. There are two cases to be considered: either $\lambda = \alpha + 1$ for some ordinal α or λ is a limit ordinal. In the first case

$$\begin{aligned} &\langle S, s\rangle \xrightarrow{\alpha+1}\!\!\epsilon_d\, P \\ &\Leftrightarrow\ \exists W \subseteq \mathit{Conf}_1\colon \langle S, s\rangle \longrightarrow\!\!\epsilon_d\, W \ \&\ \forall c \in W \exists\beta \leq \alpha\colon c \xrightarrow{\beta}\!\!\epsilon_d\, P \\ &\Leftrightarrow\ \exists W \subseteq \mathit{Conf}_1\colon \langle S, s\rangle \longrightarrow\!\!\epsilon_d\, W \ \&\ \forall c \in W\colon P \in Op(d, c) \qquad [\text{Def. } Op(\cdot)] \\ &\Leftrightarrow\ P \in \bigcup\{\bigcap\{Op(d, c) \mid c \in W\} \mid \langle S, s\rangle \longrightarrow\!\!\epsilon_d\, W\}. \end{aligned}$$

In the second case λ is a limit ordinal. By Lemma 4.4.11 $\langle S, s\rangle \xrightarrow{\lambda}\!\!\epsilon_d\, P$ if and only if either $\langle S, s\rangle \xrightarrow{0}\!\!\epsilon_d\, P$ or there exists an ordinal $\alpha < \lambda$ such that $\langle S, s\rangle \xrightarrow{\alpha+1}\!\!\epsilon_d\, P$. Since $\langle S, s\rangle \notin P$, $\langle S, s\rangle \xrightarrow{0}\!\!\epsilon_d\, P$ does not hold. Hence $\langle S, s\rangle \xrightarrow{\lambda}\!\!\epsilon_d\, P$ if and only if there exists $\alpha < \lambda$ such that $\langle S, s\rangle \xrightarrow{\alpha+1}\!\!\epsilon_d\, P$. We have already seen that the latter is equivalent to

$$P \in \bigcup\{\bigcap\{Op(d, c) \mid c \in W\} \mid \langle S, s\rangle \longrightarrow\!\!\epsilon_d\, W\}.$$

Therefore $Op(\cdot)$ satisfies the two recursive equations above.

Let now $F \in \mathit{Sem}_1$ be another function such that, for $d \in \mathit{Decl}_1$, $s \in \mathtt{St}$, and $S \in \mathit{Stat}_1$,

$$\begin{aligned} F(d, s) &= \{P \subseteq \mathtt{St} \mid s \in P\} \\ F(d, \langle S, s\rangle) &= \bigcup\{\bigcap\{F(d, c') \mid c' \in W\} \mid \langle S, s\rangle \longrightarrow\!\!\epsilon_d\, W\}. \end{aligned}$$

We prove, by induction on λ, that

$$c \xrightarrow{\lambda}\!\!\epsilon_d\, P \quad\Rightarrow\quad P \in F(d, c), \tag{4.8}$$

for all $c \in \mathit{Conf}_1$ and $P \subseteq \mathtt{St}$. It follows that $Op(d, c) \subseteq F(d, c)$.

For $\lambda = 0$ Equation (4.8) clearly holds. Assume it holds for every ordinal $\alpha \leq \lambda$. Then

$$\begin{aligned} &c \xrightarrow{\lambda+1}\!\!\epsilon_d\, P \\ &\Leftrightarrow\ \exists W \subseteq \mathit{Conf}_1\colon c \longrightarrow\!\!\epsilon_d\, W \ \&\ \forall c' \in W \exists\alpha \leq \lambda\colon c' \xrightarrow{\alpha}\!\!\epsilon_d\, P \end{aligned}$$

$$\begin{array}{ll} \Rightarrow & \exists\, W \subseteq \mathit{Conf}_1 \colon c \longrightarrow\!\!\!\in_d W \;\&\; \forall c' \in W \colon P \in F(d, c') \quad \text{[induction]} \\ \Leftrightarrow & P \in F(d, c). \end{array}$$

In the last equivalence we used the fact that $c \longrightarrow\!\!\!\in_d W$ if and only if $c = \langle S, s\rangle$ for some $S \in \mathit{Stat}_1$ and $s \in \mathtt{St}$.

Finally, let λ be a limit ordinal and assume that Equation (4.8) holds for all ordinals $\alpha < \lambda$. Then

$$\begin{array}{ll} & c \xrightarrow{\lambda}\!\!\!\in_d P \\ \Leftrightarrow & \exists \alpha < \lambda \colon c \xrightarrow{\alpha}\!\!\!\in_d P \\ \Rightarrow & \exists \alpha < \lambda \colon P \in F(d, c) \quad \text{[induction]} \\ \Leftrightarrow & P \in F(d, c). \end{array}$$

Hence Equation (4.8) holds for all ordinals. □

The above theorem shows that the operational semantics $\mathit{Op}[\![\langle d, S\rangle]\!](s)$ of a command $\langle d, S\rangle$ in $\mathcal{L}_1$ at input $s \in \mathtt{St}$ abstracts from the intermediate configurations reached by a transition sequence starting from $\langle d, S\rangle$ and collects only the final outcomes.

Operational equals denotational semantics

Next we want to relate the state transformer semantics $\mathit{St}[\![\cdot]\!]$ to the hyper transition system $\langle \mathit{Conf}_1, \longrightarrow\!\!\!\in_d\rangle$. First we need to extend $\mathit{St}[\![\cdot]\!]$ to configurations. Define the function $\mathit{St}^* \colon \mathit{Decl}_1 \times \mathit{Conf}_1 \to \mathit{CDL}(\mathtt{St})$, for $d \in \mathit{Decl}_1$ and $c \in \mathit{Conf}_1$, by

$$\mathit{St}^*(d, c) \;=\; \begin{cases} \{P \subseteq \mathtt{St} \mid s \in P\} & \text{if } c = s \in \mathtt{St} \\ \mathit{St}[\![\langle d, S\rangle]\!](s) & \text{if } c = \langle S, s\rangle \in \mathit{Stat}_1 \times \mathtt{St}. \end{cases}$$

The function St^* is a fixed point of an equation defined in terms of the hyper-transition system $\langle \mathit{Conf}_1, \longrightarrow\!\!\!\in_d\rangle$.

4.4.15. Theorem. *For every $\langle d, S\rangle \in \mathcal{L}_1$ and $s \in \mathtt{St}$,*

$$\mathit{St}^*(d, \langle S, s\rangle) \;=\; \bigcup\{\bigcap\{\mathit{St}^*(d, c) \mid c \in W\} \mid \langle S, s\rangle \longrightarrow\!\!\!\in_d W\}.$$

Proof: In order to simplify the notation, let for $W \subseteq \mathit{Conf}_1$

$$\mathit{lhs}(W) = \bigcap\{\mathit{St}^*(d, c) \mid c \in W\}.$$

To prove the theorem we need to prove for all $P \subseteq \mathtt{St}$,

$$P \in \mathit{St}[\![\langle d, S\rangle]\!](s) \;\Leftrightarrow\; \exists W \subseteq \mathit{Conf}_1 \colon \langle S, s\rangle \longrightarrow\!\!\!\in_d W \;\&\; P \in \mathit{lhs}(W). \quad (4.9)$$

We proceed by induction on $\mathit{wgt}_1(S)$. We treat only two base cases. The cases when $S \equiv \{V\}$ and $S \equiv \langle f\rangle$ can be treated in a way similar to the one below.

$$
\begin{aligned}
& P \in St[\![\langle d, V{\rightarrow}\rangle]\!](s) \\
\Leftrightarrow\ & s \in V \ \Rightarrow\ s \in P \quad \text{[Definition } St[\![\cdot]\!]\text{]} \\
\Leftrightarrow\ & \langle V{\rightarrow}, s\rangle \longrightarrow\!\!\in_d P \quad \text{[Definition } \longrightarrow\!\!\in_d\text{]} \\
\Leftrightarrow\ & \langle V{\rightarrow}, s\rangle \longrightarrow\!\!\in_d P \ \&\ P \in \mathit{lhs}(P),
\end{aligned}
$$

where $P \in \mathit{lhs}(P)$ because, by definition of St^*, $\mathit{lhs}(P) = \{Q \subseteq \mathtt{St} \mid P \subseteq Q\}$.

Let now $x \in PVar$. We have

$$
\begin{aligned}
& P \in St[\![\langle d, x\rangle]\!](s) \\
\Leftrightarrow\ & P \in St[\![\langle d, d(x)\rangle]\!](s) \quad \text{[Definition of } St[\![\cdot]\!]\text{]} \\
\Leftrightarrow\ & P \in St^*(d, \langle d(x), s\rangle) \quad \text{[Definition of } St^*\text{]} \\
\Leftrightarrow\ & \langle x, s\rangle \longrightarrow\!\!\in_d \{\langle d(x), s\rangle\} \ \&\ P \in \mathit{lhs}(\langle d(x), s\rangle). \quad \text{[Definition } \longrightarrow\!\!\in_d\text{]}
\end{aligned}
$$

Next we consider commands $\langle d, S\rangle \in \mathcal{L}_1$ with $wgt_1(S) > 1$. We begin by proving Equation (4.9) for the command $\langle d, \bigvee_I S_i\rangle$:

$$
\begin{aligned}
& P \in St[\![\langle d, \textstyle\bigvee_I S_i\rangle]\!](s) \\
\Leftrightarrow\ & \exists k \in I\colon P \in St[\![\langle d, S_k\rangle]\!](s) \quad \text{[Definition } St[\![\cdot]\!]\text{]} \\
\Leftrightarrow\ & \exists k \in I \exists W_k \subseteq \mathit{Conf}_1 : \langle S_k, s\rangle \longrightarrow\!\!\in_d W_k \ \&\ P \in \mathit{lhs}(W_k) \quad \text{[induction]} \\
\overset{1}{\Leftrightarrow}\ & \exists W \subseteq \mathit{Conf}_1 : \langle \textstyle\bigvee_I S_i, s\rangle \longrightarrow\!\!\in_d W \ \&\ P \in \mathit{lhs}(W),
\end{aligned}
$$

where, by definition of $\longrightarrow\!\!\in_d$, ($\overset{1}{\Rightarrow}$) holds by taking $W = W_k$ whereas ($\overset{1}{\Leftarrow}$) holds by taking $W_k = W$.

Then we prove (4.9) for the command $\langle d, \bigwedge_I S_i\rangle$:

$$
\begin{aligned}
& P \in St[\![\langle d, \textstyle\bigwedge_I S_i\rangle]\!](s) \\
\Leftrightarrow\ & \forall i \in I\colon P \in St[\![\langle d, S_i\rangle]\!](s) \quad \text{[Definition } St[\![\cdot]\!]\text{]} \\
\Leftrightarrow\ & \forall i \in I \exists W_i \subseteq \mathit{Conf}_1 : \langle S_i, s\rangle \longrightarrow\!\!\in_d W_i \ \&\ P \in \mathit{lhs}(W_i) \quad \text{[induction]} \\
\overset{2}{\Leftrightarrow}\ & \exists W \subseteq \mathit{Conf}_1 \forall i \in I : \langle S_i, s\rangle \longrightarrow\!\!\in_d W \ \&\ P \in \mathit{lhs}(W), \\
\Leftrightarrow\ & \exists W \subseteq \mathit{Conf}_1 : \langle \textstyle\bigwedge_I S_i, s\rangle \longrightarrow\!\!\in_d W \ \&\ P \in \mathit{lhs}(W), \quad \text{[Def. } \longrightarrow\!\!\in_d\text{]}
\end{aligned}
$$

where ($\overset{2}{\Rightarrow}$) holds by taking $W = \bigcup_I W_i$ because $\mathit{lhs}(\bigcup_I W_i) = \bigcap_I \mathit{lhs}(W_i)$ (a proof of this statement is immediate) and, by Lemma 4.4.8, $\langle S_i, s\rangle \longrightarrow\!\!\in_d \bigcup_I W_i$ for all $i \in I$. Conversely, ($\overset{2}{\Leftarrow}$) holds by taking $W_i = W$ for all $i \in I$ because, by definition of $\longrightarrow\!\!\in_d$ and Lemma 4.4.8, if $\langle \bigwedge_I S_i, s\rangle \longrightarrow\!\!\in_d W$ then $\langle S_i, s\rangle \longrightarrow\!\!\in_d W$ for all $i \in I$.

It remains to prove Equation (4.9) for the command $\langle d, S_1 \,;\, S_2\rangle$:

$$P \in St[\![\langle d, S_1 \,;\, S_2\rangle]\!](s)$$

$$
\begin{aligned}
&\Leftrightarrow\ s \in Wp_1[\![\langle d, S_1\ ;\ S_2\rangle]\!](P) \quad \text{[Theorem 4.3.6]}\\
&\Leftrightarrow\ s \in Wp_1[\![\langle d, S_1\rangle]\!](Wp_1[\![\langle d, S_2\rangle]\!](P)) \quad \text{[Definition 4.2.2]}\\
&\Leftrightarrow\ Wp_1[\![\langle d, S_2\rangle]\!](P) \in St[\![\langle d, S_1\rangle]\!](s) \quad \text{[Theorem 4.3.6]}\\
&\Leftrightarrow\ \exists\, W \subseteq Conf_1 : \langle S_1, s\rangle \longrightarrow\!\!\in_d W\ \&\ Wp_1[\![\langle d, S_2\rangle]\!](P) \in lhs(W)\\
&\qquad\text{[induction]}\\
&\overset{3}{\Leftrightarrow}\ \exists\,\overline{W} \subseteq Conf_1 : \langle S_1\ ;\ S_2, s\rangle \longrightarrow\!\!\in_d \overline{W}\ \&\ P \in lhs(\overline{W}),
\end{aligned}
$$

where ($\overset{3}{\Rightarrow}$) holds by taking

$$\overline{W} = \{\langle S_2, t\rangle \mid t \in W \cap \mathtt{St}\} \cup \{\langle S_1'\ ;\ S_2, t\rangle \mid \langle S_1', t\rangle \in W\}.$$

Notice that $\langle S_1, s\rangle \longrightarrow\!\!\in_d W$ implies $\langle S_1\ ;\ S_2, s\rangle \longrightarrow\!\!\in_d \overline{W}$, and

$$
\begin{aligned}
&Wp_1[\![\langle d, S_2\rangle]\!](P) \in lhs(W)\\
&\Leftrightarrow\ \forall c \in W : Wp_1[\![\langle d, S_2\rangle]\!](P) \in St^*(d, c) \quad \text{[Definition } lhs(W)\text{]}\\
&\Leftrightarrow\ (\forall t \in W \cap \mathtt{St} : t \in Wp_1[\![\langle d, S_2\rangle]\!](P))\ \&\\
&\qquad (\forall \langle S_1', t\rangle \in W : Wp_1[\![\langle d, S_2\rangle]\!](P) \in St[\![\langle d, S_1'\rangle]\!](t)) \quad \text{[Definition } St^*\text{]}\\
&\Leftrightarrow\ (\forall t \in W \cap \mathtt{St} : P \in St[\![\langle d, S_2\rangle]\!](t))\ \&\\
&\qquad (\forall \langle S_1', t\rangle \in W : t \in Wp_1[\![\langle d, S_1'\rangle]\!](Wp_1[\![\langle d, S_2\rangle]\!](P))) \quad \text{[Th. 4.3.6]}\\
&\Leftrightarrow\ (\forall t \in W \cap \mathtt{St} : P \in St[\![\langle d, S_2\rangle]\!](t))\ \&\\
&\qquad (\forall \langle S_1', t\rangle \in W : t \in Wp_1[\![\langle d, S_1'\ ;\ S_2\rangle]\!](P)) \quad \text{[Definition 4.2.2]}\\
&\Leftrightarrow\ (\forall t \in W \cap \mathtt{St} : P \in St[\![\langle d, S_2\rangle]\!](t))\ \&\\
&\qquad (\forall \langle S_1', t\rangle \in W : P \in St[\![\langle d, S_1'\ ;\ S_2\rangle]\!](t)) \quad \text{[Theorem 4.3.6]}\\
&\Leftrightarrow\ \forall c \in \overline{W} : P \in St^*(d, c) \quad \text{[Definition } \overline{W} \text{ and } St^*\text{]}\\
&\Leftrightarrow\ P \in lhs(\overline{W}). \quad \text{[Definition } lhs(W)\text{]}
\end{aligned}
$$

Conversely ($\overset{3}{\Leftarrow}$) holds by taking

$$W = \{t \mid \langle S_2, t\rangle \in \overline{W}\} \cup \{\langle S_1', t\rangle \mid \langle S_1'\ ;\ S_2, t\rangle \in \overline{W}\}.$$

As above, if $\langle S_1\ ;\ S_2, s\rangle \longrightarrow\!\!\in_d \overline{W}$ then $\langle S_1, s\rangle \longrightarrow\!\!\in_d W$, and $P \in lhs(\overline{W})$ implies $Wp_1[\![\langle d, S_2\rangle]\!](P) \in lhs(W)$. □

As a consequence of the above theorem together with Theorem 4.4.14 we have that $Op[\![\langle d, S\rangle]\!](s) \subseteq St[\![\langle d, S\rangle]\!](s)$ for all commands $\langle d, S\rangle$ in $\mathcal{L}_1$ and inputs $s \in \mathtt{St}$. In order to prove the converse we need to show that the function $Op[\![\cdot]\!]$ satisfies the equations characterizing the forward semantics $St[\![\langle d, S\rangle]\!](s)$ given in Corollary 4.3.7. First we show that every function satisfying the fixed point characterization of the operational semantics $Op[\![\cdot]\!]$ satisfies also many of the equations characterizing the state transformer semantics $St[\![\cdot]\!]$.

4.4.16. Lemma. *Let $F : Decl_1 \times Conf_1 \to \mathcal{P}(\mathcal{P}(\mathtt{St}))$ be a function such that, for $d \in Decl_1$, $s \in \mathtt{St}$, and $S \in Stat_1$,*

$$F(d,s) = \{P \subseteq \mathtt{St} \mid s \in P\} \quad (4.10)$$
$$F(d,\langle S,s\rangle) = \bigcup\{\bigcap\{F(d,c') \mid c' \in W\} \mid \langle S,s\rangle \longrightarrow_d W\}. \quad (4.11)$$

Then, for every $d \in Decl_1$ and $s \in \mathtt{St}$,

(i) $F(d,\langle V\!\to, s\rangle) = \{P \subseteq \mathtt{St} \mid s \in V \Rightarrow s \in P\}$,
(ii) $F(d,\langle\{V\}, s\rangle) = \{P \subseteq \mathtt{St} \mid s \in V \cap P\}$,
(iii) $F(d,\langle\langle f\rangle\rangle) = \{P \subseteq \mathtt{St} \mid f(s) \in P\}$,
(iv) $F(d,\langle x, s\rangle) = F(d,\langle d(x), s\rangle)$,
(v) $F(d,\langle \bigvee_I S_i, s\rangle) = \bigcup\{F(d,\langle S_i,s\rangle) \mid i \in I\}$,
(vi) $F(d,\langle \bigwedge_I S_i, s\rangle) = \bigcap\{F(d,\langle S_i,s\rangle) \mid i \in I\}$.

Proof: We begin by proving item (i).

$$P \in F(d,\langle V\!\to, s\rangle)$$
$$\Leftrightarrow \exists W \subseteq Conf_1 : \langle V\!\to, s\rangle \longrightarrow_d W \;\&\; \forall c \in W : P \in F(d,c) \quad [(4.11)]$$
$$\Leftrightarrow \exists W \subseteq Conf_1 : (s \in V \Rightarrow s \in W) \;\&\; \forall c \in W : P \in F(d,c)$$
$$[\text{Definition } \longrightarrow_d]$$
$$\overset{1}{\Leftrightarrow} s \in V \Rightarrow P \in F(d,s)$$
$$\Leftrightarrow s \in V \Rightarrow s \in P,$$

where ($\overset{1}{\Leftarrow}$) holds by taking $W = \{s\}$. Items (ii) and (iii) can be treated similarly.

Item (iv) follows immediately from the definition of $\longrightarrow_d$:

$$P \in F(d,\langle x, s\rangle)$$
$$\Leftrightarrow \exists W \subseteq Conf_1 : \langle x, s\rangle \longrightarrow_d W \;\&\; \forall c \in W : P \in F(d,c) \quad [(4.11)]$$
$$\overset{2}{\Leftrightarrow} P \in F(d,\langle d(x), s\rangle),$$

where ($\overset{2}{\Leftarrow}$) holds by taking $W = \{\langle d(x), s\rangle\}$, whereas ($\overset{2}{\Rightarrow}$) holds because $\langle d(x), s\rangle \in W$ by definition of $\longrightarrow_d$.

Next we prove item (v):

$$P \in F(d,\langle \bigvee_I S_i, s)$$
$$\Leftrightarrow \exists W \subseteq Conf_1 : \langle \bigvee_I S_i, s\rangle \longrightarrow_d W \;\&\; \forall c \in W : P \in F(d,c) \quad [(4.11)]$$
$$\Leftrightarrow \exists W \subseteq Conf_1\, \exists k \in I : \langle S_k, s\rangle \longrightarrow_d W \;\&\; \forall c \in W : P \in F(d,c) \quad [\text{Def. } \longrightarrow_d]$$
$$\Leftrightarrow \exists k \in I : P \in F(d,\langle S_k, s) \quad [(4.11)]$$

$$\Leftrightarrow \quad P \in \bigcup\{F(d, \langle S_i, s) \mid i \in I\}.$$

In order to prove item (vi) we use the fact that $\langle \bigwedge_I S_i, s\rangle \mathrel{-\!\!\in}_d W$ if and only if $\langle S_i, s\rangle \mathrel{-\!\!\in}_d W$ for all $i \in I$. This statement is a consequence of Lemma 4.4.8 and the definition of $\mathrel{-\!\!\in}_d$. We have

$$
\begin{aligned}
& P \in F(d, \langle \textstyle\bigwedge_I S_i, s) \\
& \Leftrightarrow \quad \exists\, W \subseteq \mathit{Conf}_1 : \langle \textstyle\bigwedge_I S_i, s\rangle \mathrel{-\!\!\in}_d W \;\&\; \forall c \in W : P \in F(d, c) \qquad [(4.11)] \\
& \overset{4}{\Leftrightarrow} \quad \forall i \in I \exists\, W_i \subseteq \mathit{Conf}_1 : \langle S_i, s\rangle \mathrel{-\!\!\in}_d W_i \;\&\; \forall c \in W_i : P \in F(d, c) \\
& \Leftrightarrow \quad \forall i \in I : P \in F(d, \langle S_i, s) \qquad [(4.11)] \\
& \Leftrightarrow \quad P \in \bigcap\{F(d, \langle S_i, s) \mid i \in I\},
\end{aligned}
$$

where ($\overset{3}{\Leftarrow}$) holds by taking $W = \bigcup W_i$, while ($\overset{3}{\Rightarrow}$) holds by taking $W_i = W$ for all $i \in I$. □

Next we prove that the operational semantics of the sequential composition of two statements can be expressed in terms of the components.

4.4.17. Lemma. *For $d \in \mathit{Decl}_1$, $s \in \mathtt{St}$, and $S_1, S_2 \in \mathit{Stat}_1$,*

$$\mathit{Op}[\![\langle d, S_1 \,;\, S_2\rangle]\!](s) = \bigcup\{\bigcap\{\mathit{Op}[\![\langle d, S_2\rangle]\!](t) \mid t \in Q\} \mid Q \in \mathit{Op}[\![\langle d, S_1\rangle]\!](s)\}.$$

Proof: The proof consists of two parts. In the first part we show the inclusion from left to right, whereas in the second part we show the converse.

Let $d \in \mathit{Decl}_1$ be a fixed but arbitrary declaration. To prove the inclusion from left to right it is enough to show, by induction on λ, that, for all $P \subseteq \mathtt{St}$, $s \in \mathtt{St}$ and $S_1, S_2 \in \mathit{Stat}_1$, if

$$\langle S_1 \,;\, S_2, s\rangle \overset{\lambda}{\mathrel{-\!\!\in}}_d P$$

then

$$\exists Q \subseteq \mathtt{St}\colon \langle S_1, s\rangle \overset{\lambda}{\mathrel{-\!\!\in}}_d Q \;\&\; (\forall t \in Q\colon P \in \mathit{Op}[\![\langle d, S_2\rangle]\!](t)). \tag{4.12}$$

For $\lambda = 0$ the above assertion is always true because $\langle S_1 \,;\, S_2, s\rangle \notin P$.

Assume now $\langle S_1 \,;\, S_2, s\rangle \overset{\lambda+1}{\mathrel{-\!\!\in}}_d P$. By definition of the transition relation $\overset{\lambda+1}{\mathrel{-\!\!\in}}_d$, there exists $W \subseteq \mathit{Conf}_1$ such that

$$\langle S_1 \,;\, S_2, s\rangle \mathrel{-\!\!\in}_d W \;\&\; \forall c \in W \exists \alpha \leq \lambda\colon c \overset{\alpha}{\mathrel{-\!\!\in}}_d P. \tag{4.13}$$

Put $\overline{W} = \{t \mid \langle S_2, t\rangle \in W\} \cup \{\langle S, t\rangle \mid \langle S \,;\, S_2, t\rangle \in W\}$. By (4.13) and the definition of the hyper transition system for $\mathcal{L}_1$ we have that $\langle S_1, s\rangle \mathrel{-\!\!\in}_d \overline{W}$. Moreover, by (4.13),

$$\forall t \in \overline{W} \exists \alpha \leq \lambda\colon \langle S_2, t\rangle \overset{\alpha}{\mathrel{-\!\!\in}}_d P \tag{4.14}$$

and also

$$\forall \langle S, t\rangle \in \overline{W} \exists \alpha \leq \lambda \colon \langle S \,;\, S_2, t\rangle \stackrel{\alpha}{\relbar\!\!\relbar\!\!\in}_d P. \tag{4.15}$$

By definition of the function $Op[\![\cdot]\!]$, (4.14) implies

$$\forall t \in \overline{W} \colon P \in Op[\![\langle d, S_2\rangle]\!](t). \tag{4.16}$$

By induction hypothesis, (4.15) implies that for all $\langle S, t\rangle \in \overline{W}$ there exists $Q(\langle S, t\rangle) \subseteq \mathtt{St}$ and $\alpha \leq \lambda$ such that

$$\langle S, t\rangle \stackrel{\alpha}{\relbar\!\!\relbar\!\!\in}_d Q(\langle S, t\rangle) \;\&\; \forall t' \in Q(\langle S, t\rangle) \colon P \in Op[\![\langle d, S_2\rangle]\!](t'). \tag{4.17}$$

Take now $\overline{Q} = \{Q(\langle S, t\rangle) \mid \langle S, t\rangle \in \overline{W}\}$. Because $\stackrel{\alpha}{\relbar\!\!\relbar\!\!\in}_d$ is upper closed on the right hand side (4.17) implies that for all $\langle S, t\rangle \in \overline{W}$ there exists $\alpha \leq \lambda$ such that

$$\langle S, t\rangle \stackrel{\alpha}{\relbar\!\!\relbar\!\!\in}_d \overline{Q} \;\&\; \forall t' \in \overline{Q} \colon P \in Op[\![\langle d, S_2, t'\rangle]\!]. \tag{4.18}$$

Finally, put $Q = \overline{Q} \cup \{t \in \mathtt{St} \mid t \in \overline{W}\}$. Because $\stackrel{\alpha}{\relbar\!\!\relbar\!\!\in}_d$ is upper closed on the right hand side, and $t \stackrel{0}{\relbar\!\!\relbar\!\!\in}_d Q$ for all $t \in \overline{W} \cap \mathtt{St}$ we have, combining (4.16) and (4.18),

$$\forall c \in \overline{W} \exists \alpha \leq \lambda \colon c \stackrel{\alpha}{\relbar\!\!\relbar\!\!\in}_d Q \;\&\; (\forall t \in Q \colon P \in Op[\![\langle S_2, t\rangle]\!](t)).$$

Since $\langle S_1, s\rangle \relbar\!\!\relbar\!\!\in_d \overline{W}$ we obtain, by definition of $\stackrel{\lambda+1}{\relbar\!\!\relbar\!\!\in}_d$,

$$\langle S_1, s\rangle \stackrel{\lambda+1}{\relbar\!\!\relbar\!\!\in}_d Q \;\&\; (\forall t \in Q \colon P \in Op[\![\langle d, S_2\rangle]\!](t)).$$

Therefore if λ is a successor ordinal and $\langle S_1 \,;\, S_2, s\rangle \stackrel{\lambda}{\relbar\!\!\relbar\!\!\in}_d P$ then (4.12) holds. In fact it holds for every ordinal because of Lemma 4.4.11. Hence

$$Op[\![\langle d, S_1 \,;\, S_2\rangle]\!](s) \subseteq (Op[\![\langle d, S_1\rangle]\!] \,;\, Op[\![\langle d, S_2\rangle]\!])(s),$$

for all $s \in \mathtt{St}$.

To prove the converse we show that for a fixed declaration $d \in Decl_1$ and for all ordinals λ, $P \subseteq \mathtt{St}$, $s \in \mathtt{St}$ and $S_1, S_2 \in Stat_1$ if

$$\exists Q \subseteq \mathtt{St} \colon \langle S_1, s\rangle \stackrel{\lambda}{\relbar\!\!\relbar\!\!\in}_d Q \;\&\; (\forall t \in Q \colon P \in Op[\![\langle d, S_2\rangle]\!](t)) \tag{4.19}$$

then

$$P \in Op[\![\langle d, S_1 \,;\, S_2\rangle]\!](s).$$

We proceed by induction on λ. In case $\lambda = 0$ clearly there is no $Q \subseteq \mathtt{St}$ such that $\langle S_1, s\rangle \stackrel{0}{\relbar\!\!\relbar\!\!\in}_d Q$. Hence the statement (4.19) implies $P \in Op[\![\langle d, S_1 \,;\, S_2\rangle]\!](s)$ is clearly true.

Assume now there exists $Q \subseteq \mathtt{St}$ such that

$$\langle S_1, s\rangle \xrightarrow{\lambda+1}\!\!\in_d Q \quad \& \quad \forall t \in Q\colon P \in Op[\![\langle d, S_2\rangle]\!])(t). \tag{4.20}$$

By definition of $\xrightarrow{\lambda+1}\!\!\in_d$ there exists $W \subseteq Conf_1$ such that

$$\langle S_1, s\rangle \longrightarrow\!\!\in_d W \;\&\; \forall c \in W \exists \alpha \leq \lambda\colon c \xrightarrow{\alpha}\!\!\in_d Q. \tag{4.21}$$

Observe that the configuration c in W can be of two types: either $c = t \in \mathtt{St}$ or $c = \langle S, t\rangle \in Stat_1 \times \mathtt{St}$. In the first case, by definition of $\xrightarrow{\alpha}\!\!\in_d$ and Lemma 4.4.11, $t \xrightarrow{\alpha}\!\!\in_d Q$ implies $\alpha = 0$. Hence $t \in Q$, from which it follows, by (4.20), that

$$\forall t \in W \cap \mathtt{St}\colon P \in Op[\![\langle d, S_2\rangle]\!](t). \tag{4.22}$$

In the second case $\langle S, t\rangle \xrightarrow{\alpha}\!\!\in_d Q$ with $\alpha \leq \lambda$ and $P \in Op[\![\langle d, S_2\rangle]\!])(t')$ for all $t' \in Q$ (by (4.20)) implies, by induction hypothesis, that

$$\forall \langle S, t\rangle \in W\colon P \in Op[\![\langle d, S\,;\,S_2\rangle]\!](t). \tag{4.23}$$

Define now $\overline{W} = \{\langle S_2, t\rangle \mid t \in W\} \cup \{\langle S\,;\,S_2, t\rangle \mid \langle S, t\rangle \in W\}$. By definition of $\longrightarrow\!\!\in_d$ and (4.21), $\langle S_1\,;S_2, s\rangle \longrightarrow\!\!\in_d \overline{W}$. By (4.22) $\langle S_2, t\rangle \in \overline{W}$ implies $P \in Op[\![\langle d, S_2\rangle]\!](t)$; and by (4.23) $\langle S\,;S_2, t\rangle \in \overline{W}$ implies $P \in Op[\![\langle d, S\,;S_2\rangle]\!](t)$. Thus

$$\langle S_1\,;\,S_2, s\rangle \in \overline{W} \;\&\; \forall c \in \overline{W}\colon P \in Op(\langle d, c\rangle).$$

By Theorem 4.4.14 this implies $P \in Op(\langle d, \langle S_1\,;\,S_2, s\rangle\rangle) = Op[\![\langle d, S_1\,;\,S_2]\!](s)$.

Therefore if (4.19) holds for a successor ordinal λ then $P \in Op[\![\langle d, S_1\,;\,S_2\rangle]\!](s)$. If λ is a limit ordinal then (4.19) implies $P \in Op[\![\langle d, S_1\,;\,S_2\rangle]\!](s)$ by Lemma 4.4.11 and the above. Hence we can conclude that

$$(Op[\![\langle d, S_1\rangle]\!]\,;\,Op[\![\langle d, S_2\rangle]\!])(s) \subseteq Op[\![\langle d, S_1\,;\,S_2]\!](s)$$

for all $s \in \mathtt{St}$. □

The above lemma together with Lemma 4.4.16 applied to the function $Op(\cdot)$ imply that $Op[\![\cdot]\!]$ satisfies the same equations that are satisfied by the state transformer semantics $St[\![\cdot]\!]$. Since the latter is the least function satisfying the equations given in Corollary 4.3.7 we obtain that the forward semantics $St[\![\cdot]\!]$ coincides with the operational semantics $Op[\![\cdot]\!]$.

4.4.18. Theorem. *For every $\langle d, S\rangle \in \mathcal{L}_1$ and $s \in \mathtt{St}$,*

$$Op[\![\langle d, S\rangle]\!](s) = St[\![\langle d, S\rangle]\!](s).$$

Proof: By Theorem 4.4.15, the function St^* satisfies both the Equations (4.10) and (4.11). By Theorem 4.4.14, $Op(\cdot)$ is the least function which satisfies those equations. Hence $Op[\![\langle d, S\rangle]\!](s) \subseteq St[\![\langle d, S\rangle]\!](s)$ for all $s \in \mathtt{St}$.

By Corollary 4.3.7, Lemmas 4.4.16 and 4.4.17, we obtain the converse. Therefore $Op[\![\langle d, S\rangle]\!](s) = St[\![\langle d, S\rangle]\!](s)$. □

This result and Theorem 4.3.6 demonstrate that the operational semantics $Op[\![\cdot]\!]$ and the predicate transformer semantics $Wp_1[\![\cdot]\!]$ are isomorphic.

A game-theoretical interpretation

We now briefly develop an alternative interpretation of a hyper transition system based on a game between two players, one called *angel* and another called *demon*. Our notion of game is inspired by the game interpretation of the refinement calculus put forward by Back and Von Wright [BW90a] and formally developed in [Hes94] and [BW96].

A hyper transition system $\langle X, \longrightarrow\!\!\in \rangle$ defines the possible configurations of the game by means of the set X, and the possible moves of the game by means of the relation $\longrightarrow\!\!\in$.

The game starts in a given configuration $x \in X$. The angels aims to stop in a configuration $y \in P$ for a given set of terminating configurations $P \subseteq X$, whereas the demon aims to prevent it. The angel plays first by choosing $W \subseteq X$ such that $x \longrightarrow\!\!\in W$. Then, the demon plays by choosing a configuration $y \in W$ and the game restarts from the configuration y. The game terminates if no move is possible. There are two cases: either the game is in a configuration x but there is no $W \subseteq X$ such that $x \longrightarrow\!\!\in W$, or the angel has already chosen a set of configurations W but there is no $y \in W$ (that is, $W = \emptyset$). In the first case, if $x \notin P$ then the demon wins. Otherwise the angel wins.

A winning strategy for the angel is a function $F : X \to \mathcal{P}(\mathcal{P}(X))$ such that $P \in F(x)$ if and only if either $x \in P$ and there is no move for the angel (that is, there is no $W \subseteq X$ such that $x \longrightarrow\!\!\in W$), or there exists a move for the angel who chooses $W \subseteq X$ such that $x \longrightarrow\!\!\in W$ and for all possible choices $y \in W$ of the demon, $P \in F(y)$.

In Theorem 4.4.18 we have proved that the least winning strategy for the angel with respect to the game defined by the hyper transition system $\langle Conf_1, \longrightarrow\!\!\in_d \rangle$ induced by $\mathcal{L}_1$ coincides with the state transformer semantics $St[\![\cdot]\!]$ of $\mathcal{L}_1$. In other words, $P \in St[\![\langle d, S\rangle]\!](s)$ (or, equivalently, $s \in Wp_1[\![\langle d, S\rangle]\!](P)$) if and only if there exists a play in the game defined by $\langle Conf_1, \longrightarrow\!\!\in_d \rangle$ which starts in the configuration $\langle S, s\rangle$ and terminates in P with the victory of the angel.

4.5 Concluding notes

In the refinement calculus commands are identified with predicate transformers in order to avoid problems associated with the existence of infinitary free algebras, as discussed for example in Section 2.1. Hesselink [Hes92a] discusses the existence of free complete specification algebras, where a specification algebra is an algebra with an operator of composition and a binary meet. It is called complete if it allows unbounded meets. In general the completion of a specification algebra does not need to exist, since it can be a proper class rather than a set. The isomorphism of Theorem 4.3.3 clarifies what are the right equations for ensuring the existence of a complete specification algebra: the unbounded meets should completely distribute over the unbounded joins. In Chapter 9 we will return to this topic by proving the existence of a free completely distributive lattice over a set X.

Our forward semantics for the refinement calculus is inspired by the minimal models for modal logic of Chellas [Che80]. Chellas's minimal models are a generalization of Kripke models. They are indexed functions mapping each possible world to sets of possible worlds, and are used as models of monotonic modal logic.

The operational interpretation of the refinement calculus we presented in this chapter differs in the following aspects from the game semantics of Back and Von Wright [BW96] and the game semantics of Hesselink [Hes94]. Back and Von Wright define a game interpretation of the commands of the refinement calculus using a standard transition system. A transition step corresponds to a move in the game. A configuration is said to be angelic if only the angel can make a move and is said to be demonic otherwise. This suggests a close relation to our hyper transition system model. However, every sequence of transitions in the game interpretation of Back and Von Wright is finite (in fact infinite plays are not possible), and we allow also infinite sequences. The game semantics for the refinement calculus given by Hesselink uses hyper transition systems which allow for infinite games. However, both the hyper transition system induced by the refinement calculus and the way of collecting the information from it is different from our operational approach. Furthermore, our operational interpretation can be used for the step-by-step specification of computations.

Also, our game interpretation of the refinement calculus differs from both the game semantics of Back and Von Wright [BW96] and the game semantics of Hesselink [Hes94]. The main difference is that our games are not symmetric (and therefore we do not have to take sides): the angel always makes the first move. The goal of the angel is different from the goal of the demon. Moreover, the angel and the demon take turns, whereas in the other game interpretations the choice of the player who plays depends on the configuration the game is in.

We investigated angelic non-determinism only for sequential languages. The reader interested in the connection between operational and denotational semantics for a simple language supporting angelic non-determinism and parallel composition is referred to [MO91, MO92].

We conclude with a short discussion about the size of the set $PT_M(Y, X)$ of monotonic predicate transformers. For Y an infinite set, Markowsky [Mar79, Theorem 2] proved that

$$\mid CDL(Y) \mid = 2^{2^{|Y|}},$$

where $\mid \cdot \mid$ is the function which assigns to every set its cardinality. Since $\mid ST(X, Y) \mid = \mid CDL(Y) \mid^{|X|}$, if Y is an infinite set then by Theorem 4.3.3

$$\mid PT_M(X, Y) \mid = (2^{2^{|Y|}})^{|X|} = 2^{(2^{|Y|} \times |X|)}.$$

If both X and Y are infinite countable sets then $\mid Y \mid = \mid X \mid = \omega_0$ (the cardinality of the set of all natural numbers). By Cantor's theorem the cardinality of ω_0 is strictly smaller than the cardinality of 2^{ω_0}. Hence, by [Kun80, Corollary I.10.13],

$$2^{\omega_0} \times \omega_0 = \max\{2^{\omega_0}, \omega_0\} = 2^{\omega_0}.$$

If we assume the Generalized Continuum Hypothesis [Kun80, Definition I.10.28] then

$$\mid PT_M(X, Y) \mid = 2^{(2^{\mid Y \mid} \times \mid X \mid)} = 2^{(2^{\omega_0} \times \omega_0)} = 2^{2^{\omega_0}} = 2^{\omega_1} = \omega_2.$$

If Y is a finite set then $\mid PT_M(X, Y) \mid$ can also be calculated using the more complicated characterization of the size of $CDL(Y)$ given in [Mar73].

Part II

Topological dualities

Chapter 5

Topology and affirmative predicates

In the first part of this thesis we considered predicates to be subsets of an abstract set of states. If we think of the states as the denotations of results of computations of programs then predicates become computationally meaningful in the sense that we can use partial information about a computation to tell whether or not a predicate holds for that computation. A predicate for which only finite information about a computation is needed to affirm whether it holds is called an *affirmative predicate*.

Affirmative predicates are closed under finite intersections and arbitrary unions. Hence they can be identified with the open sets of a topological space. The idea that 'open sets are observable predicates' was proposed by Smyth in [Smy83b], although it is also briefly mentioned in [Plo81a]. Smyth interprets open sets as semi-decidable properties in some 'effectively given' topological space. More generally, open sets can be interpreted as (finitely) observable predicates [Abr87, Smy92]. Alpern and Schneider [AS85] and Kwiatkowska [Kwi91] use open sets as 'finite liveness predicates' and closed sets as 'safety predicates' to formalize the informal characterization of liveness and safety properties of Lamport [Lam77]. The name 'affirmative predicates' has been introduced by Vickers [Vic89] for denoting the abstract open sets of a frame. Affirmative predicates are also called verifiable predicates by Rewitzky [Rew96], who uses the term observable for predicates which are both affirmative and refutative.

In this chapter we introduce a few topological concepts which we will also need in the subsequent chapters. We motivate these concepts from the point of view of the affirmative predicates.

5.1 Affirmative and refutative predicates

Assume that we run a program which outputs a sequence of states in Σ. Let P be a predicate on Σ^∞, the set of all finite and infinite sequences of states. Following the definitions of Chapter 3, the predicate P can be seen, extensionally, as a subset of Σ^∞, which holds for a sequence $w \in \Sigma^\infty$ if $w \in P$. In practice, we can inspect the output sequence w of the program as it proceeds. Hence, based only on the finite segments of w which have been output so far, we can sometimes affirm whether the predicate P holds for w. We can never affirm, on the basis of our finite observations, whether the predicate

$$P = \{v \in \Sigma^\infty \mid v \text{ has infinitely many occurrences of } s \in \Sigma\ \}$$

holds for w. We need to refine our definition of predicate to capture predicates that we can observe. Informally, a predicate P on a set X is said to be *affirmative* if we can affirm that it holds for some x in X only on the basis of what we can actually observe, where an observation must be made within a finite amount of time. In general there is no requirement that the absence of a property should be observable. A predicate P is said to be *refutative* if we can refute it for some x in X on the basis of finite information.

Different physical assumptions on the nature of the observations will describe different collections of affirmative predicates. For example, we can assume that our program can diverge, that is, it can produce some finite output and then compute forever without any further output. Hence we cannot distinguish on the basis of finite segments of an output w between a computation which halts and a computation which diverges. Under this assumption a predicate P on Σ^∞ is affirmative if for all $w \in P$ there exists a finite segment v of w such that every extension of v belongs to P. Clearly, the predicate $\{w\}$ is not affirmative for all sequences w, whereas the set $\uparrow w$ of all extensions of w is an affirmative predicate if the sequence w is of finite length.

Alternatively we can assume that our program can continue forever outputting an infinite sequence, but that it has also the additional capacity to halt, for example by signaling when a computation terminates. Thus we have that for a finite sequence w both the predicates $\{w\}$ and $\uparrow w$ are affirmative. Technically this can be obtained as follows. A predicate P on Σ^∞ is affirmative if for all $w \in P$ there exists a natural number n such that if the length of the longest common prefix of w and any other string v is less than n then v belongs to P.

For every set X, to affirm $x \in X$ no observation is necessary. It can be just affirmed. Hence X itself is always an affirmative predicate. Also, we can never affirm $x \in \emptyset$ for all $x \in X$. Hence $\emptyset$ is an affirmative predicate of X.

In general, affirmative properties over a set X are closed under arbitrary unions and finite intersections. Let P_i, for $i \in I$, be an arbitrary collection of properties on X. To affirm $x \in \bigcup_I P_i$ it is enough to affirm $x \in P_i$ for some $i \in I$. Hence, if all P_i are affirmative properties of X then also their union $\bigcup_I P_i$ is an affirmative predicate. The same cannot be said for arbitrary intersections. To affirm $x \in \bigcap_I P_i$ we need to affirm $x \in P_i$ for all $i \in I$. If I is an infinite set, this may take an infinite amount of time even if all P_i are affirmative properties.

However, if I is a finite index set, and all P_i are affirmative properties, then also $\bigcap_I P_i$ is affirmative.

The complement of an affirmative predicate is, in general, not affirmative. Indeed, to affirm $x \in X \setminus P$ we must refute $x \in P$. Therefore, complement transforms affirmative properties in refutative ones, and vice versa. Using the De Morgan's laws, we have that refutative properties are closed under finite unions and arbitrary intersections. Since the classical implication $P \Rightarrow Q$ can be defined in terms of complement, neither affirmative nor refutative properties are closed under classical implication.

The closure of affirmative properties under finite intersections and arbitrary unions implies that they form a topology on X [Smy83b, Smy92].

5.1.1. Definition. A *topology* on a set X is a collection $\mathcal{O}(X)$ of subsets of X that is closed under finite intersections and arbitrary unions. A *topological space* is a set X together with a topology $\mathcal{O}(X)$ on X. The elements of $\mathcal{O}(X)$ are the *open sets* of the space.

To simplify notation we usually write X for a topological space $(X, \mathcal{O}(X))$. Notice that a topology $\mathcal{O}(X)$ on a set X is a complete lattice when ordered by subset inclusion. Since arbitrary unions distribute over finite intersections, the lattice of open sets of a topological space X is a frame.

A subset c of a space X is said to be *closed* if it is the complement of an open subset of X. The collection of all closed sets of X is denoted by $\mathcal{C}(X)$ and, dually to the case of open sets, is closed under finite unions and arbitrary intersections. Closed sets are ordered by superset inclusion. Any topology on X induces a closure operator. For every subset $V \subseteq X$ define its *closure* $cl(V)$ as the smallest closed set including V, that is

$$cl(V) = \bigcap\{c \in \mathcal{C}(X) \mid V \subseteq c\}.$$

One can easily verify that $V \subseteq cl(V)$, $cl(\emptyset) = \emptyset$, and $cl(V) = cl(cl(V))$. The latter implies that the fixed points of cl are exactly the closed sets of X. Moreover, for any subsets V_1 and V_2 of X, $cl(V_1 \cup V_2) = cl(V_1) \cup cl(V_2)$, and if $V_1 \subseteq V_2$ then $cl(V_1) \subseteq cl(V_2)$.

On a set X we can always define at least two topologies: the *discrete topology* $\mathcal{O}_d(X) = \mathcal{P}(X)$ (every predicate is affirmative), and the *indiscrete topology* $\mathcal{O}_i(X) = \{\emptyset, X\}$ (no non-trivial predicate is affirmative).

A topology on a set X can be specified in terms of a collection of elementary affirmative properties. Other properties can then be constructed by closing them under arbitrary unions and finite intersections.

5.1.2. Definition. A *sub-base* B of a topology $\mathcal{O}(X)$ is a collection of open sets such that every open set is the union of intersections of finitely many elements of B. If B is already closed under finite intersections, then it forms a *basis* and its elements are called *basic opens*. A space having a countable base is said to be *second countable* .

For example, the collection of all singletons $\{x\}$, with $x \in X$, is a sub-base for the discrete topology. The singleton $\{x\}$ represents the most elementary (non-trivial) affirmation we can make about X.

Once we have fixed a collection of (sub-basic) affirmative properties on a set X, then we can use it to determine which elements are observationally equivalent. Even more, we can use affirmative properties to determine an information preorder between points: x_2 has all observable information of x_1 if every affirmative predicate of x_1 is also an affirmative predicate of x_2.

5.1.3. Definition. Let X be a topological space. The *specialization preorder* $\lesssim_{\mathcal{O}}$ on X induced by the topology $\mathcal{O}(X)$ is defined, for x_1 and x_2 in X, by

$$x_1 \lesssim_{\mathcal{O}} x_2 \text{ if and only if } \forall o \in \mathcal{O}(X) : x_1 \in o \;\Rightarrow\; x_2 \in o\,.$$

For example, consider the set of finite and infinite strings Σ^∞ together with the topology defined by taking as basic open sets the sets $\uparrow w$ of all extensions of w, for all finite strings w. For arbitrary strings v_1 and v_2 of Σ^∞, we have $v_1 \lesssim_{\mathcal{O}} v_2$ if and only if every finite prefix w of v_1 is also a finite prefix of v_2. But this is equivalent to stating that v_1 is a prefix of v_2. Hence, the prefix order on strings and the specialization preorder coincide.

If we take as sub-basic opens for Σ^∞ both $\uparrow w$ and $\{w\}$ (for finite strings w), then $v_1 \lesssim_{\mathcal{O}} v_2$ if and only if $v_1 = v_2$. To prove the last statement it is enough to consider the following two cases:

(i) if v_1 is a finite string then $\{v_1\}$ is an affirmative predicate of both v_1 and v_2 if and only if $v_1 = v_2$;

(ii) if v_1 is infinite, then $v_1 \lesssim_{\mathcal{O}} v_2$ if and only if every finite prefix of v_1 is a prefix of v_2, that is, $v_1 = v_2$.

Topological spaces can be classified on the basis of the possibility to separate different points by means of opens.

5.1.4. Definition. A space X is said to be $\mathcal{T}_0$ if the induced specialization preorder $\lesssim_{\mathcal{O}}$ is antisymmetric, that is, it is a partial order. If $\lesssim_{\mathcal{O}}$ is also discrete then X is said to be a $\mathcal{T}_1$ space. Finally, X is said to be a $\mathcal{T}_2$ space (or *Hausdorff*) if, whenever x_1 and x_2 are two distinct points of X, there are two disjoint open sets containing x_1 and x_2 respectively.

Every $\mathcal{T}_2$ space is $\mathcal{T}_1$, and every $\mathcal{T}_1$ space is $\mathcal{T}_0$. In practice we almost always identify any two points of a space X which have the same information, that is, most of the computationally interesting spaces are at least $\mathcal{T}_0$. Define an equivalence relation $\sim$ on X by

$$x_1 \sim x_2 \quad \text{if and only if} \quad x_1 \lesssim_{\mathcal{O}} x_2 \text{ and } x_2 \lesssim_{\mathcal{O}} x_1.$$

If we now write $[x]$ for the equivalence class containing x, and $X/\sim$ for the set of equivalence classes, then the $\mathcal{T}_0$-*ification* of X is defined as the space $X/\sim$ with as opens the collection of all sets $\{[x] \mid x \in o\}$ for all $o \in \mathcal{O}(x)$.

Let $f : X \to Y$ be a function between topological spaces, and let P be a predicate on Y. To affirm that $f(x)$ has the predicate P, it should suffice to affirm $x \in f^{-1}(P)$.

5.1.5. Definition. Let X and Y be two spaces. A function $f : X \to Y$ is called *continuous* if, for all opens o of Y,

$$f^{-1}(o) = \{x \mid f(x) \in o\}$$

is open in X (or, equivalently, if the inverse of each closed set is closed). Topological spaces form a category $\mathbf{Sp}$ with as morphisms the continuous functions. We write $\mathbf{Sp}_0$ and $\mathbf{Sp}_1$ for the full sub-categories of $\mathcal{T}_0$ and $\mathcal{T}_1$ spaces, respectively.

It is easy to see that a continuous function $f : X \to Y$ is monotonic with respect to the specialization orders of X and Y [Smy92, Proposition 4.2.4]. Hence continuous functions preserve the observable information. For example, the assignment $x \mapsto [x]$ from a space X to its $\mathcal{T}_0$-ification $X/\sim$ defines a continuous function.

5.2 Specifications, saturated sets and filters

In [Smy83b] it was suggested that a specification of an object (a program, for example) can be an arbitrary list of affirmative predicates, understood as a conjunction, that the object has to satisfy. Although in practice lists of finite or countable length of affirmative predicates are enough as specifications, lists of arbitrary length are a useful mathematical generalization which will make the theory we develop in the successive chapters easier. In our view of affirmative predicates as open sets of a topological space, a specifiable predicate is a set obtained as the intersection of arbitrarily many open sets, that is, a saturated set. Saturated sets which are intersections of countably many open sets are often called G_δ sets in the literature.

5.2.1. Definition. Let X be a topological space. A subset q of X is said to be *saturated* if

$$q = \bigcap\{o \in \mathcal{O}(X) \mid q \subseteq o\}.$$

The collection of the saturated subsets of X is denoted by $\mathcal{Q}(X)$.

An *intersection system* on a set X is a collection of subsets of X closed under arbitrary intersection. For every space X, the collection $\mathcal{Q}(X)$ of saturated sets is, by definition, the least intersection system including all open subsets of X. Moreover, by using the complete distributivity, we can express an arbitrary union of saturated sets as an intersection of opens. Since the latter is a saturated set, we have that saturated sets are closed under arbitrary unions and arbitrary intersections. Therefore $\mathcal{Q}(X)$ is a ring of subsets of X, from which it follows that $\mathcal{Q}(X)$ ordered by subset inclusion is a completely distributive lattice.

Notice also that $\mathcal{Q}(X)$ is a topology on X which is closed under arbitrary intersections. Hence, for a $\mathcal{T}_0$ space X, $\mathcal{Q}(X)$ coincides with the collection of all upper sets of X with respect to the specialization order $\lesssim_{\mathcal{O}}$ [Joh82a, page 45].

5.2.2. Lemma. *For a $\mathcal{T}_0$ space X and $A \subset X$,*

$$\uparrow A = \bigcap\{o \in \mathcal{O}(X) \mid A \subseteq o\},$$

where $\uparrow A$ is the upper closure of A with respect to the specialization preorder $\lesssim_{\mathcal{O}}$ on X induced by the topology $\mathcal{O}(X)$.

Proof: The inclusion from left to right is immediate since every open set is upper closed with respect to the specialization preorder $\lesssim_{\mathcal{O}}$. Conversely, let $x \in \bigcap\{o \in \mathcal{O}(X) \mid A \subseteq o\}$ and assume $x \notin \uparrow A$. Then $a \not\lesssim_{\mathcal{O}} x$ for all $a \in A$. Thus for all $a \in A$ there exists $o_a \in \mathcal{O}(X)$ such that $a \in o_a$ and $x \notin o_a$. For $o = \bigcup\{o_a \mid a \in A\}$ we then have the contradiction that $A \subseteq o$ and $x \notin o$. $\square$

In case X is a $\mathcal{T}_1$ space, the specialization order is the identity. Thus every subset of X is upper closed. From the above discussion it follows that every predicate of X is specifiable, that is, $\mathcal{Q}(X) = \mathcal{P}(X)$.

A specification $\mathcal{F}$, understood as list of affirmative predicates over a space X, is said to be

(i) *proper* if $\emptyset \notin \mathcal{F}$;
(ii) *deductively closed* if $P \in \mathcal{F}$ and $P \subseteq Q$ implies $Q \in \mathcal{F}$; and
(iii) *consistent* if $P \in \mathcal{F}$ and $Q \in \mathcal{F}$ implies $P \cap Q \in \mathcal{F}$.

The above merely says that a proper, deductively closed and consistent specification is a filter of the lattice of opens $\mathcal{O}(X)$. If we want to specify a single element of X then completely prime filters are more adequate: a point which satisfies the disjunction of some predicates, satisfies at least one of these predicates. Indeed it can be easily checked that a space X is $\mathcal{T}_0$ if and only if for every completely prime filter $\mathcal{F}$ of $\mathcal{O}(X)$ there exists at most one point $x \in X$ such that

$$\mathcal{F} = \{o \in \mathcal{O}(X) \mid x \in o\}.$$

A space with a bijective correspondence between points and their specifications is called sober.

5.2.3. Definition. A space X is said to be *sober* if for every completely prime filter $\mathcal{F}$ of $\mathcal{O}(X)$ there exists exactly one point $x \in X$ such that

$$\mathcal{F} = \{o \in \mathcal{O}(X) \mid x \in o\}.$$

For example, every Hausdorff space is sober [Smy92, Proposition 4.3.14]. From the above characterization of $\mathcal{T}_0$ spaces in terms of completely prime filters, it follows that every sober space is $\mathcal{T}_0$. The full sub-category of **Sp** whose objects are sober spaces will be denoted by **Sob**. For an example of a $\mathcal{T}_1$ space which is not sober, and of a sober space which is not $\mathcal{T}_1$ we refer to [Smy92, IV, Example 4.1.4].

5.3 Examples of topological spaces

In this section we introduce the topologies which we will use in the rest of this thesis.

Alexandrov topology

Given a poset P, the *Alexandrov topology* $\mathcal{O}_A(P)$ on P is defined as the collection of all upper closed subsets of P. Clearly, if P is a discrete poset, then the Alexandrov topology on P coincides with the discrete topology. In general, a poset P with the Alexandrov topology is a $\mathcal{T}_0$ space.

The specialization preorder induced by the Alexandrov topology coincides with the partial order on P. Hence the collection of saturated subsets of P coincides with the collection of Alexandrov open subsets of P. A function $f : P \to Q$ between two posets is monotone if and only if it is continuous with respect to their Alexandrov topologies [Ale37].

Scott topology

The Alexandrov topology on a poset is not always computationally adequate: it should be refined in a such way that if we can affirm that a predicate holds for the least upper bound of a directed set $\mathcal{V}$ then we can affirm it already for some of its approximants in $\mathcal{V}$.

5.3.1. Definition. The *Scott topology* $\mathcal{O}_S(P)$ on a dcpo P consists of all the upper-closed subsets of P such that for any directed set $D \subseteq X$,

$$\bigvee D \in o \;\Rightarrow\; D \cap o \neq \emptyset.$$

As in the case of the Alexandrov topology, the specialization preorder induced by the Scott topology on a dcpo P coincides with the partial order on

P [GHK$^+$80, Remark II.1.4]. Hence the collection of saturated sets of P coincides with the Alexandrov topology on P. Also, a function $f : P \to Q$ between two dcpo's is continuous for the Scott topologies of P and Q if and only if it preserves directed joins [Sco72].

The Scott topology generalizes the discrete topology in the following sense. For a set X, if we assume that singleton sets $\{x\}$ are the most elementary affirmations we can make then the collection of all affirmative predicates is the discrete topology on X. Assume that we can also affirm that no element has been observed yet, for example because of divergence. This can be described as the Scott topology on the flat cpo $X_\perp$: every subset of X is Scott open as well as the set $X \cup \{\perp\}$. Notice that for flat cpo's, the Alexandrov and the Scott topology coincide.

The following proposition gives a connection between sober spaces and dcpo's [Joh82a, Lemma 1.9].

5.3.2. Proposition. *If X is a sober space then the specialization preorder on X has all directed joins. Moreover, every open set $o \in \mathcal{O}(X)$ is Scott open.* □

The converse of the above proposition is false: not every dcpo is sober even when taken with the Scott topology [Joh81]. The desired result can be obtained if we impose more structure on a dcpo. In particular, an algebraic dcpo P taken with the Scott topology is sober [Hof79, Mis91]. In this case we can describe the Scott topology by means of the compact elements of P: the set of all upper closed sets $\uparrow b$ for compacts $b \in \mathcal{K}(P)$ forms a basis for the Scott topology of P.

The lattice of open sets of a space X is clearly a dcpo. In the previous section we suggested that specifications are lists of open sets, and hence a predicate on the lattice $\mathcal{O}(X)$. A specification $\mathcal{F}$ is said to be *finitary* if it is a Scott open set of the lattice $\mathcal{O}(X)$, that is, whenever the union of a directed set $D \subseteq \mathcal{O}(X)$ of affirmative predicates is in $\mathcal{F}$, then some predicate in D is already in $\mathcal{F}$. Scott open filters allow us to define compact subsets as finitarily specifiable subsets.

5.3.3. Definition. A subset S of a space X is *compact* if and only if the set

$$\{o \in \mathcal{O}(X) \mid S \subseteq o\}$$

is a Scott-open filter of $\mathcal{O}(X)$. A space X is said to be *compact* if the set X is compact.

Equivalently, one can use the following (more standard) definition of compactness. A subset S of a space X is compact if and only if for every directed collection D of open sets in $\mathcal{O}(X)$ such that $S \subseteq \bigcup D$ there exists an open set $o \in D$ such that $S \subseteq o$. Notice that an open subset of a space X is compact if and only if it is a compact element of the dcpo $\mathcal{O}(X)$ in the domain theoretical sense (as introduced in Section 2.2).

For example, in any space X, every finite subset of X is compact, as well as any arbitrary subset containing the least element $\perp$ with respect to the specialization preorder on X. Also, for an algebraic dcpo P taken with the Scott topology, every basic open set $\uparrow b$, with $b \in \mathcal{K}(P)$, is compact.

A useful tool for proving compactness of a space is the Alexander sub-basis theorem [Ale39]

5.3.4. Proposition. *Let X be a space with sub-base B. Then X is compact if for every directed collection D of sub-basic open sets in B such that $X \subseteq \bigcup D$ there exists an open set $b \in D$ with $X \subseteq b$.* □

Stone and spectral spaces

We have seen in the previous subsection that an open subset of a space X is compact if and only if it is a compact element of the dcpo $\mathcal{O}(X)$ in the domain theoretical sense. However the lattice $\mathcal{O}(X)$ need not to be algebraic.

5.3.5. Definition. A space X with an algebraic lattice of opens $\mathcal{O}(X)$ is said to be *locally open compact*.

This means that the collection $\mathcal{KO}(X)$ of compact open subsets of X forms a basis for the topology $\mathcal{O}(X)$. Therefore locally open compact space are often called spaces with a basis of compact opens [GHK$^+$80]. In terms of points, locally open compactness can be formulated as follows (for a proof of the proposition below see either [GHK$^+$80] or [BK94a]).

5.3.6. Proposition. *Let X be a space. The lattice $\mathcal{O}(X)$ is an algebraic dcpo if and only if for every point $x \in X$ and open set $o \in \mathcal{O}(X)$ such that $x \in o$ there exists a compact open set $u \in \mathcal{KO}(X)$ satisfying $x \in u \subseteq o$.* □

Locally open compactness does not imply soberness: every poset P taken with the Alexandrov topology is a locally open compact space but need not to be sober. For $x \in P$ and Alexandrov open o such that $x \in o$, if we take the compact Alexandrov open set $\uparrow x$ then we have $x \in \uparrow x \subseteq o$.

Also the Scott topology of an algebraic dcpo P is locally open compact (and sober). For $x \in P$ and Scott open o satisfying $x \in o$, by definition of algebraicity and of the Scott topology, there exists a compact element $b \in \mathcal{K}(P)$ such that $b \leq x$ and $b \in o$. Hence $x \in \uparrow b \subseteq o$. Since $\uparrow b$ is compact in the Scott topology of P, it follows that P as a topological space is locally open compact.

Locally open compact spaces are of interest because they are finitary in the following sense: every affirmative predicate can be retrieved by the finitarily specifiable affirmative predicates because the lattice of opens $\mathcal{O}(X)$ is (isomorphic to) the ideal completion of its basis of compact opens $\mathcal{KO}(X)$.

5.3.7. Definition. A topological space X is said to be *spectral* if the set $\mathcal{KO}(X)$ of compact open subsets of X forms a basis for $\mathcal{O}(X)$ and it is closed under finite intersections. If, moreover, compact opens are closed under complement, then X is said to be a *Stone space*.

Since finite unions of compact opens are again compact open sets, it follows that in a spectral space X, the lattice $\mathcal{KO}(X)$ of compact opens is distributive, while in a Stone space X the lattice $\mathcal{KO}(X)$ of compact opens is a Boolean algebra.

Our interest in spectral and Stone spaces is justified by the following observation. Every SFP domain taken with the Scott topology is spectral [Plo81a], and every compact ultra-metric space taken with the metric topology (to be defined below) is a Stone space [Smy92, Corollary 6.4.8].

Coherent spaces

The key property of spectral and Stone spaces is the fact that their lattice of open sets is algebraic. Hence every open set can be obtained as union of compact open sets. A weaker but similar result can be obtained for a larger class of topological spaces: the coherent spaces.

5.3.8. Definition. For a space X let $\mathcal{KQ}(X)$ be the set of all compact saturated sets of X. The space X is said to be *coherent* if it is sober, $\mathcal{KQ}(X)$ is closed under finite intersections, and, for all open sets o, it holds that

$$o = \bigcup\{u \in \mathcal{O}(X) \mid \exists q \in \mathcal{KQ}(X) \colon u \subseteq q \subseteq o\}.$$

By the above definition it follows that, in a coherent space X, every open set is the directed union of all compact saturated sets which are included in it, that is, every affirmative predicate on X can be approximated by finitarily specifiable predicates.

Every spectral space (and hence every Stone space) is coherent. However, not every algebraic dcpo, even if taken with the Scott topology, is coherent. Coherent spaces as finitary algebraic structures (proximity lattices) are studied in [JS96].

Metric topology

Partial orders and metric spaces play a central role in the semantics of programming languages (see, e.g. [Win93] and [BV96]). The order can be used to give a comparative description of computations, whereas the metric gives quantitative information. This quantitative information can be used to define a topology on the underlying set.

5.3.9. Definition. The *metric topology* on a metric space X is defined by taking as open sets all subsets $o \subseteq X$ with the following property:

$$x \in o \;\Rightarrow\; \exists \epsilon > 0 \colon B_\epsilon(x) \subseteq o,$$

where $B_\epsilon(x) = \{y \mid d_X(x, y) < \epsilon\}$.

For every $\epsilon > 0$, and $x \in X$ the ball $B_\epsilon(x)$ is an open set in the metric topology. Even more, the set of all balls $B_\epsilon(x)$ for every $\epsilon > 0$ and $x \in X$ forms a basis for the metric topology. One can easily verify that every metric space with the above topology forms an Hausdorff space, and hence that it is a sober space. It follows that the collection of saturated sets of a metric space X is $\mathcal{P}(X)$, that is, every predicate on X is specifiable. For a discrete metric space the metric topology coincides with the discrete topology.

Closed and compact subsets of the space induced by a metric space X coincide, respectively, with the closed and compact subsets of the metric space X as defined in Section 2.3 (see [Eng89]).

5.3.10. Proposition. *Let X be a metric space and $S \subseteq X$. The set S is closed in the metric topology if and only if the limit of every convergent sequence in S is an element of S. Also, S is compact in the metric topology if and only if for every sequence in S there is a sub-sequence converging in S.* □

Every non-expansive function $f : X \to Y$ between two metric spaces is continuous with respect to their metric topologies. However, the converse does not hold: f is continuous if and only if

$$\forall x_1 \in X \forall \epsilon > 0 \exists \delta > 0 \forall x_2 \in X \colon d_X(x_1, x_2) \leq \delta \;\Rightarrow\; d_Y(f(x_1), f(x_2)) \leq \epsilon.$$

5.4 Final remarks

Affirmative predicates on X can be described intensionally as continuous functions from the space X to the poset $2 = \{0, 1\}$ with $0 \leq 1$ taken with the Alexandrov topology. In fact we have an order-isomorphism

$$\mathcal{O}(X) \;\cong\; X \to_c 2$$

where $X \to_c 2$ is the continuous function space ordered pointwise. An interesting generalization would be to consider functions from X to the closed interval of reals $[0, 1]$. These functions can be thought of as fuzzy predicates of X. The value a function $\phi : X \to [0, 1]$ assigns to an element x in X can be thought of as a measure for the extent to which x is an element of ϕ. The connections between fuzzy predicates and affirmative predicates have been exploited in the context of generalized metric spaces by Lawvere [Law73], and more recently in [BBR96b].

We conclude this chapter with a few remarks on generalized metric spaces. Since they are not objects of study in the present work, we give only pointers to some of the literature.

Generalized metric spaces provide a framework for the study of both preorders and ordinary metric spaces. A generalized metric space consists of a set X together with a distance function which does not need to be symmetric. Moreover, different points can have zero distance. Generalized metric spaces were introduced by Lawvere [Law73, Law86] as an illustration of the thesis that fundamental structures are categories, and they were subsequently studied in a topological context by Smyth [Smy87, Smy91] as computational spaces: they combine the qualitative information of (observational) preorders with the quantitative (behavioural) information of a distance function.

Some of the basic theory of generalized (ultra-)metric spaces has been developed in [Wag94, Rut95, FK96], where an approach to the solution of recursive domain equations is presented which extends both the order-theoretic [SP82] and the metric [AR89] approaches.

Fundamental constructions for generalized (ultra-)metric spaces like completion [Smy87, BBR96a, FW96] and powerdomains [Smy91, BBR96a] reconcile respective constructions for preorders and metric spaces.

Both the Alexandrov and the Scott topology for preorders can be extended to generalized metric spaces in a such way that for ordinary metric spaces they both correspond to the metric topology [Smy87, BBR96a, BBR96b]. For the restricted class of algebraic complete quasi metric spaces the generalized Scott topology has been shown to be sober [FS96].

Chapter 6

Powerspaces, multifunctions and predicate transformers

Programming languages can often be defined in terms of atomic statements (like assignments to variables), a set of statement operators (like sequential composition and non-deterministic choice), a set of process variables, and a recursion operator for each process variable. To give a compositional semantics to a program it is therefore necessary to define a semantic domain in which atomic statements and statements operators can be interpreted. Modeling recursion is one of the difficult aspects of building a compositional semantics. For this reason the input and output state spaces of a program are often structured, complete with respect to some limit construction, and recursively defined.

In Chapter 3 we introduced two different models for a compositional semantics of a programming language: the state transformer model and the predicate transformer model. A rich collection of semantic constructions is available for state transformers on structured sets of states. However, the same cannot be said for predicate transformers. Nothing, or very little, is known about compositional predicate transformer semantics for programs which interact with their environment.

In this chapter we investigate the relationships between state transformers and predicate transformers in a general topological setting. Topological dualities between predicate and state transformers provide a mathematical approach to predicate transformers between structured sets of states. The connection between state transformers and topological predicate transformers was first studied by Smyth [Smy83b] who placed the result of Plotkin [Plo79] for the Smyth powerdomain in a broader topological framework using the upper powerspace. This work is our starting point. However Smyth restricts to sober spaces while we use $\mathcal{T}_0$ spaces. Also, our techniques are more in line with the ones used in [Plo79]. Besides the upper powerspace we consider also the lower and the Vietoris powerspaces, and we show that the three isomorphisms established in Chapter 3 also hold in this general setting. In passing, topological representations of order theoretic and metric powerdomains are given.

All topological dualities we describe are order-preserving. As a consequence, to define a predicate transformer semantics from a state transformer semantics it is sufficient to define only predicate transformers for the atomic statements and operators on predicate transformers corresponding to syntactical operators. Recursive constructs can be handled in the predicate transformer semantics exactly in the same way as for the state transformer semantics.

6.1 Multifunctions as state transformers

One way to capture a compositional semantics of a concurrent program is to consider it as a function from input states to the set of all intermediate states through which the program P passes after one atomic step, followed by the semantics of the remaining part of the program to be executed. In order to deal with this recursive definition, states are usually endowed with a topological structure (usually a partial order or a distance function). To model non-deterministic computations, semantic functions can be represented by many-valued functions, or multifunctions.

6.1.1. Definition. A *multifunction* $f : X \to Y$ with values in $\mathcal{V} \subseteq \mathcal{P}(Y)$ is a function that assigns to every element x of a topological space X a subset $f(x) \in \mathcal{V}$ of a topological space Y.

For a multifunction $f : X \to Y$ and a predicate P on Y, we denote by $f^+(P)$ the *upper inverse* of f, that is, the set of all inputs x of f such that every element of $f(x)$ satisfies the predicate P. The *lower inverse* of f is denoted by $f^-(P)$ and is defined as the set of all inputs x of f such that some elements of $f(x)$ satisfy the predicate P. Formally, for $P \subseteq Y$:

$$f^+(P) = \{x \in X \mid f(x) \subseteq P\} \quad \text{and} \quad f^-(P) = \{x \in X \mid f(x) \cap P \neq \emptyset\}.$$

Notice that f^+ and f^- are *dual function* in the sense that $f^+(P) = X \setminus f^-(Y \setminus P)$ for all $P \subseteq Y$. Different ways of defining inverse give rise to different ways of defining continuity. Below we list three definitions of continuity for a multifunction [Ber63].

6.1.2. Definition. Let X and Y be two topological spaces. A multifunction $f : X \to Y$ with values in $\mathcal{V} \subseteq \mathcal{P}(Y)$ is said to be

(i) *lower semi-continuous* if $f^-(o) \in \mathcal{O}(X)$ for every $o \in \mathcal{O}(Y)$,

(ii) *upper semi-continuous* if $f^+(o) \in \mathcal{O}(X)$ for every $o \in \mathcal{O}(Y)$, and

(iii) *continuous* if it is both upper and lower semi-continuous.

Since f^- and f^+ are dual functions, the above notions of continuity could also have been expressed in terms of refutative predicates rather than affirmative ones. For example, f is lower semi-continuous if and only if $f^+(c)$ is a closed subset of X for every closed subset c of Y. For every notion of continuity of a multifunction there is a related topology on the collection of subsets of the codomain [Mic51, Nad78].

6.1.3. Definition. Let $\mathcal{V}$ be a set of subsets of a space X.

(i) The *lower topology* on $\mathcal{V}$ has as sub-base the collection of all sets of the form L_o for $o \in \mathcal{O}(X)$, where

$$L_o \;=\; \{S \in \mathcal{V} \mid S \cap o \neq \emptyset\}.$$

(ii) The *upper topology* on $\mathcal{V}$ is defined by taking as base the the collection of all sets of the form U_o for $o \in \mathcal{O}(X)$, where

$$U_o \;=\; \{S \in \mathcal{V} \mid S \subseteq o\}.$$

(iii) The *Vietoris topology* on $\mathcal{V}$ has as sub-base the union of the base of the upper topology and the sub-base of the lower topology.

The definitions of the (sub-)bases of the above topologies are chosen in this way in order to make the proof of the following proposition trivial [Mic51] (see also [Smy83b]).

6.1.4. Proposition. *Let X, Y be two topological spaces, and let $f : X \to Y$ be a multifunction with values in $\mathcal{V} \subseteq \mathcal{P}(Y)$ with $\mathcal{V} \neq \emptyset$. Then*

(i) *$f : X \to Y$ is lower semi-continuous if and only if the corresponding function $f : X \to \mathcal{V}$ is continuous with respect to the lower topology on $\mathcal{V}$;*

(ii) *$f : X \to Y$ is upper semi-continuous if and only if $f : X \to \mathcal{V}$ is continuous with respect to the upper topology on $\mathcal{V}$; and*

(iii) *$f : X \to Y$ is continuous if and only if $f : X \to \mathcal{V}$ is continuous with respect to the Vietoris topology on $\mathcal{V}$.*

Moreover, these three topologies on $\mathcal{V}$ are the only ones that have these properties. □

In general, for an arbitrary collection of subsets $\mathcal{V}$ of a space X, the upper, lower and Vietoris topologies on $\mathcal{V}$ do not ensure that the resulting space is $\mathcal{T}_0$.

6.1.5. Lemma. *Let $\mathcal{V}$ be a set of subsets of a space X, and $A, B \in \mathcal{V}$,*

(i) *$A \lesssim B$ in the preorder induced by the lower topology on $\mathcal{V}$ if and only if $cl(A) \subseteq cl(B)$, where cl is the closure operator induced by the topology on X;*

(ii) *$A \lesssim B$ in the preorder induced by the upper topology on $\mathcal{V}$ if and only if $\uparrow A \supseteq \uparrow B$, where the upper closure is taken with respect to the specialization preorder of X;*

(iii) *$A \lesssim B$ in the preorder induced by the Vietoris topology on $\mathcal{V}$ if and only if both $cl(A) \subseteq cl(B)$ and $\uparrow A \supseteq \uparrow B$.*

Proof: (i) Let $o \in \mathcal{O}(X)$ be such that $A \in L_o$. Since $A \subseteq cl(A)$, $cl(A) \cap o \neq \emptyset$. If $cl(A) \subseteq cl(B)$ then also $cl(B) \cap o \neq \emptyset$. It follows that also $B \cap o \neq \emptyset$, because otherwise $B \subseteq X \setminus o$ would imply $cl(B) \subseteq X \setminus o$, contradicting $B \cap o \neq \emptyset$. Hence $B \in L_o$. For the converse, assume $A \in L_o$ implies $B \in L_o$ for every open o. Since $B \subseteq cl(B)$, $B \notin L_{X \setminus cl(B)}$. Hence also $A \notin L_{X \setminus cl(B)}$, that is, $A \subseteq cl(B)$. Therefore $cl(A) \subseteq cl(B)$.

(ii) Assume $A \in U_o$ for some $o \in \mathcal{O}(X)$. Then $\uparrow A \subseteq o$ by definition of specialization preorder. If $\uparrow A \supseteq \uparrow B$, then also $\uparrow B \subseteq o$. But $B \subseteq \uparrow B$, thus $B \in U_o$. Conversely, assume $A \in U_o$ implies $B \in U_o$ for every open o. Since $\uparrow A = \bigcap\{o \in \mathcal{O}(X) \mid A \subseteq o\}$, we can immediately conclude that $\uparrow A \supseteq \uparrow B$.

(iii) Combine the two previous items. $\square$

The above lemma justifies the following restriction which considers only certain kinds of subsets of a space. Starting from a topological space X, we consider three spaces of subsets of X:

(i) the *lower powerspace* of X, denoted by $\mathcal{P}_l(X)$ and defined as the collection of all closed subsets of X taken with the lower topology;

(ii) the *upper powerspace* of X, denoted by $\mathcal{P}_u(X)$ and defined as the collection of all upper closed subsets of X taken with the upper topology; and

(iii) the *convex powerspace* of X, denoted by $\mathcal{P}_c(X)$ and defined as the collection of all convex closed subsets of X taken with the Vietoris topology, where $S \subseteq X$ is convex closed if $S = cl(S) \cap \uparrow S$.

Variations of the above powerspaces can be obtained by deleting the empty set or restricting to finitarily specifiable subsets using compact sets. Below we denote by $\mathcal{P}_u^{co}(X)$ the collection of all upper closed and compact subsets of X taken with the upper topology, whereas $\mathcal{P}_c^{co}(X)$ denotes the collection of all convex closed and compact subsets of X taken with the Vietoris topology.

From Lemma 6.1.5 it follows that the three powerspaces above are $\mathcal{T}_0$ (more precisely, they are isomorphic in **Sp** to the $\mathcal{T}_0$-ification of $\mathcal{P}(X)$ taken with the lower, upper and Vietoris topology, respectively).

Let X and Y be two topological spaces. Three posets of *topological state transformers* can be identified:

- the *lower state transformers*, i.e. continuous functions from X to $\mathcal{P}_l(Y)$ ordered by the pointwise extension of the specialization preorder induced by the lower topology;

- the *upper state transformers*, i.e. continuous functions from X to $\mathcal{P}_u(Y)$ ordered by the pointwise extension of the specialization preorder induced by the upper topology;

- the *convex state transformers*, i.e. continuous functions from X to $\mathcal{P}_c(Y)$ ordered by the pointwise extension of the specialization preorder induced by the Vietoris topology.

The above domains of topological state transformers can be related to the three domains of state transformers introduced in Chapter 3 as follows. Let X, Y be two sets, and consider the flat cpo $Y_\perp$ taken with the Scott topology. Then, by definition of the Scott topology on $Y_\perp$,

$$\begin{array}{rcl} \mathcal{P}_l(Y_\perp) & = & \{S \subseteq Y \cup \{\perp\} \mid S \neq \emptyset \ \Rightarrow\ \perp \in S\}, \\ \mathcal{P}_u(Y_\perp) & = & \mathcal{P}(Y) \cup \{Y_\perp\}, \\ \mathcal{P}_c(Y_\perp) & = & \mathcal{P}(Y \cup \{\perp\}). \end{array}$$

Hence $\mathcal{P}_l(Y_\perp) \setminus \{\emptyset\} \cong \mathcal{P}(Y)$. If we take X with the discrete topology then every function from X to one of the three powerspaces above is continuous. By the Definitions 3.2.6, 3.2.1 and 3.2.9 of the Hoare, Smyth, and Egli-Milner state transformers, respectively, it follows that

$$\begin{array}{rcl} ST_H(X,Y) & \cong & X \to (\mathcal{P}_l(Y_\perp) \setminus \{\emptyset\}), \\ ST_S(X,Y) & = & X \to \mathcal{P}_u(Y_\perp), \text{ and} \\ ST_E(X,Y) & = & X \to \mathcal{P}_c(Y_\perp). \end{array}$$

A subset of $Y_\perp$ is compact in the Scott topology if and only if it is either finite or contains the bottom element $\perp$. Therefore

$$ST_S^{fin}(X,Y) = X \to \mathcal{P}_u^{co}(Y_\perp) \quad \text{and} \quad ST_E^{fin}(X,Y) = X \to \mathcal{P}_c^{co}(Y_\perp).$$

In the next section we relate the lower, the upper and the convex state transformers with predicate transformers between affirmative predicates.

6.2 Topological predicate transformers

Since affirmative predicates are identified with the open sets of a topological space, functions from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$ are the appropriate *topological* generalization of predicate transformers. For ordinary predicate transformers, *complete multiplicativity* (preservation of arbitrary meets) is required to rule out those predicate transformers which represent 'imaginary programs' (specifications). In addition, Scott continuity is required on predicate transformers to characterize 'computable' programs. While the latter constraint can be easily exported to our topological generalization of predicate transformers (open sets are closed under arbitrary unions), the condition of complete multiplicativity requires more attention: open sets are not closed under arbitrary intersections.

6.2.1. Definition. Let X and Y be two topological spaces. A function π from the lattice of opens $\mathcal{O}(Y)$ to the lattice of opens $\mathcal{O}(X)$ is said to be *M-multiplicative* if whenever $\bigcap P \subseteq \bigcap Q$ then also $\bigcap \pi(P) \subseteq \bigcap \pi(Q)$, for all $P, Q \subseteq \mathcal{O}(Y)$. The collection of all M-multiplicative functions from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$ is denoted by $\mathcal{O}(Y) \to_M \mathcal{O}(X)$.

Intuitively, an M-multiplicative predicate transformer preserves specifications: if a specification Q on the output space of some program (denoted by an M-multiplicative predicate transformer π) is refined by another specification P, then every input x which makes the output of the program satisfy P should also make the the output of the program satisfy Q.

One can easily verify that M-multiplicative functions are monotone with respect to subset inclusion. Moreover, they preserve all intersections of open sets which are open. Since in every space the empty intersection is the top element in the lattice of open sets, M-multiplicative functions are top-preserving. For M-multiplicative functions we can prove the following *stability lemma* which generalizes Lemma 3.3.5.

6.2.2. Lemma. *Given two spaces X and Y, let $\pi : \mathcal{O}(Y) \to_M \mathcal{O}(X)$ be an M-multiplicative function. Then*

$$x \in \pi(u) \quad \textit{if and only if} \quad \bigcap\{o \in \mathcal{O}(Y) \mid x \in \pi(o)\} \subseteq u$$

for every u open in Y and $x \in X$.

Proof: The direction from left to right is obvious. For the converse we use M-multiplicativity: if $\bigcap\{o \in \mathcal{O}(Y) \mid x \in \pi(o)\} \subseteq u$ then $\bigcap\{\pi(o) \mid x \in \pi(o)\} \subseteq \pi(u)$. Hence $x \in \pi(u)$. □

The M-multiplicative functions arise naturally from upper semi-continuous multifunctions. If $f : X \to Y$ is an upper semi-continuous multifunction, then its upper inverse $f^+ : \mathcal{O}(Y) \to \mathcal{O}(X)$ is an M-multiplicative predicate transformer. Assume $\bigcap P \subseteq \bigcap Q$ for P and Q arbitrary collections of opens of Y, and let $x \in \bigcap\{f^+(o) \mid o \in P\}$. Then $f(x) \subseteq o$ for all $o \in P$ and hence $f(x) \subseteq o$ for all $o \in Q$. Therefore $x \in \bigcap\{f^+(o) \mid o \in Q\}$, which proves f^+ is M-multiplicative.

Dually, if $f : X \to Y$ is a lower semi-continuous multifunction then its lower inverse $f^- : \mathcal{O}(Y) \to \mathcal{O}(X)$ preserves all unions, that is, f^- is *completely additive.* The collection of all completely additive functions from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$ is denoted by $\mathcal{O}(Y) \to_A \mathcal{O}(X)$. For completely additive functions we have the following stability lemma.

6.2.3. Lemma. *Given two spaces X and Y, let $\pi : \mathcal{O}(Y) \to_A \mathcal{O}(X)$ be a completely additive function. Then*

$$x \notin \pi(u) \quad \textit{if and only if} \quad u \subseteq \bigcup\{o \in \mathcal{O}(Y) \mid x \notin \pi(o)\}$$

for every u open in Y and $x \in X$.

Proof: The direction from left to right is obvious. Conversely, if $u \subseteq \bigcup\{o \in \mathcal{O}(Y) \mid x \notin \pi(o)\}$ then $\pi(u) \subseteq \bigcup\{\pi(o) \mid x \notin \pi(o)\}$ because π is completely additive (and hence also monotone). Therefore $x \notin \pi(u)$. □

Notice that the completely additive functions are about refutative predicates. Next we provide duality results between state transformers and topological predicate transformers. They extend the results of Chapter 3 to arbitrary topological spaces. Both the order-theoretic and the metric state transformers are instances of topological state transformers. Since there is a rich semantical theory for order-based state transformers as well as for metric-based state transformers, the dualities below give an indirect way to define predicate transformer semantics for programming languages.

Lower state transformers

Lower state transformers are related to completely additive predicate transformers. The isomorphism below can be used to give a semantic interpretation of one domain in terms of the other. The mapping γ from state transformers to predicate transformers explains that lower state transformers model nondeterministic computations which 'may' satisfy an affirmative predicate. Conversely, the map γ^{-1} from predicate transformers to state transformers tells us that completely additive predicate transformers are about safety: a state x satisfies $\pi(P)$ if the computation represented by π at input x is guaranteed not to terminate in a state not satisfying the affirmative predicate P.

6.2.4. Theorem. *Let X and Y be two topological spaces. The poset of lower state transformers $X \to \mathcal{P}_l(Y)$ is order isomorphic to the poset of completely additive functions $\mathcal{O}(Y) \to_A \mathcal{O}(X)$.*

Proof: We use Proposition 6.1.4. For every continuous function $f : X \to \mathcal{P}_l(Y)$ and completely additive predicate transformer $\pi : \mathcal{O}(Y) \to_A \mathcal{O}(X)$ define the maps $f \mapsto \gamma(f)$ and $\pi \mapsto \gamma^{-1}(\pi)$ by

$$\begin{array}{lcll} \gamma(f) & = & \lambda o \in \mathcal{O}(Y).\{x \in X \mid f(x) \cap o \neq \emptyset\} & \text{and} \\ \gamma^{-1}(\pi) & = & \lambda x \in X.\, Y \setminus \bigcup\{o \in \mathcal{O}(Y) \mid x \notin \pi(o)\}. & \end{array}$$

First note that $\gamma(f)(o) = f^{-}(o)$. Hence $\gamma(f)$ is completely additive and, because f is lower semi-continuous as multifunction, well-defined. To prove that $\gamma^{-1}(\pi)$ is lower semi-continuous we see that clearly $\gamma^{-1}(\pi)(x)$ is a closed subset of Y for every $x \in X$, and moreover, for every $o \in \mathcal{O}(Y)$,

$$\begin{array}{lcll} (\gamma^{-1}(\pi))^{-}(o) & = & \{x \in X \mid \gamma^{-1}(\pi)(x) \cap o \neq \emptyset\} & \\ & = & \{x \in X \mid o \not\subseteq \bigcup\{u \in \mathcal{O}(Y) \mid x \notin \pi(u)\}\} & \\ & = & \{x \in X \mid x \in \pi(o)\} & \text{[Lemma 6.2.3]} \\ & = & \pi(o). & \end{array}$$

Since $\pi(o)$ is open in X, $\gamma^{-1}(\pi)$ is lower semi-continuous. Thus it is well-defined. The above also proves that γ^{-1} is a right inverse of γ. It is also a left inverse because, for every $x \in X$,

$$\begin{aligned}
\gamma^{-1}(\gamma(f))(x) &= Y \setminus \bigcup\{o \in \mathcal{O}(Y) \mid x \notin \gamma(f)(o)\} \\
&= Y \setminus \bigcup\{o \in \mathcal{O}(Y) \mid f(x) \cap o = \emptyset\} \\
&= \bigcap\{c \in \mathcal{C}(Y) \mid f(x) \subseteq c\} \\
&= f(x),
\end{aligned}$$

where the latter equality follows because $f(x)$ is closed in Y. Preservation of orders is immediate. □

If a continuous function $f : X \to \mathcal{P}_l(Y)$ is non-empty for all $x \in X$, then $\gamma(f)$ is strict, whereas, if a completely additive predicate transformer $\pi : \mathcal{O}(Y) \to_A \mathcal{O}(X)$ is strict then $\gamma^{-1}(\pi)(x) \neq \emptyset$ for all $x \in X$.

The following corollary restricts the above duality to a finitary one for locally open compact spaces.

6.2.5. Corollary. *Let X and Y be two locally open compact spaces. The poset of lower state transformers $X \to \mathcal{P}_l(Y)$ is order isomorphic to the poset of finitely additive functions in $\mathcal{KO}(Y) \to \mathcal{KO}(X)$.*

Proof: Since X and Y are locally open compact, the collections of their compact open sets form bases for their respective topologies. Moreover, because a function preserves all joins if and only if it preserves all the directed joins and the finite ones, the order isomorphism of Theorem 6.2.4 cuts down to an order isomorphism between $X \to \mathcal{P}_l(Y)$ and the finite unions preserving functions in $\mathcal{KO}(Y) \to \mathcal{KO}(X)$. □

A natural question is whether the locally open compact spaces are closed under the lower powerspace construction. The answer is given in the following proposition which is similar to Proposition 6.11 in [Sch93].

6.2.6. Proposition. *If X is a locally open compact space then so is $\mathcal{P}_l(X)$.*

Proof: Let X be a locally open compact space, and let $A \in \mathcal{P}_l(X)$ be such that $A \in L_{o_1} \cap ... \cap L_{o_n}$, where all o_i's are open subsets of X. In order to show that $\mathcal{P}_l(X)$ is also locally open compact we have to find an open compact set of $\mathcal{P}_l(X)$ containing A as element and contained in $L_{o_1} \cap ... \cap L_{o_n}$. Since $A \in L_{o_1} \cap ... \cap L_{o_n}$ we can find $x_i \in A \cap o_i$. By locally open compactness of X we can therefore find compact open subsets u_i of X such that $x_i \in u_i \subseteq o_i$ for all $i \in \{1, ..., n\}$. Consider the open set $L_{u_1} \cap ... \cap L_{u_n}$ of $\mathcal{P}_l(X)$. By construction $A \in L_{u_1} \cap ... \cap L_{u_n} \subseteq L_{o_1} \cap ... \cap L_{o_n}$. It remains to prove the compactness of $L_{u_1} \cap ... \cap L_{u_n}$. Using Proposition 5.3.4 (Alexander sub-basis theorem) it is enough to find a finite subset $K \subseteq J$ for every index set J such that $L_{u_1} \cap ... \cap L_{u_n} \subseteq \bigcup_J L_{o_j}$, where all o_j's are open subsets in X. If

$L_{u_1} \cap ... \cap L_{u_n} \subseteq \bigcup_J L_{o_j}$ then $u_1 \cup ... \cup u_n \subseteq \bigcup_J o_j$. Hence, by compactness of u_i it follows that there exists a finite index set $K \subseteq J$ such that $u_1 \cup ... \cup u_n \subseteq \bigcup_K o_j$. Therefore $L_{u_1} \cap ... \cap L_{u_n} \subseteq \bigcup_K L_{o_j}$, from which the required compactness follows. $\square$

Closure properties of the lower space construction have been extensively studied by Schalk in her thesis [Sch93]. Using the lower powerlocale as defined in [Rob88], Schalk [Sch93, Proposition 6.26] proved that sober spaces are closed under the lower space construction. Algebraic cpo's taken with the Scott topology are sober and locally open compact (see Chapter 5). What is the connection between the lower space and the Hoare powerdomain of an algebraic cpo? For ω-algebraic cpo's the question has been answered by Smyth [Smy83b], whereas for more general (continuous) domains the answer can be found in [Sch93, AJ94, Mis95].

6.2.7. Proposition. *For an algebraic cpo X, the Hoare powerdomain $\mathcal{H}(X)$ taken with the Scott topology is isomorphic in* **Sp** *to the non-empty lower space $\mathcal{P}_l(X) \setminus \{\emptyset\}$.* $\square$

Since continuous functions between two algebraic cpo's X and Y with the Scott topology are exactly the functions preserving the least upper bounds of directed sets, from Theorem 6.2.4 and the above proposition we have the following duality. The poset of Scott continuous functions $X \to \mathcal{H}(Y)$ is order isomorphic to the poset of all strict and completely additive functions from the lattice of Scott opens $\mathcal{O}(Y)$ to the lattice of Scott opens $\mathcal{O}(X)$. If X and Y are SFP-domains, then they are spectral in the Scott topology. Hence, by Corollary 6.2.5, the poset of Scott continuous functions $X \to \mathcal{H}(Y)$ is order isomorphic to the poset of strict and finitely additive functions from the distributive lattice of Scott compact opens $\mathcal{KO}(Y)$ to the lattice of Scott compact opens $\mathcal{KO}(X)$.

Let Y be a metric space taken with the metric topology. By definition, the underlying set of the lower space $\mathcal{P}_l(Y)$ coincides with that of the closed powerdomain $\mathcal{P}_{cl}(Y)$. If X is any discrete metric space, then every function from X to $\mathcal{P}_l(Y)$ is lower semi-continuous. By Theorem 6.2.4, the set of all functions $X \to \mathcal{P}_{cl}(Y)$ is isomorphic to the set of all completely additive functions from the lattice of metric opens $\mathcal{O}(Y)$ to the lattice of metric opens $\mathcal{O}(X)$ (since X is discrete, the latter coincides with the discrete topology on X). In case both X and Y are compact ultra-metric spaces (and hence Stone spaces in their metric topology) we can use the characterization of Corollary 6.2.5. Notice that the lower topology on $\mathcal{P}_{cl}(Y)$ (which is $\mathcal{T}_0$) does not coincide with the metric topology (which is $\mathcal{T}_2$). We need to consider non-symmetric metric spaces. For ω-algebraic complete quasi metric spaces a result which generalizes Proposition 6.2.7 is presented in [BBR96a].

Upper state transformers

Next we give a duality between upper state transformers and M-multiplicative predicate transformers. Intuitively, upper state transformers are models for non-deterministic computations of which the outputs 'must' satisfy a given affirmative predicate. For an M-multiplicative predicate transformer π, a state x satisfies $\pi(P)$ if the computation represented by π at input x is guaranteed to terminate in a state satisfying the affirmative predicate P.

According to the informal definition of safety and liveness predicates given in [Lam77], an arbitrary predicate can always be expressed as the intersection of a safety and a liveness predicate. This fact leads [AS85] to the topological definition of safety predicates as closed subsets, whereas a liveness predicate can be identified with a *dense subset* (the complement does not contain non-empty open sets). It is not hard to see that in any topological space X, any subset of X can be expressed as the intersection of a closed set with a dense one.

Since we are concerned with affirmative and refutative predicates, it is clear that M-multiplicative predicate transformers are not liveness predicate transformers in the sense of [Lam77].

6.2.8. Theorem. *Let X and Y be two topological spaces. The poset of upper state transformers $X \to \mathcal{P}_u(Y)$ is order isomorphic to the poset of M-multiplicative functions $\mathcal{O}(Y) \to_M \mathcal{O}(X)$.*

Proof: The proof is similar to that of Theorem 6.2.4, making use of Proposition 6.1.4. For every continuous function $f : X \to \mathcal{P}_u(Y)$ and M-multiplicative predicate transformer $\pi : \mathcal{O}(Y) \to_M \mathcal{O}(X)$ define maps $f \mapsto \omega(f)$ and $\pi \mapsto \omega^{-1}(\pi)$ by

$$\begin{array}{lcll} \omega(f) & = & \lambda o \in \mathcal{O}(Y).\{x \in X \mid f(x) \subseteq o\} & \text{and} \\ \omega^{-1}(\pi) & = & \lambda x \in X. \bigcap\{o \in \mathcal{O}(Y) \mid x \in \pi(o)\}. & \end{array}$$

For every open o of Y, $\omega(f)(o) = f^+(o)$. Hence $\omega(f)$ is M-multiplicative and, because f is upper semi-continuous as multifunction, well-defined. To prove that $\omega^{-1}(\pi)$ is well-defined, observe that an arbitrary intersection of open sets is upper closed with respect to the specialization order, and, for every $o \in \mathcal{O}(Y)$,

$$\begin{array}{lcll} (\omega^{-1}(\pi))^+(o) & = & \{x \in X \mid \omega^{-1}(\pi)(x) \subseteq o\} & \\ & = & \{x \in X \mid \bigcap\{o \in \mathcal{O}(Y) \mid x \in \pi(o)\} \subseteq o\} & \\ & = & \{x \in X \mid x \in \pi(o)\} & \text{[Lemma 6.2.2]} \\ & = & \pi(o). & \end{array}$$

Since $\pi(o)$ is open in X, $\omega^{-1}(\pi)$ is upper semi-continuous. Thus it is well-defined. The above also proves that ω^{-1} is a right inverse of ω. It is also a left inverse because, for every $x \in X$,

$$\begin{aligned}\omega^{-1}(\omega(f))(x) &= \bigcap\{o \in \mathcal{O}(Y) \mid x \in \omega(f)(o)\} \\ &= \bigcap\{o \in \mathcal{O}(Y) \mid f(x) \subseteq o\} \\ &= f(x),\end{aligned}$$

where the latter equality follows because $f(x)$ is upper closed in Y, and hence a saturated set. Preservation of orders is immediate. □

As for the case of lower state transformers, if we exclude the empty set as possible output result of an upper state transformer then the corresponding restriction on M-multiplicative predicate transformers is strictness.

For every space X, the underlying set of the upper space $\mathcal{P}_u(X)$ ordered by the specialization preorder is a complete lattice. If X is a $\mathcal{T}_1$ space, then the underlying set of $\mathcal{P}_u(X)$ coincides with the full powerset of X because every set is upper closed.

What restrictions are needed on the underlying space in order that the compact restriction of the upper powerspace is a dcpo or an algebraic dcpo? For a dcpo the question has been answered in [HM81]: the underlying space should be sober. This is proved using a bijective correspondence between the elements of the compact upper powerspace (compact saturated sets) and the Scott open filters of the lattice of opens sets (for a proof of this statement we refer the reader to Corollary 9.3.11).

6.2.9. Proposition. *Let X be a sober space. If S is an arbitrary collection of compact saturated subsets of X directed with respect to superset inclusion then $\bigcap S$ is also saturated and compact. Moreover, for any open $o \in \mathcal{O}(X)$, if $\bigcap S \subseteq o$ then there exists $q \in S$ such that $q \subseteq o$.* □

The first statement of the above proposition gives soberness as a sufficient condition for the compact upper powerspace to be a dcpo (more generally, Schalk proved that if a space is sober then so is its non-empty compact upper space [Sch93, Lemma 7.20]). The second statement says that compact opens are compact elements for the dcpo $\mathcal{P}_u^{co}(X)$. However this dcpo need not to be algebraic. The algebraicity is obtained by restricting the attention to sober and locally open compact spaces.

6.2.10. Lemma. *Let X be a sober locally open compact space. The underlying set of the compact upper space $\mathcal{P}_u^{co}(X)$ ordered by the specialization order is an algebraic dcpo with as compact elements the compact open sets. Moreover, the Scott topology on $\mathcal{P}_u^{co}(X)$ coincides with the upper topology.*

Proof: We need to prove that every compact saturated set q can be expressed as least upper bound of the compact open sets below q. Because X is locally open compact, every open set can be obtained as a directed union of compact opens. Hence, if q is a compact saturated set such that $q \subseteq o$ for some open set o, then there exists a compact open u such that $q \subseteq u \subseteq o$. For a compact saturated sets q, this implies

$$q = \bigcap\{o \in \mathcal{O}(X) \mid q \subseteq o\} = \bigcap\{u \in \mathcal{KO}(X) \mid q \subseteq u\}.$$

Hence the collection of compact saturated sets is an algebraic dcpo when ordered by superset inclusion.

Next we prove that the Scott topology and the upper topology on $\mathcal{P}_u(X)$ coincide. The upper closure of a compact open o in $\mathcal{P}_u(X)$ is a basic open for the Scott topology, and by definition it coincides with the basic open $U_o = \{q \mid q \subseteq o\}$ of the upper topology. Hence the Scott topology on $\mathcal{P}_u(X)$ is included in the upper topology. Conversely, let $o \in \mathcal{O}(X)$ and consider the basic open set U_o of the upper topology on $\mathcal{P}_u(X)$. It is clearly upper closed, and if $S \subseteq \mathcal{P}_u(X)$ is a directed set such that $\bigcap S \in U_o$ then, by Corollary 6.2.9, there exists $q \in S$ such that $q \subseteq o$. Therefore U_o is Scott open. □

Since algebraic dcpo's taken with the Scott topology are sober, the above lemma implies that the compact upper space of a locally open compact sober space is again sober. In particular, if X is an algebraic cpo, then so is the poset of all Scott compact saturated subsets of X ordered by superset inclusion. The following characterization theorem can be found in [Smy83b] for ω-algebraic cpo's, and in [AJ94, Mis95] for the general case.

6.2.11. Proposition. *Let X be an algebraic cpo taken with the Scott topology. The Smyth powerdomain $\mathcal{S}(X)$ together with its Scott topology is isomorphic in* **Sp** *to the non-empty, compact upper powerspace $\mathcal{P}_u^{co}(X) \setminus \{\emptyset\}$.* □

In order to apply the isomorphism of Theorem 6.2.8 to upper state transformers with values in an upper compact subset, we need to find a corresponding restriction on the predicate transformer side. The definition of compact sets as finitarily specifiable theory introduced in Chapter 5 is of help here.

6.2.12. Theorem. *Let X and Y be two topological spaces. The isomorphism of Theorem 6.2.8 restricts to an order isomorphism between the poset of compact upper state transformers $X \to \mathcal{P}_u^{co}(Y)$ and the poset of M-multiplicative and Scott continuous functions $\mathcal{O}(Y) \to_{c,M} \mathcal{O}(X)$.*

Proof: Let $f \in X \to \mathcal{P}_u^{co}(Y)$. Also let $\pi : \mathcal{O}(Y) \to_{c,M} \mathcal{O}(X)$ be a Scott continuous function. We need to prove $\omega(f)$ Scott continuous and $\omega^{-1}(\pi)(x)$ compact for all $x \in X$. Let S be a directed subset of opens in Y.

If $x \in \omega(f)(\bigcup S)$ then $f(x) \subseteq \bigcup S$. By compactness of $f(x)$ it follows that $f(x) \subseteq o$ for some $o \in S$. Therefore $x \in \bigcup\{\omega(f)(o) \mid o \in S\}$. Since $\omega(f)$ is monotone, being M-multiplicative, it follows that $\omega(f)$ is Scott continuous.

Take now $x \in X$ and assume $\omega^{-1}(\pi)(x) \subseteq \bigcup S$. By Lemma 6.2.2 then $x \in \pi(\bigcup S)$. Since π is Scott continuous, there exists $o \in S$ such that $x \in \pi(o)$. Using Lemma 6.2.2 again it follows that $\omega^{-1}(\pi)(x) \subseteq o$, that is $\omega^{-1}(\pi)(x)$ is compact. $\Box$

Using Corollary 9.3.11, Smyth [Smy83b] was the first who realized that for a sober space Y, the poset of upper state transformers $X \to \mathcal{P}_u^{co}(Y)$ is order isomorphic to the poset of finitely multiplicative and Scott continuous functions in $\mathcal{O}(Y) \to \mathcal{O}(X)$. In the above theorem, we do not have the requirement of Y being sober, but we consider M-multiplicativity instead of finitely multiplicativity. Hence, if Y is a sober space then a (Scott-)continuous function in $\mathcal{O}(Y) \to \mathcal{O}(X)$ is finite multiplicative if and only if it is M-multiplicative. Best [Bes83] has proved a similar result for countable flat cpo's.

6.2.13. Corollary. *Let X and Y be two sets and let $\pi : \mathcal{P}(Y) \to \mathcal{P}(X)$ Scott continuous. If π preserves binary intersections then it preserves all non-empty intersections.*

Proof: Consider the flat dcpo $Y_\bot$ taken with the Scott topology. Notice that the latter equals $\mathcal{P}(Y) \cup \{Y_\bot\}$. Hence we can extend π to a Scott-continuous function from $\mathcal{O}(Y_\bot) \to \mathcal{P}(X)$ by mapping $\pi(Y_\bot) = X$. If π preserves binary intersections then its extension preserves all finite intersections (being top preserving). Since $Y_\bot$ is a sober space, the extension of π is M-multiplicative. Hence $\pi : \mathcal{P}(Y) \to \mathcal{P}(X)$ preserves all non-empty intersections. $\Box$

Another consequence of the combination of the result of Smyth [Smy83b] and Theorem 6.2.12 is the following.

6.2.14. Corollary. *Let X and Y be two spectral spaces. The poset of compact upper state transformers $X \to \mathcal{P}_u^{co}(Y)$ is order isomorphic to the poset of finitely multiplicative functions in $\mathcal{KO}(Y) \to \mathcal{KO}(X)$.*

Proof: Since Y is spectral, it is also sober. Moreover the intersection of compact opens is compact open by definition. Hence every Scott continuous and M-multiplicative function $\pi : \mathcal{O}(Y) \to \mathcal{O}(X)$ restricts to a finite meet preserving function in $\mathcal{KO}(Y) \to \mathcal{KO}(X)$. Conversely, every finite meet preserving function $\pi : \mathcal{KO}(Y) \to \mathcal{KO}(X)$ extends by means of ideal completion uniquely to a Scott continuous and finite meet preserving function in $\mathcal{O}(Y) \to \mathcal{O}(X)$ the restriction of which to compact sets is exactly π. Since Y is sober, this extension of π is M-multiplicative. $\Box$

Let X and Y be two algebraic cpo's. From Theorem 6.2.8 and Proposition 6.2.11 we have that the poset of Scott continuous functions from $X \to \mathcal{S}(Y)$ is order isomorphic to the poset of strict, Scott continuous and finite multiplicative functions from the lattice of Scott opens $\mathcal{O}(Y)$ to the lattice of Scott opens

$\mathcal{O}(X)$. Moreover, if X and Y are SFP-domains, then they are spectral in the Scott topology. Hence, by Corollary 6.2.14, the poset of Scott continuous functions from $X \to \mathcal{S}(Y)$ is order isomorphic to the poset of strict and finitely multiplicative functions from the distributive lattice of Scott compact opens $\mathcal{KO}(Y)$ to the lattice of Scott compact opens $\mathcal{KO}(X)$.

Let X be a discrete metric space and let Y be a metric space. Thus every function from X to $\mathcal{P}_u^{co}(Y)$ is continuous. Notice that the underlying sets of $\mathcal{P}_u^{co}(Y)$ and of the metric compact powerdomain $\mathcal{P}_{co}(Y)$ coincide. Therefore, by Theorem 6.2.12, the set of all functions $X \to \mathcal{P}_{co}(Y)$ is isomorphic to the set of all Scott continuous and finitely multiplicative functions from the lattice of metric opens $\mathcal{O}(Y)$ to the lattice of metric opens $\mathcal{O}(X)$. In case both X and Y are compact ultra-metric spaces, the set $X \to \mathcal{P}_{co}(Y)$ is isomorphic to the set of strict and finitely multiplicative functions from the distributive lattice of metric compact opens $\mathcal{KO}(Y)$ to the lattice of metric compact opens $\mathcal{KO}(X)$.

6.3 Pairs of predicate transformers

In Chapter 3 we have seen that the Egli-Milner state transformers are dual to the Nelson predicate transformers. The natural topological generalization of the Egli-Milner state transformers are the convex state transformers. In order to generalize the Nelson predicate transformers we need to consider pairs $\langle \pi, \rho \rangle$ of topological predicate transformers, where π is M-multiplicative and ρ is completely additive. In this way we can model both the positive and the negative information about a computation. However, we have to restrict our considerations to those pairs $\langle \pi, \rho \rangle$ of predicate transformers which represent the same computation. What we need is a stability lemma similar to Lemma 6.2.2 and Lemma 6.2.3. The definition below is inspired by the work of Johnstone [Joh82b] on the Vietoris powerlocale.

6.3.1. Definition. Given two spaces X and Y, a pair $\langle \pi, \rho \rangle$ of functions from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$ is said to be *jointly multiplicative* if π is M-multiplicative, ρ is completely additive, and

(i) $\pi(o_1 \cup o_2) \subseteq \pi(o_1) \cup \rho(o_2)$, and
(ii) $(\bigcap S \cap o_1) \subseteq o_2$ implies $(\bigcap\{\pi(o) \mid o \in S\} \cap \rho(o_1)) \subseteq \rho(o_2)$.

for opens o_1, o_2 of Y and $S \subseteq \mathcal{O}(Y)$. Jointly multiplicative functions are ordered componentwise by the extension to functions of subset inclusion.

For a pair $\langle \pi, \rho \rangle$ of jointly multiplicative functions, according to the above definition there are two 'non-observable' requirements (in the sense that they involve sets which need not to be open): the M-multiplicativity of π and the second condition of joint multiplicativity. In the previous section we have shown that if π is Scott continuous and the space Y is sober, then M-multiplicativity is equivalent to finite multiplicativity. The latter is clearly an observable and finitary requirement.

The non-observability of condition (ii) of Definition 6.3.1 is more delicate. In locale theory, for the construction of the Vietoris powerlocale the following condition is required [Joh82b] instead of (ii),

$$\pi(o_1) \cap \rho(o_2) \subseteq \rho(o_1 \cap o_2) \tag{6.1}$$

for all $o_1, o_2 \in \mathcal{O}(Y)$. Notice that (i) of Definition 6.3.1 and the above (6.1) are the modal axioms relating the $\Box$ and $\Diamond$ operators in negation free modal logic (often called Hennessy-Milner logic) [HM85].

For all spaces X and Y, if $\langle \pi, \rho \rangle$ is a jointly multiplicative pair of functions from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$ then (6.1) clearly holds. The converse holds if we restrict Y to be a coherent space and π to be Scott continuous.

6.3.2. Lemma. *Let X and Y be two spaces such that Y is coherent. For every pair $\langle \pi, \rho \rangle$ of Scott continuous functions from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$ such that π preserves finite meets, and ρ preserves finite joins, the following two statements are equivalent:*

(i) $\pi(o_1) \cap \rho(o_2) \subseteq \rho(o_1 \cap o_2)$ *for all* $o_1, o_2 \in \mathcal{O}(Y)$*;*

(ii) $(\bigcap S \cap o_1) \subseteq o_2$ *implies* $(\bigcap\{\pi(o) \mid o \in S\} \cap \rho(o_1)) \subseteq \rho(o_2)$ *for all* $o_1, o_2 \in \mathcal{O}(Y)$ *and* $S \subseteq \mathcal{O}(Y)$.

Moreover, if Y is spectral then both (i) *and* (ii) *are equivalent to*

(iii) $\pi(o_1) \cap \rho(o_2) \subseteq \rho(o_1 \cap o_2)$ *for all* $o_1, o_2 \in \mathcal{KO}(Y)$.

Proof: Obviously (ii) implies (i). Hence we concentrate on the opposite direction. Assume $\pi(o) \cap \rho(o') \subseteq \rho(o \cap o')$ for all opens o and o' of Y. Let $S \subseteq \mathcal{O}(Y)$ and $o_1, o_2 \in \mathcal{O}(Y)$. Because Y is coherent, every open set o of Y is the union of all the compact saturated subsets q of Y such that there exists $u \in \mathcal{O}(Y)$ with $u \subseteq q \subseteq o$. Hence the set $\bigcap S \cap o_1$ is equivalent to the set

$$\bigcap\{q \in \mathcal{KQ}(Y) \mid \exists u \in \mathcal{O}(Y){:}\, \exists o \in S \cup \{o_1\}{:}\, u \subseteq q \subseteq o\}.$$

Next, we use the fact that Y is sober and that compact saturated sets are closed under finite intersections to reformulate Proposition 6.2.9 as follows. Whenever the intersection of compact saturated sets is contained in an open set then the same is true for an intersection of finitely many of them. This fact justifies that $(\bigcap S \cap o_1) \subseteq o_2$ implies that there exist finitely many compact saturated sets $q_1, ... q_n$ such that $q_1 \cap ... \cap q_n \subseteq o_2$, with $u_i \subseteq q_i \subseteq o_i$ for some $o_i \in S \cup \{o_1\}$ and open $u_i \in \mathcal{O}(Y)$. Hence $(q_1 \cap ... \cap q_n) \cap o_1 \subseteq o_2$, where, without loss of generality, we can assume, for all $1 \leq i \leq n$, $u_i \subseteq q_i \subseteq o_i$ for some $o_i \in S$ and $u_i \in \mathcal{O}(Y)$.

Let $u = u_1 \cap ... \cap u_n$. Since $u_1, .., u_n$ are finitely many open sets, u is also an open set. Moreover $u \cap o_1 \subseteq o_2$ because $u \subseteq q_1 \cap ... \cap q_n$. By our assumption on the pair $\langle\pi,\rho\rangle$, $\pi(u)\cap\rho(o_1)\subseteq\rho(u\cap o_1)$. But ρ is monotone and $u\cap o_1\subseteq o_2$. Thus $\pi(u)\cap\rho(o_1)\subseteq\rho(o_2)$. Notice that $\bigcap S\subseteq u$ because, for all $1\leq i\leq n$, $o_i\in S$ and

$$o_i=\bigcup\{u\in\mathcal{O}(Y)\mid\exists q\in\mathcal{KQ}(Y)\colon u\subseteq q\subseteq o_i\}$$

as Y is coherent. Thus $\bigcap\{\pi(o)\mid o\in S\}\subseteq\pi(u)$, from which follows that $(\bigcap\{\pi(o)\mid o\in S\}\cap\rho(o_1))\subseteq\rho(o_2)$.

Assume now Y is a spectral space. We prove that (iii) implies (i). The other direction follows immediately.

Let o_1 and o_2 be two open sets of Y. Because Y is spectral they can be written as directed union of all compact open sets below them. Below, let u and v range over compact open sets. Because π and ρ are both Scott continuous, we have

$$\begin{aligned}\pi(o_1)\cap\rho(o_2) &= \pi(\bigcup\{u\mid u\subseteq o_1\})\cap\rho(\bigcup\{v\mid v\subseteq o_2\})\\ &= \bigcup\{\pi(u)\mid u\subseteq o_1\}\cap\bigcup\{\rho(v)\mid v\subseteq o_2\}\\ &= \bigcup\{\pi(u)\cap\rho(v)\mid u\subseteq o_1\ \&\ v\subseteq o_2\}\\ &\subseteq \bigcup\{\rho(u\cap v)\mid u\cap v\subseteq o_1\cap o_2\}\quad\text{[by (iii)]}\\ &= \rho(o_1\cap o_2).\end{aligned}$$

Since spectral spaces are coherent, (i) is equivalent to (ii). Hence (iii) implies both (i) and (ii). □

As a consequence, if Y is a coherent space then the jointly multiplicative and Scott continuous predicate transformers $\langle\pi,\rho\rangle$ from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$ can be described using only open sets, substituting finite multiplicativity for M-multiplicativity, and condition (ii) of Definition 6.3.1 by the equivalent condition (6.1).

6.3.3. Lemma. *Let X and Y be two spaces such that Y is coherent. The poset of all jointly multiplicative and Scott continuous predicate transformers from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$ is a cpo.*

Proof: Let $D=\{\langle\pi_i,\rho_i\rangle\mid i\in I\}$ be a directed set of jointly multiplicative and Scott continuous functions from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$. Define $\pi(o)=\bigcup_I\pi_i(o)$ and $\rho(o)=\bigcup_I\rho_i(o)$ for every open o of Y. By Proposition 6.2.9 and Theorem 6.2.12 the function π is M-multiplicative and Scott continuous. Thus π is the least upper bound of all π_i's. The function ρ is completely additive by definition, and hence is the least upper bound of all ρ_i's. We need to prove that $\langle\pi,\rho\rangle$ is a jointly multiplicative pair.

Let o_1 and o_2 be two open sets of Y. If $x\in\pi(o_1\cup o_2)$ then there exists $k\in I$ such that $x\in\pi_k(o_1\cup o_2)$. Since $\langle\pi_k,\rho_k\rangle$ is jointly multiplicative, $\pi_k(o_1\cup o_2)\subseteq$

$\pi_k(o_1) \cup \rho_k(o_2)$. But $\pi_k(o_1) \subseteq \pi(o_1)$ and $\rho_k(o_2) \subseteq \rho(o_2)$. Thus $x \in \pi(o_1) \cup \rho(o_2)$, that is, condition (i) of Definition 6.3.1 holds.

Because Y is a coherent space, by Lemma 6.3.2, condition (ii) of Definition 6.3.1 is equivalent to the finitary condition (6.1). Assume o_1 and o_2 are two open sets of Y and let $x \in \pi(o_1) \cap \rho(o_2)$. By directness of the set D and the definitions of π and ρ, there exists $k \in I$ such that $x \in \pi_k(o_1)$ and $x \in \rho_k(o_2)$. Since $\langle \pi_k, \rho_k \rangle$ is jointly multiplicative, $\pi_k(o_1) \cap \rho_k(o_2) \subseteq \rho_k(o_1 \cap o_2)$. Thus $x \in \rho_k(o_1 \cap o_2) \subseteq \rho(o_1 \cap o_2)$.

Therefore $\langle \pi, \rho \rangle$ is jointly multiplicative and the poset of all jointly multiplicative and Scott continuous predicate transformers from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$ is a dcpo. The pair of functions mapping every open set of Y to the empty set is jointly multiplicative and Scott continuous. Hence they form the bottom element of the dcpo of jointly multiplicative and Scott continuous predicate transformers. $\Box$

We are interested in jointly multiplicative predicate transformers because they represent the positive and the negative information of the same computation, as formally stated in the following stability lemma.

6.3.4. Lemma. *Let X and Y be two spaces and $\langle \pi, \rho \rangle$ be a a pair of jointly multiplicative functions from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$. For $x \in X$, let us denote by $q(x, \pi)$ and $o(x, \rho)$ the sets $\bigcap \{o \in \mathcal{O}(Y) \mid x \in \pi(o)\}$ and $\bigcup \{o \in \mathcal{O}(Y) \mid x \notin \rho(o)\}$, respectively. For every $u \in \mathcal{O}(Y)$ we have*

(i) $x \in \pi(u)$ *if and only if* $q(x, \pi) \cap (Y \setminus o(x, \rho)) \subseteq u$,
(ii) $x \notin \rho(u)$ *if and only if* $q(x, \pi) \cap (Y \setminus o(x, \rho)) \subseteq Y \setminus u$.

Proof: (i) The direction from left to right is trivial and hence omitted. Assume $q(x, \pi) \cap (Y \setminus o(x, \rho)) \subseteq u$. Then $q(x, \pi) \subseteq u \cup o(x, \rho)$. Since $o(x, \rho)$ is open (being union of opens) and π is M-multiplicative, $x \in \pi(u \cup o(x, \rho))$ by Lemma 6.2.2. But $\pi(u \cup o(x, \rho)) \subseteq \pi(o) \cup \rho(o(x, \rho))$ because π and ρ are jointly multiplicative. Since ρ is completely additive we have that $x \notin \rho(o(x, \rho))$ by definition of $o(x, \rho)$. Therefore $x \in \pi(u)$.

(ii) As above, the direction from left to right is trivial. Assume $q(x, \pi) \cap (Y \setminus o(x, \rho)) \subseteq Y \setminus u$. Then $q(x, \pi) \cap u \subseteq o(x, \rho)$, which implies

$$(\bigcap \{\pi(o) \mid x \in \pi(o)\} \cap \rho(u)) \subseteq \rho(o(x, \rho))$$

because $\langle \pi, \rho \rangle$ is jointly multiplicative. Since ρ is completely additive, $x \notin \rho(o(x, \rho))$. But $x \in \bigcap \{\pi(o) \mid x \in \pi(o)\}$, therefore $x \notin \rho(u)$. $\Box$

Next we use the above stability lemma to relate jointly multiplicative predicate transformers and convex state transformers by an isomorphism that generalizes the result in Chapter 3 for Nelson predicate transformers.

6.3.5. Theorem. *Let X and Y be two spaces. The poset of convex state transformers $X \to \mathcal{P}_c(Y)$ is order isomorphic to the poset of all jointly multiplicative pairs of predicate transformers in $\mathcal{O}(Y) \to \mathcal{O}(X)$. Also, the above isomorphism cuts down to an order isomorphism between compact and convex state transformers $X \to \mathcal{P}_c^{co}(Y)$ and the poset of all pairs of Scott continuous functions in $\mathcal{O}(Y) \to \mathcal{O}(X)$ which are jointly multiplicative.*

Proof: For a convex state transformer $f : X \to \mathcal{P}_c(Y)$ define $\eta(f)$ to be the pair of functions from $\mathcal{O}(Y)$ to $\mathcal{O}(X)$

$$\begin{array}{ll} \omega(f) = \lambda o \in \mathcal{O}(Y).\{x \mid f(x) \subseteq o\} & \text{and} \\ \gamma(f) = \lambda o \in \mathcal{O}(Y).\{x \mid f(x) \cap o \neq \emptyset\}. & \end{array}$$

Since f is continuous as a multifunction, both functions above are well-defined. Moreover, $\omega(f)$ is M-multiplicative and $\gamma(f)$ is completely additive. Next we prove they are jointly multiplicative.

Let o_1 and o_2 be two open subsets of Y. If $x \in \omega(f)(o_1 \cup o_2)$ then $f(x) \subseteq o_1 \cup o_2$. Towards a contradiction, assume both $f(x) \not\subseteq o_1$ and $f(x) \cap o_2 = \emptyset$. Then $f(x) \not\subseteq o_1 \cup o_2$, hence the contradiction. Thus $f(x) \subseteq o_1$ or $f(x) \cap o_2 \neq \emptyset$, that is, $x \in \omega(f)(o_1) \cup \gamma(f)(o_2)$.

Let $S \subseteq \mathcal{O}(Y)$ and let o_1, o_2 be two open subsets of Y such that $\bigcap S \cap o_1 \subseteq o_2$. If $x \in \omega(f)(o)$ for all $o \in S$ and $x \in \gamma(f)(o_1)$, then $f(x) \subseteq \bigcap S$ and $f(x) \cap o_1 \neq \emptyset$. Hence there exists $y \in f(x)$ such that $y \in \bigcap S \cap o_1 \subseteq o_2$. It follows that $f(x) \cap o_2 \neq \emptyset$, and hence $x \in \gamma(f)(o_2)$. Therefore the pair $\eta(f) = \langle \omega(f), \gamma(f) \rangle$ is jointly multiplicative.

Consider now the pair $\langle \pi, \rho \rangle$ of jointly multiplicative predicate transformers in $\mathcal{O}(Y) \to \mathcal{O}(X)$. Define $\eta^{-1}(\langle \pi, \rho \rangle)(x)$, for every $x \in X$, by

$$\bigcap\{o \in \mathcal{O}(Y) \mid x \in \pi(o)\} \cap (Y \setminus \bigcup\{o \in \mathcal{O}(Y) \mid x \notin \rho(o)\}). \tag{6.2}$$

We prove that $\eta^{-1}(\langle \pi, \rho \rangle)(x)$ is convex closed. Let *cl* be the closure operator induced by the topology $\mathcal{O}(Y)$. Since $Y \setminus \bigcup\{o \in \mathcal{O}(Y) \mid x \notin \rho(o)\}$ is a closed set,

$$cl(\eta^{-1}(\langle \pi, \rho \rangle)(x)) \subseteq Y \setminus \bigcup\{o \in \mathcal{O}(Y) \mid x \notin \rho(o)\}.$$

Similarly, the upper closure $\uparrow \eta^{-1}(\langle \pi, \rho \rangle)(x)$, with respect to the order induced by $\mathcal{O}(Y)$, is included in the saturated set $\bigcap\{o \in \mathcal{O}(Y) \mid x \in \pi(o)\}$. Hence the convex closure of $\eta^{-1}(\langle \pi, \rho \rangle)(x)$ is included in (6.2). Since the other direction is trivial, $\eta^{-1}(\langle \pi, \rho \rangle)(x)$ is convex closed.

Next we prove that $\eta^{-1}(\langle \pi, \rho \rangle)$ is both upper and lower semi-continuous. For every $o \in \mathcal{O}(Y)$ we have

$$\begin{array}{rcll} \eta^{-1}(\langle \pi, \rho \rangle)^+(o) & = & \{x \in X \mid \eta^{-1}(\langle \pi, \rho \rangle)(x) \subseteq o\} & \\ & = & \{x \in X \mid x \in \pi(o)\} & \text{[Lemma 6.3.4]} \\ & = & \pi(o), & \end{array}$$

and also

$$\begin{array}{rcl}\eta^{-1}(\langle\pi,\rho\rangle)^{-}(o) & = & \{x\in X \mid \eta^{-1}(\langle\pi,\rho\rangle)(x)\cap o\neq\emptyset\}\\ & = & \{x\in X \mid \eta^{-1}(\langle\pi,\rho\rangle)(x)\not\subseteq Y\setminus o\}\\ & = & \{x\in X \mid x\in\rho(o)\} \quad \text{[Lemma 6.3.4]}\\ & = & \rho(o).\end{array}$$

This proves not only that $\eta^{-1}(\langle\pi,\rho\rangle)$ is a convex state transformer, but also that $\eta^{-1}(\langle\pi,\rho\rangle)$ is a right inverse of η. It not hard to see that it is also a left inverse by combining Theorem 6.2.4 and Theorem 6.2.8.

Further, η and η^{-1} are both monotone due to Lemma 6.1.5 and Theorems 6.2.4 and 6.2.8.

By Theorem 6.2.12 and because the intersection of a compact set with a closed one gives again a compact set, it follows that the isomorphism (η,η^{-1}) cuts down to an order-isomorphism between the poset of compact and convex state transformers, and the poset of all pairs of Scott continuous functions in $\mathcal{O}(Y)\to\mathcal{O}(X)$ which are jointly multiplicative. □

As for the cases of the upper space and of the lower space, the above isomorphisms cuts down to a finitary isomorphism if we consider spectral spaces.

6.3.6. Corollary. *Let X and Y be two spectral spaces. The poset of compact convex state transformers $X\to\mathcal{P}^{co}_{c}(Y)$ is order isomorphic to the poset of all pairs $\langle\pi,\rho\rangle$ of functions in $\mathcal{KO}(Y)\to\mathcal{KO}(X)$ such that*

(i) *π is finitely multiplicative;*
(ii) *ρ is finitely additive;*
(iii) *$\pi(o_1\cup o_2)\subseteq\pi(o_1)\cup\rho(o_2)$ for all $o_1,o_2\in\mathcal{KO}(Y)$;*
(iv) *$\pi(o_1)\cap\rho(o_2)\subseteq\rho(o_1\cap o_2)$ for all $o_1,o_2\in\mathcal{KO}(Y)$.*

Proof: Immediate from Corollary 6.2.5, Corollary 6.2.14 and Lemma 6.3.2. □

Despite the mathematical elegance of the presentation of the convex space, it does not have many of the closure properties which the other power constructions enjoy. In general, the underlying set of $\mathcal{P}^{co}_{c}(X)$ taken with the order induced by the Vietoris topology, is not a complete lattice nor a dcpo even if X is an algebraic dcpo with the Scott topology [AJ94, Exercise 11.(e)]. As a consequence, neither sober spaces nor sober and locally open compact spaces are closed under the compact convex space construction. Using the above isomorphism and Lemma 6.3.3 we obtain an easy proof that $\mathcal{P}^{co}_{c}(X)$ is a cpo whenever X is a coherent space. The general situation, i.e. to find a topological characterization of the Plotkin powerdomain, seems to be hopeless. The following characterization theorems can be found in [Smy83b] and [AJ94, Mis95].

6.3.7. Proposition. *Let X be an ω-algebraic cpo taken with the Scott topology. The Plotkin powerdomain $\mathcal{E}(X)$ together with its Scott topology is isomorphic in* **Sp** *to the non-empty, compact convex space $\mathcal{P}_c^{co}(X) \setminus \{\emptyset\}$. The same holds if X is an algebraic cpo such that, when taken with the Scott topology, it forms a coherent space.* $\Box$

From the above proposition and Theorem 6.3.5, we have, for ω-algebraic cpo's X and Y, that the poset of Scott continuous functions $X \to \mathcal{E}(Y)$ is order isomorphic to the poset of all pairs of strict and Scott continuous functions from the lattice of Scott opens $\mathcal{O}(Y)$ to the lattice of Scott opens $\mathcal{O}(X)$ which are jointly multiplicative. If X and Y are SFP domains, then we can apply Corollary 6.3.6 to obtain a finitary duality.

For metric spaces we have the following characterization result [Mic51].

6.3.8. Proposition. *Let X be a metric space taken with the metric topology. The metric compact powerdomain $\mathcal{P}_{co}(X)$ together with the metric topology coincides with the compact convex space $\mathcal{P}_c^{co}(X)$* $\Box$

The above proposition can be applied as follows. If X and Y are two metric spaces, by Theorem 6.3.5, the set of all metric continuous functions $X \to \mathcal{P}_{co}(Y)$ (seen as a discrete poset) is order-isomorphic to the poset of all pairs of Scott continuous functions from the lattice of metric opens $\mathcal{O}(Y)$ to the lattice of metric opens $\mathcal{O}(X)$ which are jointly multiplicative. If X and Y are compact ultra-metric spaces, then in their metric topologies they are Stone spaces. Hence we can apply Corollary 6.3.6 to obtain a finitary duality.

It is easy to see that if X is a set then the set of all finite subsets of X (taken with the discrete topology) coincides with the compact convex powerspace of X. Again, we can apply Theorem 6.3.5 to describe it by jointly multiplicative functions.

6.4 Concluding notes

Dualities for the convex powerspace provide a natural setting for negation-free modal logics (also called Hennessy-Milner logics). Our approach differs from the one taken by Goldblatt [Gol89] and Abramsky [Abr91a] because our axioms relating the $\Box$ operator with the $\Diamond$ operator hold also in an infinitary setting. It is an important topic for further investigation to define an infinitary Hennessy-Milner logic for the convex powerspace.

The results in this chapter are in the concrete framework where predicate transformers are functions between collections of open sets. More abstractly, we could have used frames to represent abstract collections of affirmative predicates by restricting our attention to sober spaces (for the results of last section coherent spaces would be necessary). The duality between frames and sober spaces [Joh82a] could then be used to reconstruct points from frames. In Chapter 8 we discuss an abstract algebraic representation of $\mathcal{T}_0$ spaces. All results in this section can be easily adapted to this algebraic framework.

To fully generalize the results of Part I, it remains a challenge to define a 'meaningful' notion of topological state transformers which are dual to the monotonic (or perhaps Scott continuous) functions between lattices of affirmative predicates. More speculatively, for algebraic cpo's the duality of Chapter 4 seems to suggest the composition of the Smyth with the Hoare powerdomain (or vice-versa, since they commute [Hec91, Lib92]).

Chapter 7

Predicate transformer semantics for concurrency

Topological dualities can be used to define an indirect predicate transformer semantics for programming languages: given a forward semantics, the duality can be used to generate an equivalent backward semantics. A better approach could be the following. Based on computational considerations construct a semantic domain of predicate transformers, and then define semantic operators between predicate transformers corresponding to the syntactic operators. Dualities with state transformers then can be used to prove the correctness of the domain as well as of the semantic operators.

The main contribution of the present chapter is a direct construction of a compositional predicate transformer semantics for a simple concurrent language with recursion. The correctness of the semantics is shown on the one hand with respect to a metric state transformer semantics using a topological duality, and on the other hand with respect to the weakest (liberal) precondition semantics that we defined in Chapter 3.

Several authors proposed a predicate transformer semantics for concurrent programs. For example, Van Lamsweerde and Sintzoff [LS79], Haase [Haa81], Flon and Suzuki [FS81], Elrad and Francez [EF84], Zwiers [Zwi87], Best [Bes83, Bes89], Lamport [Lam90], Scholefield and Zedan [SZ92], Van Breugel [Bre93], and Lukkien [Luk91, Luk94]. Taken all together, none of these references combines compositionality and recursion for an explicit parallel operator.

7.1 A simple concurrent language

In this section we define a simple concurrent language with recursion. We consider a small variation of the sequential finite non-deterministic language $\mathcal{L}_0$ introduced in Chapter 3 extended with a parallel operator.

As for $\mathcal{L}_0$, to define the language $\mathcal{L}_2$, we need as basic blocks the abstract sets $(v \in)\, \mathit{IVar}$ of (individual) variables , $(e \in)\, \mathit{Exp}$ of expressions, $(b \in)\, \mathit{BExp}$

of Boolean expressions, and $(x \in) PVar$ of procedure variables, respectively. For a fixed set of values *Val*, the set of (program) states $(s, t \in) \mathtt{St}$ is given by $\mathtt{St} = IVar \to Val$. Also, we postulate valuations

$$\mathcal{E}_{\mathcal{V}} : Exp \to (\mathtt{St} \to Val) \quad \text{and} \quad \mathcal{B}_{\mathcal{V}} : BExp \to \mathcal{P}(\mathtt{St}).$$

The language $\mathcal{L}_2$ below has assignments, conditionals '$b\!\to$', sequential composition ';', choice '$\Box$', parallel composition '$\|$', and recursion through procedure variables. The only new operator with respect to the language $\mathcal{L}_0$ of Chapter 3 is the parallel operator '$\|$'. Intuitively, the parallel composition of two statements executes in an interleaved way actions of both statements, while preserving the relative order of the actions in the statements.

7.1.1. Definition. (i) The set $(S \in) Stat_2$ of *statements* is given by

$$S \ ::= \ v := e \mid b\!\to \mid x \mid S \, ; S \mid S \,\Box\, S \mid S \,\|\, S \,.$$

(ii) The set $(G \in) GStat_2$ of *guarded statements* is given by

$$G \ ::= \ v := e \mid b\!\to \mid G \, ; S \mid G \,\Box\, G \mid G \,\|\, G \,.$$

(iii) The set $(d \in) Decl_2$ of *declarations* is given by $PVar \to GStat_2$.

(iv) The language $\mathcal{L}_2$ is given by $Decl_2 \times Stat_2$.

Assignments and conditionals are the only atomic statements. Their execution may not be interrupted by the other processes. Though resembling in their name, the guarded *statements* in the above definition and elsewhere in this chapter are completely different—both syntactically and as to their intended meaning—from Dijkstra's guarded *commands* [Dij76]. The declarations $d \in Decl_2$ associate a procedure body to each procedure variable x. For technical reasons (obtaining contractive higher-order transformations with semantic mappings as their unique fixed point), we restrict procedure bodies to guarded statements. Essentially, in a guarded statement G of $GStat_2$, every occurrence of a procedure variable is preceded either by an assignment or by a conditional statement.

7.2 Metric predicate transformers

In this section we introduce a domain Π_2 for a compositional backward semantics of the language $\mathcal{L}_2$ with parallel composition. This domain is obtained as the solution of a domain equation involving a functor which delivers metric predicate transformers. They are topological predicate transformers endowed with a distance which can be characterized in terms of saturated sets. This turns out to be convenient in formulating some properties of the domain Π_2. Because metric spaces taken with the ordinary metric topology are $\mathcal{T}_1$, every subset is saturated. Thus, in the light of our discussion in Chapter 5, every predicate is specifiable by a list of affirmative ones.

7.2.1. Definition. Let X be a set and Y be a metric space. A *metric predicate transformer* over Y and X is a function $\pi : \mathcal{P}(Y) \to \mathcal{P}(X)$ such that it is multiplicative (i.e. preserves arbitrary intersections) and, for all directed collections D of subsets which are open in the metric topology of Y,

$$\pi(\bigcup D) \quad = \quad \bigcup\{\pi(o) \mid o \in D\}.$$

We denote the set of all metric predicate transformers over Y and X by $MPT(Y, X)$.

Since a metric predicate transformer preserves arbitrary intersections, it is determined by its values on the metric open subsets: for a metric predicate transformer π and $P \subseteq Y$,

$$\pi(P) = \pi(\bigcap\{o \in \mathcal{O}(Y) \mid P \subseteq o\}) = \bigcap\{\pi(o) \mid P \subseteq o\}.$$

It follows that $MPT(Y, X)$ coincides with the set of all M-multiplicative and Scott continuous functions from $\mathcal{O}(Y)$ to $\mathcal{P}(X)$.

The set of all metric predicate transformers $MPT(Y, X)$ can be turned into a metric space as follows. For $\pi_1, \pi_2 \in MPT(Y, X)$ define their *backward distance* by

$$d_B(\pi_1, \pi_2) = \sup_{x \in X} d_{\mathcal{P}(\mathcal{P}(Y))}(\{P \mid x \in \pi_1(P)\}, \{P \mid x \in \pi_2(P)\}).$$

Next we want to prove that if Y is a complete metric space, so is $MPT(Y, X)$ for every set X. Before proving this, formally presented below as Theorem 7.2.3, we give an alternative definition of this metric. This alternative definition is based on Proposition 2.3.3 and on the following lemma which resembles Lemma 3.3.5 and Lemma 6.2.2.

7.2.2. Lemma. *Let $\pi : \mathcal{P}(Y) \to \mathcal{P}(X)$, where Y is a metric space. Then π is a metric predicate transformer in $MPT(Y, X)$ if and only if for every $x \in X$ there exists a unique compact subset $q(x, \pi)$ of Y such that*

$$x \in \pi(P) \quad \textit{if and only if} \quad q(x, \pi) \subseteq P, \tag{7.1}$$

for all $P \subseteq Y$.

Proof: Suppose π is in $MPT(Y, X)$ and define, for every $x \in X$, the set

$$q(x, \pi) = \bigcap\{P \subseteq Y \mid x \in \pi(P)\}.$$

Since π is multiplicative, $q(x, \pi)$ satisfies the stability condition (7.1), whereas Scott continuity with respect to metric opens implies that $q(x, \pi)$ is a metric compact subset of Y. Uniqueness of $q(x, \pi)$ can be proved as follows. For

$x \in X$, assume $A \subseteq Y$ such that $A \subseteq P$ if and only if $x \in \pi(P)$ for all $P \in \mathcal{P}(Y)$. If $a \in A \setminus q(x, \pi)$ then we have the following contradiction:

$$q(x, \pi) \subseteq Y \setminus \{a\} \;\Leftrightarrow\; a \in \pi(Y \setminus \{a\}) \;\Leftrightarrow\; A \subseteq Y \setminus \{a\}.$$

Therefore $q(x, \pi) \subseteq A$. We conclude, on symmetric considerations, $q(x, \pi) = A$.

Conversely, suppose $\pi : \mathcal{P}(Y) \to \mathcal{P}(X)$ satisfies the proposed criterion. To see that it is multiplicative we have, for an arbitrary set I and $P_i \subseteq Y$ for all $i \in I$,

$$\begin{aligned} x \in \pi(\bigcap_I P_i) \quad &\Leftrightarrow \quad q(x, \pi) \subseteq \bigcap_I P_i \\ &\Leftrightarrow \quad \forall i \in I \colon q(x, \pi) \subseteq P_i \\ &\Leftrightarrow \quad x \in \bigcap_I \pi(P_i). \end{aligned}$$

From compactness of $q(x, \pi)$ it follows that π preserves directed unions of metric opens. Therefore $\pi \in MPT(Y, X)$. $\square$

As consequence of the above lemma we have that $x \in \pi(q(x, \pi))$ and hence $q(x, \pi) \in \{P \mid x \in \pi(P)\}$. Thus, by Theorem 2.3.3, for every $x \in X$ and $\pi_1, \pi_2 \in MPT(Y, X)$,

$$d_{\mathcal{P}(\mathcal{P}(Y))}(\{P \mid x \in \pi_1(P)\}, \{P \mid x \in \pi_2(P)\}) = d_{\mathcal{P}(Y)}(q(x, \pi_1), q(x, \pi_2)).$$

Therefore we can conclude for every $\pi_1, \pi_2 \in MPT(Y, X)$,

$$d_B(\pi_1, \pi_2) = \sup_{x \in X} d_{\mathcal{P}(Y)}(q(x, \pi_1), q(x, \pi_2)).$$

We use this alternative definition of the metric on $MPT(Y, X)$ to prove the completeness of the space.

7.2.3. Theorem. *If Y is a complete metric space then so is $MPT(Y, X)$ for every set X.*

Proof: In verifying that d_B is a metric, we only check that $d_B(\pi_1, \pi_2) = 0$ implies $\pi_1 = \pi_2$. The other conditions follow from the respective properties of the Hausdorff distance on the compact subsets of Y. Suppose $d_B(\pi_1, \pi_2) = 0$. Then $q(x, \pi_1) = q(x, \pi_2)$ for all $x \in X$. By Lemma 7.2.2, for all $P \subseteq Y$,

$$x \in \pi_1(P) \;\Leftrightarrow\; q(x, \pi_1) \subseteq P \;\Leftrightarrow\; q(x, \pi_2) \subseteq P \;\Leftrightarrow\; x \in \pi_2(P).$$

So, $\pi_1(P) = \pi_2(P)$ for all P and hence $\pi_1 = \pi_2$.

Next we check completeness of $MPT(Y, X)$. Suppose $(\pi_i)_i$ is a Cauchy sequence in $MPT(Y, X)$. Then, for each $x \in X$ the sequence $(q(x, \pi_i))_i$ is Cauchy with respect to the Hausdorff distance in $\mathcal{P}_{co}(Y)$. Since the latter is a complete metric space, $\lim_i q(x, \pi_i)$ exists for every $x \in X$ and it is a compact subset of Y. Define the function $\pi : \mathcal{P}(Y) \to \mathcal{P}(X)$ by

$$\pi(P) = \{x \in X \mid \lim_i q(x, \pi_i) \subseteq P\},$$

for every $P \subseteq Y$. By definition, the compact subset $\lim_i q(x, \pi_i)$ of Y satisfies condition (7.1). Hence, by Lemma 7.2.2, $\pi \in MPT(Y, X)$. Notice that this implies that $q(x, \pi) = q(x, \lim_i \pi_i) = \lim_i q(x, \pi_i)$ for all $x \in X$. From

$$d_B(\pi, \pi_i) = \sup_{x \in X} d_{\mathcal{P}(Y)}(q(x, \pi), q(x, \pi_i))$$

it follows that $\pi = \lim_i \pi_i$ in $MPT(Y, X)$. $\square$

For a fixed set X, the assignment $Y \mapsto MPT(Y, X)$ between complete metric spaces can be extended to a functor on the category **CMS** of complete metric spaces with non-expansive maps. For a non-expansive function $f : Y_1 \to Y_2$ define $MPT(f, X) = \lambda\pi.\pi \circ f^{-1}$, that is,

$$MPT(f, X)(\pi)(P) = \pi(f^{-1}(P))$$

for all $\pi \in MPT(Y_1, X)$ and $P \subseteq Y_2$. For proving that the functor $MPT(-, X)$: **CMS** $\to$ **CMS** is well-defined we need the following proposition.

7.2.4. Proposition. *Let $f : Y_1 \to Y_2$ be a non-expansive map between complete metric spaces, and let $\pi \in MPT(Y_1, X)$ for a fixed set X. Then $\pi \circ f^{-1}$ is a metric predicate transformer in $MPT(Y_2, X)$ such that, for any $x \in X$,*

$$q(x, \pi \circ f^{-1}) \;\; = \;\; f(q(x, \pi)).$$

Proof: Multiplicativity of $\pi \circ f^{-1} : \mathcal{P}(Y) \to \mathcal{P}(X)$ follows from set-theoretic laws for f^{-1}, while Scott continuity with respect to the metric opens follows from the metric continuity of f (the inverse image of an open set is open). Furthermore, for $x \in X$ and $P \subseteq Y_2$,

$$\begin{aligned} f(q(x, \pi)) \subseteq P \quad &\Leftrightarrow \quad q(x, \pi) \subseteq f^{-1}(P) \\ &\Leftrightarrow \quad x \in \pi(f^{-1}(P)) \quad \text{[Lemma 7.2.2 for } \pi\text{]} \end{aligned}$$

where the first equivalence is obtained from a standard set-theoretic argument. Using Lemma 7.2.2 for $\pi \circ f^{-1}$ we obtain $q(x, \pi \circ f^{-1}) = f(q(x, \pi))$. $\square$

7.2.5. Lemma. *The functor $MPT(-, X)$*: **CMS** $\to$ **CMS** *for some fixed set X is well-defined and locally non-expansive.*

Proof: To prove well-definedness of the functor $MPT(-, X)$ we first check whether $MPT(f, X)$ is non-expansive for non-expansive $f : Y_1 \to Y_2$. Let $\pi_1, \pi_2 \in MPT(M, X)$. Then

$$\begin{aligned}
& d_B(MPT(f,X)(\pi_1), MPT(f,X)(\pi_2)) \\
& \quad = \ \sup_{x\in X} d_{\mathcal{P}(Y)}(q(x,\pi_1\circ f^{-1}), q(x,\pi_2\circ f^{-1})) \\
& \quad = \ \sup_{x\in X} d_{\mathcal{P}(Y)}(f(q(x,\pi_1)), f(q(x,\pi_2))) \quad \text{[Proposition 7.2.4]} \\
& \quad \le \ \sup_{x\in X} d_{\mathcal{P}(Y)}(q(x,\pi_1), q(x,\pi_2)) \quad [f \text{ non-expansive}] \\
& \quad = \ d_B(\pi_1,\pi_2).
\end{aligned}$$

For proving well-definedness of $MPT(-,X)$ it remains to establish its functoriality. Preservation of identities is immediate, whereas $MPT(g\circ f, X) = MPT(g,X)\circ MPT(f,X)$, for $f\colon Y_1\to Y_2$ and $g\colon Y_2\to Y_3$ in **CMS**, is directly verified for all arguments $\pi\in MPT(Y_1,X)$ using the equality $(g\circ f)^{-1} = f^{-1}\circ g^{-1}$.

Finally, we check that the functor $MPT(-,X) : \mathbf{CMS}\to\mathbf{CMS}$ is locally non-expansive. Take $f,g\in\mathbf{CMS}(Y_1,Y_2)$. We have to check

$$d(MPT(f,X), MPT(g,X)) \le d(f,g) \tag{7.2}$$

with the distance on the left-hand side taken in $MPT(Y_1,X)\to MPT(Y_2,X)$, and on the right-hand side taken in $Y_1\to Y_2$. So, pick $\pi\in MPT(Y_1,X)$. Then

$$\begin{aligned}
& d_B(MPT(f,X)(\pi), MPT(g,X)(\pi)) \\
& \quad = \ \sup\{d_{\mathcal{P}(Y_1)}(q(x, MPT(f,X)(\pi)), q(x, MPT(g,X)(\pi))) \mid x\in X\} \\
& \quad = \ \sup\{d_{\mathcal{P}(Y_1)}(f(q(x,\pi)), g(q(x,\pi))) \mid x\in X\} \\
& \quad \le \ d(f,g),
\end{aligned}$$

since, for any $x\in X$, $d(f(q(x,\pi)), g(q(x,\pi)) \le d(f,g)$. Thus

$$d_B(MPT(f,X)(\pi), MPT(g,X)(\pi)) \ \le \ d(f,g).$$

Now (7.2) follows directly by definition of the distance on $MPT(Y_1,X)\to MPT(Y_2,X)$. $\square$

We are now ready to define a recursive domain of metric predicate transformers Π_2 using the functor $MPT(-,X)$. In the next section we will use this domain to give a compositional semantics to the language $\mathcal{L}_2$.

7.2.6. Definition. Let $\mathtt{St}$ be the set of (program) states. The complete metric space $(\pi\in)\,\Pi_2$ of *metric predicate transformers with resumptions* is defined as the unique (up to isometry) fixed point of the contractive functor in **CMS**

$$MPT(\mathtt{St} + \mathtt{St}\times \tfrac{1}{2}\cdot -, \mathtt{St}).$$

Therefore Π_2 is the unique (up to isometry) complete metric space satisfying

$$\Pi_2 \ \cong \ MPT(\mathtt{St} + \mathtt{St}\times\tfrac{1}{2}\cdot\Pi_2, \mathtt{St}).$$

Below we will omit the isomorphism pair relating the right and the left hand side of the above domain equation.

The intuition behind an element $\pi \in \Pi_2$ is as follows. Given a set $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$ it yields the set of all states $s \in \mathtt{St}$ such that when execution of the program represented by π starts in one of these states, it either terminates after an atomic step in a state $t \in \mathtt{St}$ satisfying P, or it makes a first (atomic) step to a state s and immediately gives the control to another program, represented by the predicate transformer ρ, with $\langle s, \rho \rangle \in P$. We have not yet made a formal semantic mapping from syntax to the domain Π_2, but it might help the reader to consider the following example. Consider the statement $v := 1\ ;\ v := 2$. The corresponding metric predicate transformer π is

$$\pi(P) \quad = \quad \{s \mid \langle s[1/v], \rho \rangle \in P\}$$

with ρ (the semantics of $v := 2$) such that $\rho(Q) = \{s \mid s[2/v] \in Q\}$. The semantics is backwards: to give a semantic meaning to $v{:=}1 ; v{:=}2$ we first need to give a semantic meaning to $v := 2$ and then combine it with the semantical meaning of $v := 1$.

7.3 Metric predicate transformer semantics

The language $\mathcal{L}_2$ can be considered to be an extension of the language $\mathcal{L}_0$ presented in Chapter 3. Hence we will base our present semantics on the weakest precondition semantics $Wp_0[\![\cdot]\!]$ of Chapter 3. However, the presence of the parallel operator '$\|$' in $\mathcal{L}_2$, invokes, when insisting on a compositional treatment, a more involved domain than $PT_T(\mathtt{St}, \mathtt{St})$ used for the semantics of the language $\mathcal{L}_0$. We will employ the branching domain Π_2 given in Definition 7.2.6 above. As maybe expected, atomic statements are treated in the same way as for the language $\mathcal{L}_0$. Hence we introduce the predicate transformers for the semantics of the atomic statements: assignments and conditionals.

7.3.1. Definition. For every function $f{:}\mathtt{St} \to \mathit{Val}$, individual variable $v \in \mathit{IVar}$ and subset V of $\mathtt{St}$ define the predicate transformers '$[f/v]$' and '$V{\rightarrow}$' in Π_2 by

$$\begin{aligned} [f/v](P) \quad &= \quad \{s \in \mathtt{St} \mid s[f(s)/v] \in P\}, \\ V{\rightarrow}(P) \quad &= \quad \{s \in \mathtt{St} \mid s \in V \;\Rightarrow\; s \in P\}, \end{aligned}$$

for all $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$.

Both the functions above are well-defined predicate transformers in Π_2. Notice that if $P \cap \mathtt{St} = \emptyset$ then $[f/v](P) = \emptyset$ and also $V{\rightarrow}(P) = \emptyset$. In general for $\pi \in \Pi_2$, $\pi(\mathtt{St} \times \Pi_2) = \emptyset$ if and only if π corresponds to an atomic program (not necessarily in $\mathcal{L}_2$, as it can be a specification construct like, for example, an infinite multiple assignment). Every assignment $v := e$ in $Stat_2$ induces the predicate transformer $[\mathcal{E}_{\mathcal{V}}(e)/v]$ in Π_2, and every conditional $b{\rightarrow}$ in $Stat_2$ induces the predicate transformer $\mathcal{B}_{\mathcal{V}}(b){\rightarrow} \in \Pi_2$.

In the domain Π_2 interleaving points are explicitly represented. This complicates the definitions of the operators on predicate transformers in Π_2 which reflect the constructions available in $\mathcal{L}_2$. Since infinite behaviour is allowed by $\mathcal{L}_2$, some operators below have a recursive definition and therefore well-definedness must be proved.

7.3.2. Definition. The operators '$;$', '$\Box$' and '$\|$' on Π_2 are given, for $\pi_1, \pi_2 \in \Pi_2$ and $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$, by

$$\begin{array}{rcl}(\pi_1 \mathbin{;} \pi_2)(P) & = & \pi_1(\{s \mid \langle s, \pi_2\rangle \in P\} \cup \{\langle s, \rho\rangle \mid \langle s, \rho \mathbin{;} \pi_2\rangle \in P\}),\\ (\pi_1 \Box \pi_2)(P) & = & \pi_1(P) \cap \pi_2(P),\\ (\pi_1 \parallel \pi_2)(P) & = & \pi_1(\{s \mid \langle s, \pi_2\rangle \in P\} \cup \{\langle s, \rho\rangle \mid \langle s, \rho \parallel \pi_2\rangle \in P\}) \cap \\ & & \pi_2(\{s \mid \langle s, \pi_1\rangle \in P\} \cup \{\langle s, \rho\rangle \mid \langle s, \pi_1 \parallel \rho\rangle \in P\}).\end{array}$$

The intuition behind the above operators will be given after Lemma 7.3.12 by means of some examples about the weakest precondition semantics for $\mathcal{L}_2$. Let us for the moment concentrate on the well-definedness of the above operators. For the '$\Box$' operator it is immediate.

7.3.3. Lemma. *The mapping '$\Box$' is well-defined and non-expansive.*

Proof: Choose $\pi_1, \pi_2 \in \Pi_2$ arbitrarily. Multiplicativity of $\pi_1 \Box \pi_2$ is straightforward. Suppose $\mathcal{V}$ is a directed set of metric opens of $\mathtt{St} + \mathtt{St} \times \frac{1}{2} \cdot \Pi_2$. Then we have

$$\begin{array}{l}(\pi_1 \Box \pi_2)(\bigcup \mathcal{V})\\ \quad = \bigcup\{\pi_1(P) \mid P \in \mathcal{V}\} \cap \bigcup\{\pi_2(P) \mid P \in \mathcal{V}\} \quad [\text{continuity of } \pi_1 \text{ and } \pi_2]\\ \quad = \bigcup\{\pi_1(P) \cap \pi_2(P) \mid P \in \mathcal{V}\} \quad [\mathcal{V} \text{ is directed and } \pi_1, \pi_2 \text{ are monotone}]\\ \quad = \bigcup\{(\pi_1 \Box \pi_2)(P) \mid P \in \mathcal{V}\}.\end{array}$$

Hence, $\pi_1 \Box \pi_2$ is Scott continuous on metric opens, and $\pi_1 \Box \pi_2 \in \Pi_2$. Non-expansiveness of $\Box$ is readily checked. $\Box$

Well-definedness of the operator '$;$' and '$\|$' is less trivial and is supported by higher order transformations $\Omega_{;}$ and $\Omega_{\|}$, respectively.

7.3.4. Definition. Let $(\phi \in)Opr$ be the set of all non-expansive functions in $\Pi_2 \times \Pi_2 \to \Pi_2$. The higher-order transformation $\Omega_{;} : Opr \to Opr$ is given by

$$\Omega_{;}(\phi)(\langle \pi_1, \pi_2\rangle)(P) = \pi_1(\{s \mid \langle s, \pi_2\rangle \in P\} \cup \{\langle s, \rho\rangle \mid \langle s, \phi(\langle \rho, \pi_2\rangle)\rangle \in P\}),$$

where $\pi_1, \pi_2 \in \Pi_2$ and $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$.

We will prove that the operator '$;$' as defined in Definition 7.3.2 is the unique fixed point of $\Omega_{;}$. We need three technical lemma's.

7.3.5. Lemma. *Let* $\pi \in \Pi_2$. *Put*

$$P' = \{ s \in \mathtt{St} \mid \langle s, \pi \rangle \in P \}$$

for $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$. *Then it holds that*

(i) *If* P *is open then also* P' *is open.*
(ii) *If* P *is closed then also* P' *is closed.*
(iii) *For* $\mathcal{X} \subseteq \mathcal{P}(\mathtt{St} + \mathtt{St} \times \Pi_2)$, $\bigcap \{ P' \mid P \in \mathcal{X} \} = (\bigcap \mathcal{X})'$.

Proof: We only check part (i); part (ii) is similar and part (iii) is straightforward. If $s \in P'$, then $\langle s, \pi \rangle \in P$. Hence, for some suitable $\epsilon < 1$, $B_\epsilon(\langle s, \pi \rangle) \subseteq P$ and therefore $B_\epsilon(s) \subseteq P'$, since $B_\epsilon(s) \subseteq \mathtt{St}$ by the assumption $\epsilon < 1$. □

An immediate consequence of Lemma 7.3.5.(iii) is that if $P_1 \subseteq P_2$ then $P_1' \subseteq P_2'$ for all subsets P_1 and P_2 of $\mathtt{St} + \mathtt{St} \times \Pi_2$.

7.3.6. Lemma. *Let* $\phi : \Pi_2 \times \Pi_2 \to \Pi_2$ *be non-expansive and* $\pi \in \Pi_2$. *Put*

$$P'' \quad = \quad \{ \langle t, \rho \rangle \in \mathtt{St} \times \Pi_2 \mid \langle t, \phi(\langle \rho, \pi \rangle) \rangle \in P \}$$

for $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$. *Then it holds that*

(i) *If* P *is open then also* P'' *is open.*
(ii) *If* P *is closed then also* P'' *is closed.*
(iii) *For* $\mathcal{X} \subseteq \mathcal{P}(\mathtt{St} + \mathtt{St} \times \Pi_2)$, $\bigcap \{ P'' \mid P \in \mathcal{X} \} = (\bigcap \mathcal{X})''$.

Proof: In verifying part (i) and (ii) we observe that P'' is the inverse image under the continuous function $\langle id_{\mathtt{St}}, \lambda \rho . \phi(\langle \rho, \pi \rangle) \rangle : (\mathtt{St} \times \Pi_2) \to (\mathtt{St} \times \Pi_2)$ of $P \cap (\mathtt{St} \times \Pi_2)$. Hence P'' is open if P is open, and closed if P is closed. Part (iii) is readily checked. □

As before, from Lemma 7.3.6.(iii) it follows that if $P_1 \subseteq P_2$ then $P_1'' \subseteq P_2''$ for all subsets P_1 and P_2 of $\mathtt{St} + \mathtt{St} \times \Pi_2$.

7.3.7. Lemma. (i) *For any non-expansive* $\phi : \Pi_2 \times \Pi_2 \to \Pi_2$, *and* $\pi_1, \pi_2 \in \Pi_2$ *it holds that* $\Omega_;(\phi)(\langle \pi_1, \pi_2 \rangle) \in \Pi_2$.

(ii) *For any non-expansive* $\phi : \Pi_2 \times \Pi_2 \to \Pi_2$ *the function* $\Omega_;(\phi)$ *is also non-expansive. Moreover, for any* $\pi_1, \pi_2, \rho_1, \rho_2 \in \Pi_2$:

$$d_{\Pi_2 \times \Pi_2}(\Omega_;(\phi)(\pi_1, \pi_2), \Omega_;(\phi)(\rho_1, \rho_2)) \quad \leq \quad \max\{ d_{\Pi_2}(\pi_1, \rho_1), \tfrac{1}{2} d_{\Pi_2}(\pi_2, \rho_2) \}.$$

(iii) *The transformation* $\Omega_;$ *is a* $\frac{1}{2}$*-contraction.*

Proof: (i) Let ϕ be a non-expansive map in $\Pi_2 \times \Pi_2 \to \Pi_2$, $\pi_1, \pi_2 \in \Pi_2$, and put $\pi = \Omega_;(\phi)(\langle \pi_1, \pi_2 \rangle)$. Let P', P'' be as in Lemmas 7.3.5 and 7.3.6 with respect to ϕ, π_2 and any $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$. We first verify $\Omega_;(\phi)(\langle \pi_1, \pi_2 \rangle) \in \Pi_2$.

To prove that π is multiplicative, take $\mathcal{X} \subseteq \mathcal{P}(\mathtt{St} + \mathtt{St} \times \Pi_2)$. By Lemmas 7.3.5 and 7.3.6,

$$\bigcap \{ P' \mid P \in \mathcal{X} \} = (\bigcap \mathcal{X})' \quad \text{and} \quad \bigcap \{ P'' \mid P \in \mathcal{X} \} = (\bigcap \mathcal{X})''.$$

Hence, by disjointness of the P' and P'',

$$\bigcap\{P' \cup P'' \mid P \in \mathcal{X}\} \;=\; (\bigcap \mathcal{X})' \cup (\bigcap \mathcal{X})''.$$

Multiplicativity of π_1 now delivers

$$\bigcap\{\pi_1(P' \cup P'') \mid P \in \mathcal{X}\} \;=\; \pi_1((\bigcap \mathcal{X})' \cup (\bigcap \mathcal{X})'').$$

Since $\pi_1(P' \cup P'') = \pi(P)$ by definition, we have that $\Omega_;(\phi)(\pi_1, \pi_2)$ is multiplicative.

Next we prove that π preserves directed unions of metric opens. Suppose $\mathcal{V}$ is a directed set of metric open subsets of $\mathtt{St} + \mathtt{St} \times \frac{1}{2} \cdot \Pi_2$. We then have

$$\begin{aligned}
\pi(\bigcup \mathcal{V}) &= \pi_1(\bigcup\{P' \mid P \in \mathcal{V}\} \cup \bigcup\{P'' \mid P \in \mathcal{V}\}) \\
&= \pi_1(\bigcup\{P' \cup P'' \mid P \in \mathcal{V}\}) \\
&= \bigcup\{\pi_1(P' \cup P'') \mid P \in \mathcal{V}\} \quad [\text{Lemmas 7.3.5, 7.3.6, and } \pi_1 \in \Pi_2] \\
&= \bigcup\{\pi(P) \mid P \in \mathcal{V}\}
\end{aligned}$$

Hence π preserves directed joins of opens and thus $\pi \in \Pi_2$.

(ii) Choose $\phi, \pi_1, \pi_2, \rho_1, \rho_2$ arbitrarily in Π_2 and put $\pi = \Omega_;(\phi)(\pi_1, \pi_2)$, and $\rho = \Omega_;(\phi)(\rho_1, \rho_2)$. We first establish

$$\begin{aligned}
q(s, \pi) \;=\; & \{\langle t, \pi_2\rangle \mid t \in q(s, \pi_1)\} \cup \\
& \{\langle t, \phi(\langle \pi', \pi_2\rangle)\rangle \mid \langle t, \pi'\rangle \in q(s, \pi_1)\}
\end{aligned} \tag{7.3}$$

for every $s \in S$, using Lemma 7.2.2. Let $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$. Then

$$\begin{aligned}
&\{\langle t, \pi_2\rangle \mid t \in q(s, \pi_1)\} \cup \{\langle t, \phi(\langle \pi', \pi_2\rangle \mid \langle t, \pi'\rangle \in q(s, \pi_1)\} \subseteq P \\
\Leftrightarrow\quad & q(s, \pi_1) \subseteq \{t \mid \langle t, \pi_2\rangle \in P\} \cup \{\langle t, \pi'\rangle \mid \langle t, \phi(\langle \pi', \pi_2\rangle)\rangle \in P\} \\
\Leftrightarrow\quad & q(s, \pi_1) \subseteq P' \cup P'' \quad [\text{Definitions of } P' \text{ and } P''] \\
\Leftrightarrow\quad & s \in \pi_1(P' \cup P'') \quad [\text{Lemmas 7.2.2, 7.3.5, and 7.3.6}] \\
\Leftrightarrow\quad & s \in \pi(P). \quad [\text{Definition of } \pi]
\end{aligned}$$

A similar result holds for $q(s, \rho)$. Now, for every $s \in \mathtt{St}$ we have (omitting the subscripts on the distance function d)

$$\begin{aligned}
& d(q(s, \pi), q(s, \rho)) \\
& \leq \max\{\; d(\{\langle t, \pi_2\rangle \mid t \in q(s, \pi_1)\}, \{\langle t, \rho_2\rangle \mid t \in q(s, \rho_1)\}), \\
& \qquad\qquad d(\{\langle t, \phi(\langle \pi', \pi_2\rangle)\rangle \mid \langle t, \pi'\rangle \in q(s, \pi_1)\}, \\
& \qquad\qquad\quad \{\langle t, \phi(\langle \rho', \rho_2\rangle)\rangle \mid \langle t, \rho'\rangle \in q(s, \rho_1)\}) \;\} \\
& \leq \max\{\; d(q(s, \pi_1), q(s, \rho_1)), \tfrac{1}{2} d_{\Pi_2}(\pi_2, \rho_2) \;\} \\
& \leq \max\{\; d_{\Pi_2}(\pi_1, \rho_1), \tfrac{1}{2} d_{\Pi_2}(\pi_2, \rho_2) \;\}.
\end{aligned}$$

Taking the supremum over $\mathtt{St}$, the result follows.

(iii) Let $\phi_1, \phi_2 \in \Pi_2 \times \Pi_2 \to \Pi_2$ be non-expansive. Then, for arbitrary $\pi_1, \pi_2 \in \Pi_2$ and $s \in \mathtt{St}$, we have

$$\begin{aligned}
& d(q(s, \Omega_;(\phi_1)(\pi_1, \pi_2)), q(s, \Omega_;(\phi_2)(\pi_1, \pi_2))) \\
& \leq\; d(\{\langle t, \phi_1(\langle \rho, \pi_2\rangle)\rangle \mid \langle t, \rho\rangle \in q(s, \pi_1)\}, \\
& \qquad\quad \{\langle t, \phi_2(\langle \rho, \pi_2\rangle)\rangle \mid \langle t, \rho\rangle \in q(s, \pi_1)\}) \quad [\text{Equation (7.3)}] \\
& \leq\; \tfrac{1}{2} d(\phi_1, \phi_2).
\end{aligned}$$

Hence $d(\Omega_;(\phi_1), \Omega_;(\phi_2)) \le \frac{1}{2} d(\phi_1, \phi_2)$ and $\Omega_;$ is $\frac{1}{2}$-contractive. $\square$

Now we are in a position to prove well-definedness of the operator ';': it is the unique fixed point of $\Omega_;$.

7.3.8. Theorem. *The operator ';' as defined in Definition 7.3.2 is the unique fixed point of* $\Omega_;$.

Proof: Since $\Omega_;$ is a $\frac{1}{2}$-contraction it has a unique fixed point by Banach's fixed point theorem. It is easy to verify that ';' is the fixed point of $\Omega_;$. $\square$

Next we proceed with the justification of the recursive definition of the operator '$\|$' in Π_2 given in Definition 7.3.2. We use again a higher-order transformation.

7.3.9. Definition. Let $(\phi \in)$ Opr be the set of all non-expansive maps in $\Pi_2 \times \Pi_2 \to \Pi_2$. The higher-order transformation $\Omega_\| : Opr \to Opr$ is given by

$$\Omega_\|(\phi)(\langle \pi_1, \pi_2 \rangle) = \Omega_;(\phi)(\langle \pi_1, \pi_2 \rangle) \,\square\, \Omega_;(\phi)(\langle \pi_2, \pi_1 \rangle).$$

Next we see that the higher-order transformation $\Omega_\|$ is a contraction and has, assured by Banach's theorem, a unique fixed point, being—by definition—'$\|$'.

7.3.10. Theorem. *For every $\phi \in Opr$, $\Omega_\|(\phi)$ is non-expansive, whereas the function $\Omega_\|$ is an $\frac{1}{2}$-contraction. Further, the operator '$\|$' as defined in Definition 7.3.2 is the unique fixed point of $\Omega_\|$.*

Proof: The proof of the first part of the theorem is straightforward from the definition of $\Omega_\|$, and Lemmas 7.3.3 and 7.3.7. Since $\Omega_\|$ is an $\frac{1}{2}$-contraction it has a unique fixed point by Banach's fixed point theorem. It is easy to verify that '$\|$' is indeed the fixed point of $\Omega_\|$. $\square$

We are now ready to present the semantics Wp_2 for $\mathcal{L}_2$.

7.3.11. Definition. The weakest precondition semantics Wp_2 is the unique function in $\mathcal{L}_2 \to \Pi_2$ satisfying

$$\begin{array}{lcl}
Wp_2[\![\langle d, v := e \rangle]\!] & = & [\mathcal{E}_\mathcal{V}(e)/v], \\
Wp_2[\![\langle d, b{\rightarrow} \rangle]\!] & = & \mathcal{B}_\mathcal{V}(b){\rightarrow}, \\
Wp_2[\![\langle d, x \rangle]\!] & = & Wp_2[\![\langle d, d(x) \rangle]\!], \\
Wp_2[\![\langle d, (S_1 \,;\, S_2) \rangle]\!] & = & Wp_2[\![\langle d, S_1 \rangle]\!] \,;\, Wp_2[\![\langle d, S_2 \rangle]\!], \\
Wp_2[\![\langle d, S_1 \,\square\, S_2 \rangle]\!] & = & Wp_2[\![\langle d, S_1 \rangle]\!] \,\square\, Wp_2[\![\langle d, S_2 \rangle]\!], \\
Wp_2[\![\langle d, S_1 \,\|\, S_2 \rangle]\!] & = & Wp_2[\![\langle d, S_1 \rangle]\!] \,\|\, Wp_2[\![\langle d, S_2 \rangle]\!].
\end{array}$$

Justification of the proposed definition can be obtained by an application of the higher order transformation technique in a metric setting, as proposed originally in [KR90]. We need first a map wgt_2 which assigns a natural number to every program in $\mathcal{L}_2$ to be used in proofs based on induction. It is defined inductively as follows:

$$\begin{array}{lcl}
wgt_2(\langle d, v := e\rangle) & = & 1, \\
wgt_2(\langle d, b\rightarrow\rangle) & = & 1, \\
wgt_2(\langle d, x\rangle) & = & wgt_2(\langle d, d(x)\rangle) + 1, \\
wgt_2(\langle d, S_1 \,;\, S_2\rangle) & = & wgt_2(\langle d, S_1\rangle) + 1, \\
wgt_2(\langle d, S_1 \,\Box\, S_2\rangle) & = & \max\{wgt_2(\langle d, S_1\rangle), wgt_2(\langle d, S_2\rangle)\} + 1, \\
wgt_2(\langle d, S_1 \parallel S_2\rangle) & = & \max\{wgt_2(\langle d, S_1\rangle), wgt_2(\langle d, S_2\rangle)\} + 1.
\end{array}$$

The *weight function* wgt_2 is well-defined for each pair $\langle d, S\rangle \in \mathcal{L}_1$ as can be easily seen by induction on the syntactic complexity: first on the complexity of guarded statements and then on the complexity of general statements (more information on the weight functions can be found in [Bre94] and [BV96]). We are now ready for the justification of the semantic function Wp_2. It is based on a mapping $\Psi_2 : Sem_2 \to Sem_2$ where $(F \in)Sem_2 = \mathcal{L}_2 \to \Pi_2$. Pivotal is the clause

$$\Psi_2(F)(\langle d, S_1 \,;\, S_2\rangle) \quad = \quad \Psi_2(F)(\langle d, S_1\rangle) \,;\, F(\langle d, S_2\rangle)$$

for the sequential composition. Note that F and not $\Psi_2(F)$ is applied to the second component S_2. The unique fixed point of this continuous endomorphism will satisfy the conditions of Definition 7.3.11.

7.3.12. Lemma. *Let* $F \in Sem_2 = \mathcal{L}_2 \to \Pi_2$. *Define* $\Psi_2 : Sem_2 \to Sem_2$ *inductively by*

$$\begin{array}{lcl}
\Psi_2(F)(\langle d, v := e\rangle) & = & [\mathcal{E}\nu(e)/v], \\
\Psi_2(F)(\langle d, b\rightarrow\rangle) & = & \mathcal{B}\nu(b)\rightarrow, \\
\Psi_2(F)(\langle d, x\rangle) & = & \Psi_2(F)(\langle d, d(x)\rangle), \\
\Psi_2(F)(\langle d, (S_1 \,;\, S_2)\rangle) & = & \Psi_2(F)(\langle d, S_1\rangle) \,;\, F(\langle d, S_2\rangle), \\
\Psi_2(F)(\langle d, S_1 \,\Box\, S_2\rangle) & = & \Psi_2(F)(\langle d, S_1\rangle) \,\Box\, \Psi_2(F)(\langle d, S_2\rangle), \\
\Psi_2(F)(\langle d, S_1 \parallel S_2\rangle) & = & \Psi_2(F)(\langle d, S_1\rangle) \parallel \Psi_2(F)(\langle d, S_2\rangle).
\end{array}$$

Then Ψ_2 *is* $\frac{1}{2}$*-contractive and* $Wp_2[\![\cdot]\!]$ *defined in Definition 7.3.11 is the unique fixed point of* Ψ_2.

Proof: We show, by induction on $wgt_2(\langle d, S\rangle)$, that

$$d_{\mathrm{Sem}_2}(\Psi_2(F_1)(\langle d, S\rangle), \Psi_2(F_2)(\langle d, S\rangle)) \quad \leq \quad \tfrac{1}{2} \cdot d_{\mathrm{Sem}_2}(F_1, F_2)$$

for any $F_1, F_2 \in Sem_2$. We expand two typical sub-cases (omitting the subscripts on the distance functions).

$[x]$
$$\begin{aligned} & d(\Psi_2(F_1)(\langle d, x\rangle), \Psi_2(F_2)(\langle d, x\rangle) \\ = \;& d(\Psi_2(F_1)(\langle d, d(x)\rangle), \Psi_2(F_2)(\langle d, d(x)\rangle) \\ \leq \;& \tfrac{1}{2} d(F_1, F_2). \quad \text{[induction hypothesis]} \end{aligned}$$

$[S_1 \,;\, S_2]$
$$\begin{aligned} & d(\Psi_2(F_1)(\langle d, S_1 \,;\, S_2\rangle), \Psi_2(F_2)(\langle d, S_1 \,;\, S_2\rangle) \\ = \;& d(\Psi_2(F_1)(\langle d, S_1\rangle) \,;\, F_1(\langle d, S_2\rangle), \Psi_2(F_2)(\langle d, S_1\rangle) \,;\, F_2(\langle d, S_2\rangle)) \\ \leq \;& \max\{ d(\Psi_2(F_1)(\langle d, S_1\rangle), \Psi_2(F_2)(\langle d, S_1\rangle)), \\ & \qquad \tfrac{1}{2} d(F_1(\langle d, S_2\rangle), F_2(\langle d, S_2\rangle))\} \quad \text{[Lemma 7.3.7.(ii)]} \\ \leq \;& \tfrac{1}{2} d(F_1, F_2). \quad \text{[induction hypothesis, definition } d(F_1, F_2)] \qquad \square \end{aligned}$$

The language $\mathcal{L}_2$ does not have the assert command $\{b\}$ as primitive: it is undefined if b fails whereas it acts as skip otherwise. A subsequent extension of the models is necessary. However, an additional clause involving the operator

$$Wp_2[\![\{b\}]\!](P) \quad = \quad \{ s \mid s \in \mathcal{B}_{\mathcal{V}}(b) \;\&\; s \in P \}$$

is sufficient. The underlying domain then needs to be adapted as well. Preservation of nonempty intersections will replace the multiplicativity condition. Locality can also be dealt with, using techniques developed in the metric setting (see [BKPR93] and also [BV96]). By adding a simple atomization operator on sequential and non-deterministic statements, synchronization via semaphores can be easily obtained in the setting of Π_2. It is an open problem, however, if it is possible to deal with *angelic* non-determinacy.

Let us now consider some examples, one of which involving the parallel operator, to illustrate the semantics Wp_2 and the definition of the operators given in Definition 7.3.2. The sequential statement $v_1 := 1 \,;\, v_2 := 2$ will act as the first example. Maybe surprisingly we will obtain for the predicate

$$P \quad = \quad \{ s \in \mathtt{St} \mid s(v_1) = 1 \;\&\; s(v_2) = 2 \}$$

that $Wp_2[\![\langle d, v_1 := 1 \,;\, v_2 := 2\rangle]\!](P) = \emptyset$ (for some fixed, but arbitrary declaration d). Let us write π_1, π_2 for $Wp_2[\![\langle d, v_1 := 1\rangle]\!]$, $Wp_2[\![\langle d, v_2 := 2\rangle]\!]$, respectively. We then have

$$\begin{aligned} & Wp_2[\![\langle d, v_1 := 1 \,;\, v_2 := 2\rangle]\!](P) \\ & \quad = (\pi_1 \,;\, \pi_2)(P) \\ & \quad = \pi_1(\{ s \mid \langle s, \pi_2\rangle \in P\} \cup \{\langle s, \rho\rangle \mid \langle s, \rho \,;\, \pi_2\rangle \in P\}) \\ & \quad = \{ s \mid \langle s[1/v_1], \pi_2\rangle \in P\} \\ & \quad = \emptyset. \end{aligned}$$

The point is that P only allows *immediate* terminating computations, whereas the sequential composition has also one intermediate state as reflected in the pair $\langle s[1/v_1], \pi_2\rangle$. In general, if we want to use our predicate transformer semantics to show that a program can achieve certain goals being indifferent as to how it will be reached, we can incorporate such pairs as $\langle s[1/v_1], \pi_2\rangle$ in the predicate P. So the predicate $P \subseteq \mathtt{St}$ has to be 'enhanced' with pairs representing the same input/output information to accommodate for composite elements in $\mathtt{St} + \mathtt{St} \times \Pi_2$.

7.3.13. Definition. For $P \subseteq \mathtt{St}$ define the *enhanced predicate* for total correctness P^{tc} on $\mathtt{St} + \mathtt{St} \times \Pi_2$ by

$$P^{tc} = \bigcup_n P^{tc}_n \quad \text{where} \quad \begin{cases} P^{tc}_0 & = \emptyset \\ P^{tc}_{n+1} & = P \cup \{\langle s, \rho\rangle \mid s \in \rho(P^{tc}_n)\}. \end{cases}$$

By induction on $n \geq 0$, it is straightforward to see that $P^{tc}_n \subseteq P^{tc}_{n+1}$. Furthermore, for every $n \geq 0$, P^{tc}_n is open in the metric topology of the metric space $\mathtt{St} + \mathtt{St} \times \frac{1}{2} \cdot \Pi_2$. This is a consequence of the following lemma.

7.3.14. Lemma. *Let $P \subseteq \mathtt{St}$. For every $n \geq 0$, if $x \in P^{tc}_n$ then $B_{2^{-n}}(x) \subseteq P^{tc}_n$.*

Proof: We prove the above statement by induction on $n \geq 0$. Since $P^{tc}_0 = \emptyset$ the basis case is obviously true.

Assume $z \in P^{tc}_n$ implies $B_{2^{-n}}(z) \subseteq P^{tc}_n$. Let $x \in P^{tc}_{n+1}$. By definition of P^{tc}_{n+1} we have two cases: either $x \in P$ or $x = \langle s, \pi\rangle$ with $s \in \pi(P^{tc}_n)$. In the first case

$$B_{2^{-(n+1)}}(x) = \{x\} \subseteq P \subseteq P^{tc}_{n+1}.$$

In the other case, $q(s, \pi) \subseteq P^{tc}_n$ by Lemma 7.2.2. Let $\langle t, \rho\rangle \in B_{2^{-(n+1)}}(\langle s, \pi\rangle)$. Then $t = s$ and $d_B(\pi, \rho) < 2^{-n}$. Hence $d(q(s, \pi), q(s, \rho)) < 2^{-n}$. By Proposition 2.3.2 it follows that for every $y \in q(s, \rho)$ there exists $z \in q(s, \pi)$ such that $d(y, z) < 2^{-n}$, that is $y \in B_{2^{-n}}(z)$. Since $z \in q(s, \pi) \subseteq P^{tc}_n$, by the induction hypothesis it follows $B_{2^{-n}}(z) \subseteq P^{tc}_n$. Therefore $q(s, \rho) \subseteq P^{tc}_n$. By Lemma 7.2.2, $s \in \rho(P^{tc}_n)$ and hence $\langle s, \rho\rangle \in P^{tc}_{n+1}$. $\square$

Using the property that metric predicate transformers preserve directed unions of metric open sets we have that

$$\pi(P^{tc}) = \pi(\textstyle\bigcup_n P^{tc}_n) = \displaystyle\bigcup_n \pi(P^{tc}_n),$$

for $P \subseteq \mathtt{St}$ and $\pi \in \Pi_2$. This fact will be used later in Theorem 7.5.5 in order to show the correctness of the $Wp_2[\![\cdot]\!]$ semantics with respect to the weakest precondition semantics $Wp_0[\![\cdot]\!]$ given in Chapter 3. Moreover, using the above equation it is immediate to see that for $P \subseteq \mathtt{St}$ the enhanced predicate P^{tc} is the least subset of $\mathtt{St} + \mathtt{St} \times \Pi_2$ satisfying

$$P^{tc} = P \cup \{\langle s, \rho\rangle \mid s \in \rho(P^{tc})\}.$$

Therefore P^{tc} consists of elements of P and those pairs $\langle s, \rho \rangle$ for which s is appropriate in that it will lead to P^{tc} following ρ. We will return to this point in Section 7.5.

Returning to the example for $v_1 := 1 \; ; \; v_2 := 2$ we can now calculate the semantics for P^{tc}, where $P = \{ s \in \mathtt{St} \mid s(v_1) = 1 \;\&\; s(v_2) = 2 \}$. Recall that we write π_1, π_2 for $Wp_2[\![\langle d, v_1 := 1 \rangle]\!]$, $Wp_2[\![\langle d, v_2 := 2 \rangle]\!]$, respectively. We have

$$
\begin{aligned}
& Wp_2[\![\langle d, v_1 := 1 \; ; \; v_2 := 2 \rangle]\!](P^{tc}) \\
& \quad = (\pi_1 \; ; \; \pi_2)(P^{tc}) \\
& \quad = \pi_1(\{ s \mid \langle s, \pi_2 \rangle \in P^{tc} \} \cup \{ \langle s, \rho \rangle \mid \langle s, \rho \; ; \; \pi_2 \rangle \in P^{tc} \}) \\
& \quad = \{ s \mid \langle s[1/v_1], \pi_2 \rangle \in P^{tc} \} \\
& \quad = \{ s \mid s[1/v_1] \in \pi_2(P^{tc}) \} \\
& \quad = \{ s \mid s[1/v_1][2/v_2] \in P^{tc} \} \\
& \quad = \{ s \mid s[1/v_1][2/v_2](v_1) = 1 \;\&\; s[1/v_1][2/v_2](v_2) = 2 \} \\
& \quad = \mathtt{St},
\end{aligned}
$$

which is the result to be expected.

As a second example, we compute the weakest precondition of the statement $(v_1 := 1 \parallel v_2 := 2) \parallel v_3 := 3$ for the enhanced predicate for total correctness of the predicate

$$
P \quad = \quad \{ s \in \mathtt{St} \mid s(v_1) + s(v_2) = s(v_3) \}.
$$

Let $\pi_1 = Wp_2[\![\langle d, v_1 := 1 \rangle]\!]$, $\pi_2 = Wp_2[\![\langle d, v_2 := 2 \rangle]\!]$ and $\pi_3 = Wp_2[\![\langle d, v_3 := 3 \rangle]\!]$. We have

$$
\begin{aligned}
& Wp_2[\![(v_1 := 1 \parallel v_2 := 2) \parallel v_3 := 3]\!](P^{tc}) \\
& \quad = ((\pi_1 \parallel \pi_2) \parallel \pi_3)(P^{tc}) \\
& \quad = (\pi_1 \parallel \pi_2)(\{ s \mid \langle s, \pi_3 \rangle \in P^{tc} \} \cup \{ \langle s, \rho \rangle \mid \langle s, \rho \parallel \pi_3 \rangle \in P^{tc} \}) \cap \\
& \quad \quad \pi_3(\{ s \mid \langle s, \pi_1 \parallel \pi_2 \rangle \in P^{tc} \} \cup \{ \langle s, \rho \rangle \mid \langle s, (\pi_1 \parallel \pi_2) \parallel \rho \rangle \in P^{tc} \}) \\
& \quad = \pi_1(\{ s \mid \langle s, \pi_2 \rangle \in \{ s \mid \langle s, \pi_3 \rangle \in P^{tc} \} \cup \{ \langle s, \rho \rangle \mid \langle s, \rho \parallel \pi_3 \rangle \in P^{tc} \} \} \cup \\
& \quad \quad \quad \{ \langle s, \rho \rangle \mid \langle s, \rho \parallel \pi_2 \rangle \in \{ s \mid \langle s, \pi_3 \rangle \in P^{tc} \} \cup \\
& \quad \quad \quad \quad \quad \quad \{ \langle s, \rho \rangle \mid \langle s, \rho \parallel \pi_3 \rangle \in P^{tc} \} \}) \cap \\
& \quad \quad \pi_2(\{ s \mid \langle s, \pi_1 \rangle \in \{ s \mid \langle s, \pi_3 \rangle \in P^{tc} \} \cup \{ \langle s, \rho \rangle \mid \langle s, \rho \parallel \pi_3 \rangle \in P^{tc} \} \} \cup \\
& \quad \quad \quad \{ \langle s, \rho \rangle \mid \langle s, \rho \parallel \pi_1 \rangle \in \{ s \mid \langle s, \pi_3 \rangle \in P^{tc} \} \cup \\
& \quad \quad \quad \quad \quad \quad \{ \langle s, \rho \rangle \mid \langle s, \rho \parallel \pi_3 \rangle \in P^{tc} \} \}) \cap \\
& \quad \quad \pi_3(\{ s \mid \langle s, \pi_1 \parallel \pi_2 \rangle \in P^{tc} \} \cup \{ \langle s, \rho \rangle \mid \langle s, (\pi_1 \parallel \pi_2) \parallel \rho \rangle \in P^{tc} \})
\end{aligned}
$$

$$\begin{aligned}
&= \pi_1(\{s \mid \langle s, \pi_2 \parallel \pi_3 \rangle \in P^{tc}\} \cup \{\langle s, \rho \rangle \mid \langle s, (\rho \parallel \pi_2) \parallel \pi_3 \rangle \in P^{tc}\}) \cap \\
&\quad \pi_2(\{s \mid \langle s, \pi_1 \parallel \pi_3 \rangle \in P^{tc}\} \cup \{\langle s, \rho \rangle \mid \langle s, (\pi_1 \parallel \rho) \parallel \pi_3 \rangle \in P^{tc}\}) \cap \\
&\quad \pi_3(\{s \mid \langle s, \pi_1 \parallel \pi_2 \rangle \in P^{tc}\} \cup \{\langle s, \rho \rangle \mid \langle s, (\pi_1 \parallel \pi_2) \parallel \rho \rangle \in P^{tc}\}) \\
&= \{s \mid s[1/v_1] \in (\pi_2 \parallel \pi_3)(P^{tc})\} \cap \{s \mid s[2/v_2] \in (\pi_1 \parallel \pi_3)(P^{tc})\} \cap \\
&\quad \{s \mid s[3/v_3] \in (\pi_1 \parallel \pi_2)(P^{tc})\} \\
&= \ldots \\
&= \{s \mid s[1/v_1][2/v_2][3/v_3] \in P^{tc}\} \\
&= \mathtt{St}.
\end{aligned}$$

The example also indicates a more general relationship for parallel compositions of predicate transformers. Let us use $\pi_{1,..,k}$ to denote the left associated parallel composition of $\pi_1, \ldots, \pi_k$, $\pi^r_{1,..,k}$ (with $1 \le r \le k$) for $\pi_{1,..,k}$ but leaving out the operand π_r, and $\hat{\pi}^r_{1,..,k}$ for $\pi_{1,..,k}$ but now replacing the operand π_r by the predicate transformer ρ. We then have

$$\pi_{1,..,k}(P) \;=\; \bigcap_{r=1}^{k} \pi_r(\{s \mid \langle s, \pi^r_{1,..,k} \rangle \in P\} \cup \{\langle s, \rho \rangle \mid \langle s, \hat{\pi}^r_{1,..,k} \rangle \in P\}). \quad (7.4)$$

The above equation can be verified by a straightforward inductive argument using the various definitions.

In both the above examples we have used the semantic function $Wp_2[\![-]\!]$ to study correctness properties. A more involved and probably more interesting example of application of the metric predicate transformer semantics $Wp_2[\![-]\!]$ will be given in Section 7.5, where we will make more precise the connection with the weakest (liberal) precondition semantics introduced in Chapter 3.

7.4 Relationships with state transformers

Next we turn to a state transformer domain Σ_2 suitable for a compositional forward semantics of the language $\mathcal{L}_2$. The metric resumption domain Σ_2 is used to measure the 'goodness' of the weakest precondition semantics Wp_2. We define $(\sigma \in)\Sigma_2$ as a variation of the domain introduced by De Bakker and Zucker in [BZ82]: it is the unique (up to isometry) solution of the domain equation

$$X \;=\; \mathtt{St} \to \mathcal{P}_{co}(\mathtt{St} + \mathtt{St} \times \tfrac{1}{2} \cdot X).$$

The intuition behind the above domain is as follows: statements are functions which deliver for each input state a set consisting of output states and/or pairs composed from the output after one atomic computation step together with a function representing the rest of the computation. The domain Σ_2 comes equipped with the following operations (see [BZ82] for a justification of their well-definedness).

7.4.1. Definition. For every function $f{:}\mathtt{St} \to \mathit{Val}$, individual variable $v \in \mathit{IVar}$ and subset V of $\mathtt{St}$ define the state transformers '$[f/v]$' and '$V{\to}$' in Σ_2 by

$$\begin{array}{rcl} [f/v](s) & = & \{s[f(s)/v]\}, \\ V{\to}(s) & = & \begin{cases} \{s\} & \text{if } s \in V \\ \emptyset & \text{otherwise} \end{cases} \end{array}$$

for every $s \in \mathtt{St}$. The binary operators '$;$', '$\Box$' and '$\|$' on Σ_2 are given by

$$\begin{array}{rcl} \sigma_1 \,;\, \sigma_2(s) & = & \{\langle t, \sigma_2\rangle \mid t \in \sigma_1(s)\} \cup \{\langle t, \tau \,;\, \sigma_2\rangle \mid \langle t, \tau\rangle \in \sigma_1(s)\}, \\ \sigma_1 \,\Box\, \sigma_2(s) & = & \sigma_1(s) \cup \sigma_2(s), \\ \sigma_1 \,\|\, \sigma_2(s) & = & \{\langle t, \sigma_2\rangle \mid t \in \sigma_1(s)\} \cup \{\langle t, \tau \,\|\, \sigma_2\rangle \mid \langle t, \tau\rangle \in \sigma_1(s)\} \cup \\ & & \{\langle t, \sigma_1\rangle \mid t \in \sigma_2(s)\} \cup \{\langle t, \sigma_1 \,\|\, \tau\rangle \mid \langle t, \tau\rangle \in \sigma_2(s)\}, \end{array}$$

for $\sigma_1, \sigma_2 \in \Sigma_2$ and $s \in \mathtt{St}$.

The above operations are all we need to give a compositional forward semantics for $\mathcal{L}_2$ along the lines of Definition 7.3.11. On the basis of the general isomorphisms between state and predicate transformers studied in Theorem 6.2.12 we can formulate the relationship between Σ_2 and Π_2. The two domains are isomorphic. However, the isomorphism is not an order-isomorphism but an isometry between the complete metric space Σ_2 and the complete metric space Π_2. Since the isomorphism will preserve the operations defined above, it follows that the backward and the forward semantics of $\mathcal{L}_2$ are isomorphic.

We need some preparatory steps to prove an isomorphism between the two semantic domains.

7.4.2. Definition. Define the domain transformation $pt : \Sigma_2 \to \Pi_2$ by

$$\begin{array}{rcl} pt(\sigma)(P) & = & \{s \in \mathtt{St} \mid \; (t \in \sigma(s) \;\Rightarrow\; t \in P)\ \& \\ & & \qquad\qquad (\langle t, \tau\rangle \in \sigma(s) \;\Rightarrow\; \langle t, pt(\tau)\rangle \in P)\}, \end{array}$$

for all $\sigma \in \Sigma_2$, and $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$.

In order to justify the well-definedness of pt we introduce a higher-order transformation in a such way that it is contractive and pt is its unique fixed point. Define the higher-order transformation $\Omega_{pt} : (\Sigma_2 \xrightarrow{1} \Pi_2) \to (\Sigma_2 \xrightarrow{1} \Pi_2)$ by

$$\begin{array}{rcl} \Omega_{pt}(tr)(\sigma)(P) & = & \{s \in \mathtt{St} \mid \; (t \in \sigma(s) \;\Rightarrow\; t \in P)\ \& \\ & & \qquad\qquad (\langle t, \tau\rangle \in \sigma(s) \;\Rightarrow\; \langle t, tr(\tau)\rangle \in P)\}, \end{array}$$

for all non-expansive $tr : \Sigma_2 \to \Pi_2$, $\sigma \in \Sigma_2$, and $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$.

7.4.3. Lemma. *Let $tr : \Sigma_2 \to \Pi_2$ be non-expansive and $\sigma \in \Sigma_2$. Then*

(i) $\Omega_{pt}(tr)(\sigma) \in \Pi_2$,

(ii) $\Omega_{pt}(tr)$ *is non-expansive,*
(iii) *the transformation* Ω_{pt} *is* $\frac{1}{2}$*-contractive.*
Therefore Ω_{pt} *has a unique fixed point which is the function* pt *of Definition 7.4.2.*

Proof: (i) Choose arbitrary $tr : \Sigma_2 \xrightarrow{1} \Pi_2$ and $\sigma \in \Sigma_2$, and put $\pi = \Omega_{pt}(tr)(\sigma)$. Multiplicativity of π is straightforwardly checked. Preservation of directed joins of opens by π is verified as follows. Take $\mathcal{V}$ to be a directed set of metric opens of $\mathtt{St} + \mathtt{St} \times \frac{1}{2} \cdot \Pi_2$. Note that the set

$$\{t \in \mathtt{St} \mid t \in \sigma(s)\} \quad \cup \quad \{\langle t, tr(\tau)\rangle \in \mathtt{St} \times \Pi_2 \mid \langle t, \tau\rangle \in \sigma(s)\}$$

is compact: its first constituent equals $\sigma(s) \cap \mathtt{St}$, which is the intersection of a compact and a closed set. Its second constituent is the continuous image – under $\langle id_{\mathtt{St}}, tr\rangle$ – of the compact set $\sigma(s) \cap (\mathtt{St} \times \Sigma_2)$. Therefore,

$$\begin{aligned}
& s \in \pi(\textstyle\bigcup \mathcal{V}) \\
\Leftrightarrow\ & \{t \in \mathtt{St} \mid t \in \sigma(s)\} \cup \\
& \quad \{\langle t, tr(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma(s)\} \subseteq \textstyle\bigcup \mathcal{V} \qquad [\text{definition } \Omega_{pt}] \\
\Leftrightarrow\ & \exists P \in \mathcal{V}: \{t \mid t \in \sigma(s)\} \cup \\
& \qquad \{\langle t, tr(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma(s)\} \subseteq P \qquad [\text{compactness}] \\
\Leftrightarrow\ & s \in \textstyle\bigcup \{\pi(P) \mid P \in \mathcal{V}\}.
\end{aligned}$$

(ii) For arbitrary $tr : \Sigma_2 \xrightarrow{1} \Pi_2$, $\sigma \in \Sigma_2$, and $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$ we have

$$\begin{aligned}
& s \in \Omega_{pt}(tr)(\sigma)(P) \\
\Leftrightarrow\ & (t \in \sigma(s) \ \Rightarrow\ t \in P)\ \& \\
& (\langle t, \tau\rangle \in \sigma(s) \ \Rightarrow\ \langle t, tr(\tau)\rangle \in P) \qquad [\text{definition } \Omega_{pt}] \\
\Leftrightarrow\ & \{t \in \mathtt{St} \mid t \in \sigma(s)\} \cup \{\langle t, tr(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma(s)\} \subseteq P.
\end{aligned}$$

Therefore, by Lemma 7.2.2,

$$q(s, \Omega_{pt}(tr)(\sigma)) = \{t \in \mathtt{St} \mid t \in \sigma(s)\} \cup \{\langle t, tr(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma(s)\}. \tag{7.5}$$

Now, let $tr : \Sigma_2 \xrightarrow{1} \Pi_2$ and choose, in order to show the non-expansiveness of $\Omega_{pt}(tr)$, $\sigma_1, \sigma_2 \in \Sigma_2$. We then have (omitting the subscripts of the distance functions)

$$\begin{aligned}
& d(\Omega_{pt}(tr)(\sigma_1), \Omega_{pt}(tr)(\sigma_2)) \\
=\ & \sup\{\, d(q(s, \Omega_{pt}(tr)(\sigma_1)), q(s, \Omega_{pt}(tr)(\sigma_2))) \mid s \in \mathtt{St}\,\} \\
=\ & \max\{\sup\{\, d(\{t \in \mathtt{St} \mid t \in \sigma_1(s)\}, \{t \in \mathtt{St} \mid t \in \sigma_2(s)\}) \mid s \in \mathtt{St}\,\}, \\
& \qquad \sup\{\, d(\{\langle t, tr(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma_1(s)\}, \\
& \qquad\qquad \{\langle t, tr(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma_2(s)\}) \mid s \in \mathtt{St}\,\}\,\} \qquad [\text{Equation (7.5)}] \\
\leq\ & \sup\{\, d(\sigma_1(s), \sigma_2(s) \mid s \in \mathtt{St}\,\} \qquad [tr \text{ non-expansive}] \\
=\ & d(\sigma_1, \sigma_2).
\end{aligned}$$

(iii) Pick any $tr_1, tr_2 \in \Sigma_2 \xrightarrow{1} \Pi_2$. Then we have

$$
\begin{aligned}
& d(\Omega_{pt}(tr_1), \Omega_{pt}(tr_2)) \\
& = \sup\{\, d(\Omega_{pt}(tr_1)(\sigma), \Omega_{pt}(tr_2)(\sigma)) \mid \sigma \in \Sigma_2 \,\} \quad [\text{distance on } \Sigma_2 \to \Pi_2] \\
& = \sup\{\, d(q(s, \Omega_{pt}(tr_1)(\sigma)), q(s, \Omega_{pt}(tr_2)(\sigma))) \mid \sigma \in \Sigma_2 \;\&\; s \in \mathtt{St} \,\} \\
& \leq \sup\{\, d(\{\langle t, tr_1(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma(s)\}, \\
& \qquad \{\langle t, tr_2(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma(s)\}) \mid \sigma \in \Sigma_2 \;\&\; s \in \mathtt{St} \,\} \quad [\text{Eq. (7.5)}] \\
& \leq \sup\{\, \tfrac{1}{2} d(tr_1(\tau), tr_2(\tau)) \mid \sigma \in \Sigma_2 \;\&\; \langle t, \tau\rangle \in \sigma(s) \,\} \\
& \leq \tfrac{1}{2} d(tr_1, tr_2) \quad . \qquad \square
\end{aligned}
$$

From the proof of the Lemma 7.4.3 we have, taking pt for tr,

$$q(s, \Omega_{pt}(pt)(\sigma)) \quad = \quad \{t \in \mathtt{St} \mid t \in \sigma(s)\} \cup \{\langle t, pt(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma(s)\}.$$

Since pt is the unique fixed point of Ω_{pt}, the following corollary is immediate.

7.4.4. Corollary. *For every $\sigma \in \Sigma_2$ and $s \in \mathtt{St}$, it holds that*

$$q(s, pt(\sigma)) = \{t \in \mathtt{St} \mid t \in \sigma(s)\} \cup \{\langle t, pt(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma(s)\} \qquad \square$$

Next we define a function which maps a predicate transformer to a state transformer.

7.4.5. Definition. Define the domain transformation $st : \Pi_2 \to \Sigma_2$ by

$$st(\pi)(s) \quad = \quad \{t \in \mathtt{St} \mid t \in q(s, \pi)\} \cup \{\langle t, st(\rho)\rangle \mid \langle t, \rho\rangle \in q(s, \pi)\},$$

for all $\pi \in \Pi_2$, and $s \in \mathtt{St}$.

Again, to show the well-definedness of the function st we introduce the higher-order transformation $\Omega_{st} : (\Pi_2 \xrightarrow{1} \Sigma_2) \to (\Pi_2 \xrightarrow{1} \Sigma_2)$. It is given by

$$\Omega_{st}(tr)(\pi)(s) \quad = \quad \{t \in \mathtt{St} \mid t \in q(s, \pi)\} \cup \{\langle t, tr(\rho)\rangle \mid \langle t, \rho\rangle \in q(s, \pi)\},$$

for all non-expansive $tr : \Pi_2 \to \Sigma_2$, $\pi \in \Pi_2$, and $s \in \mathtt{St}$.

7.4.6. Lemma. *Let $tr : \Pi_2 \to \Sigma_2$ be non-expansive and $\pi \in \Pi_2$. Then*
(i) *$\Omega_{st}(tr)(\pi) \in \Sigma_2$,*
(ii) *$\Omega_{st}(tr)$ is non-expansive, and*
(iii) *the transformation Ω_{st} is $\tfrac{1}{2}$-contractive.*

Proof: Similar to the proof of Lemma 7.4.3 and hence omitted. $\square$

We are now in a position to prove the isometry between Σ_2 and Π_2.

7.4.7. Theorem. *The non-expansive functions $pt : \Sigma_2 \to \Pi_2$ and $st : \Pi_2 \to \Sigma_2$ form an isometry between Σ_2 and Π_2.*

Proof: It is enough to check that $pt : \Sigma_2 \to \Pi_2$ and $st : \Pi_2 \to \Sigma_2$ satisfy

$$st \circ pt = id_{\Sigma_2} \quad \text{and} \quad pt \circ st = id_{\Pi_2}.$$

First we verify $st \circ pt = id_{\Sigma_2}$. Note that by the respective definitions and Corollary 7.4.4,

$$\begin{aligned} & st(pt(\sigma))(s) \\ &= \{t \in \mathtt{St} \mid t \in q(s, pt(\sigma))\} \cup \{\langle t, st(\rho)\rangle \mid \langle t, \rho\rangle \in q(s, pt(\sigma))\} \\ &= \{t \in \mathtt{St} \mid t \in \sigma(s)\} \cup \{\langle t, st(pt(\tau))\rangle \mid \langle t, \tau\rangle \in \sigma(s)\}. \end{aligned}$$

Suppressing the subscript Σ_2 for the moment, by the above characterization of $st(pt(\sigma))(s)$ we derive

$$\begin{aligned} & d(id, st \circ pt) \\ &= \sup\{\, d(\sigma(s), st(pt(\sigma))(s)) \mid \sigma \in \Sigma_2 \;\&\; s \in \mathtt{St} \,\} \\ &\leq \sup\{\, d(\langle t, \tau\rangle, \langle t, st(pt(\tau))\rangle) \mid \sigma \in \Sigma_2 \;\&\; s \in \mathtt{St} \;\&\; \langle t, \tau\rangle \in \sigma(s) \,\} \\ &\leq \tfrac{1}{2} \sup\{\, d(\tau, st(pt(\tau))) \mid \sigma \in \Sigma_2 \;\&\; s \in \mathtt{St} \;\&\; \langle t, \tau\rangle \in \sigma(s) \,\} \\ &\leq \tfrac{1}{2} d(id, st \circ pt). \end{aligned}$$

From this we conclude $st \circ pt = id_{\Sigma_2}$. Likewise, but slightly simpler, one derives

$$d_{\Pi_2}(id_{\Pi_2}, pt \circ st) \quad \leq \quad \tfrac{1}{2} d_{\Pi_2}(id_{\Pi_2}, pt \circ st)$$

and, consequently, $pt \circ st = id_{\Pi_2}$. $\square$

Next we prove that both the domain transformations pt and st preserve assignments and conditionals as well as the operations of sequential composition, choice and parallel composition. Since they form an isomorphism, it is enough to prove the result only for pt.

7.4.8. Lemma. *Let $f : \mathtt{St} \to Val$, $v \in IVar$, $V \subseteq \mathtt{St}$, and $\sigma_1, \sigma_2 \in \Sigma_2$. Then*
(i) $pt([f/v]) = [f/v]$;
(ii) $pt(V\!\to) = V\!\to$;
(iii) $pt(\sigma_1 \,;\, \sigma_2) = pt(\sigma_1) \,;\, pt(\sigma_2)$;
(iv) $pt(\sigma_1 \,\Box\, \sigma_2) = pt(\sigma_1) \,\Box\, pt(\sigma_2)$;
(v) $pt(\sigma_1 \parallel \sigma_2) = pt(\sigma_1) \parallel pt(\sigma_2)$.

Proof: (i) For every $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$ we have

$$\begin{aligned} & pt([f/v])(P) \\ &= \{s \in \mathtt{St} \mid s[f(s)/v] \in P\} \quad [\text{definition } [f/v] \text{ in } \Sigma_2] \\ &= [f/v](P) \quad [\text{definition } [f/v] \text{ in } \Pi_2] \end{aligned}$$

(ii) Let $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$. Then

$$\begin{aligned}&pt(V\!\rightarrow)(P)\\&\quad= \{s \in \mathtt{St} \mid s \in V \;\Rightarrow\; s \in P\} \qquad [\text{definition } V\!\rightarrow \text{ in } \Sigma_2]\\&\quad= V\!\rightarrow\!(P). \qquad [\text{definition } V\!\rightarrow \text{ in } \Pi_2]\end{aligned}$$

(iii) First notice the following

$$\begin{aligned}&s \in \pi_1 \,;\, \pi_2(P)\\&\Leftrightarrow\; s \in \pi_1(\{t \in \mathtt{St} \mid \langle t, \pi_2\rangle \in P\} \cup \{\langle t, \rho\rangle \mid \langle t, \rho \,;\, \pi_2\rangle \in P\})\\&\Leftrightarrow\; q(s, \pi_1) \subseteq \{t \in \mathtt{St} \mid \langle t, \pi_2\rangle \in P\} \cup\\&\qquad\qquad \{\langle t, \rho\rangle \mid \langle t, \rho \,;\, \pi_2\rangle \in P\} \qquad [\text{Lemma 7.2.2}]\\&\Leftrightarrow\; \{\langle t, \pi_2\rangle \mid t \in q(s, \pi_1)\} \cup \{\langle t, \rho \,;\, \pi_2\rangle \mid \langle t, \rho\rangle \in q(s, \pi_1)\} \subseteq P.\end{aligned}$$

This means, by Lemma 7.2.2 that

$$q(s, \pi_1 \,;\, \pi_2) \;=\; \{\langle t, \pi_2\rangle \mid t \in q(s, \pi_1)\} \cup \{\langle t, \rho \,;\, \pi_2\rangle \mid \langle t, \rho\rangle \in q(s, \pi_1)\}.$$

Hence, for $\pi_1 = pt(\sigma_1)$, $\pi_2 = pt(\sigma_2)$ and $\rho = pt(\tau)$ (which is always the case as pt is an isomorphism), we have

$$\begin{aligned}q(s, pt(\sigma_1) \,;\, pt(\sigma_2)) \;=\;& \{\langle t, pt(\sigma_2)\rangle \mid t \in q(s, pt(\sigma_1))\} \cup\\& \{\langle t, pt(\tau) \,;\, pt(\sigma_2)\rangle \mid \langle t, pt(\tau)\rangle \in q(s, pt(\sigma_1))\}.\end{aligned}$$

On the other hand we have also

$$\begin{aligned}&q(s, pt(\sigma_1 \,;\, \sigma_2))\\&\quad= \{t \mid t \in \sigma_1 \,;\, \sigma_2(s)\} \cup \{\langle t, pt(\tau)\rangle \mid \langle t, \tau\rangle \in \sigma_1 \,;\, \sigma_2(s)\} \qquad [\text{Corollary 7.4.4}]\\&\quad= \{\langle t, pt(\sigma_2)\rangle \mid t \in \sigma_1(s)\} \cup \{\langle t, pt(\tau;\sigma_2\rangle \mid \langle t, \tau\rangle \in \sigma_1(s)\} \qquad [\text{Definition `;'}]\\&\quad= \{\langle t, pt(\sigma_2)\rangle \mid t \in q(s, pt(\sigma_1))\} \cup\\&\qquad \{\langle t, pt(\tau \,;\, \sigma_2)\rangle \mid \langle t, pt(\tau)\rangle \in q(s, pt(\sigma_1))\}. \qquad [\text{Corollary 7.4.4}]\end{aligned}$$

It is now easy to verify that, for a fixed $\sigma_2 \in \Sigma_2$,

$$d(q(s, pt(\sigma_1) \,;\, pt(\sigma_2)), q(s, pt(\sigma_1 \,;\, \sigma_2))) \le \tfrac{1}{2} d(pt(-) \,;\, pt(\sigma_2), pt(- \,;\, \sigma_2)).$$

Therefore we can conclude that

$$\begin{aligned}&d(pt(- \,;\, \sigma_2), pt(-) \,;\, pt(\sigma_2))\\&\quad= \sup\{\, d(pt(\sigma_1 \,;\, \sigma_2), pt(\sigma_1) \,;\, pt(\sigma_2)) \mid \sigma_1 \in \Sigma_2 \,\}\\&\quad= \sup\{\, d(q(s, pt(\sigma_1 \,;\, \sigma_2)), q(s, pt(\sigma_1) \,;\, pt(\sigma_2))) \mid \sigma_1 \in \Sigma_2 \;\&\; s \in \mathtt{St} \,\}\\&\quad\le \sup\{\, \tfrac{1}{2} d(pt(- \,;\, \sigma_2), pt(-) \,;\, pt(\sigma_2)) \mid \sigma_1 \in \Sigma_2 \,\}\\&\quad= \tfrac{1}{2} d(pt(- \,;\, \sigma_2), pt(-) \,;\, pt(\sigma_2)).\end{aligned}$$

The above implies $pt(\sigma_1 \,;\, \sigma_2) = pt(\sigma_1) \,;\, pt(\sigma_2)$.

(iv) We have, for every $P \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$,

$$\begin{aligned}&pt(\sigma_1 \,\Box\, \sigma_2)(P)\\&\quad= \{s \mid (t \in \sigma_1 \,\Box\, \sigma_2(s) \;\Rightarrow\; t \in P) \;\&\\&\qquad\quad (\langle t, \tau\rangle \in \sigma_1 \,\Box\, \sigma_2(s) \;\Rightarrow\; \langle t, pt(\tau)\rangle \in P)\}\end{aligned}$$

$$\begin{aligned}
&= \{s \mid (t \in \sigma_1(s) \;\Rightarrow\; t \in P) \;\&\; (\langle t, \tau\rangle \in \sigma_1(s) \;\Rightarrow\; \langle t, pt(\tau)\rangle \in P)\} \cap \\
&\quad\;\, \{s \mid (t \in \sigma_2(s) \;\Rightarrow\; t \in P) \;\&\; (\langle t, \tau\rangle \in \sigma_2(s) \;\Rightarrow\; \langle t, pt(\tau)\rangle \in P)\} \\
&= pt(\sigma_1)(P) \cap pt(\sigma_2)(P) \\
&= pt(\sigma_1) \,\Box\, pt(\sigma_2)(P).
\end{aligned}$$

(v) The proof is similar to the proof of point (iii) and is hence omitted. □

The above lemma ensures that the predicate transformer semantics $Wp_2[\![-]\!]$ is isomorphic to the forward semantics with the operations given in Definition 7.4.1.

7.5 Partial and total correctness

In the previous section we have shown the correctness of the semantics $Wp_2[\![-]\!]$ with respect to a forward semantics establishing a duality between the domain Π_2 of metric predicate transformers and the domain Σ_2 of state transformers. In this section we answer the question concerning the correctness of the domain Π_2 with respect to the three domains of predicate transformers $PT_T(\mathtt{St}, \mathtt{St})$, $PT_P(\mathtt{St}, \mathtt{St})$, and $PT_N(\mathtt{St}, \mathtt{St})$ introduced in Chapter 3.

A metric predicate transformer in Π_2 records every intermediate step of the computations it denotes. If we want to calculate the set of those input states $s \in \mathtt{St}$ for which each computation denoted by π terminates in a final state satisfying a predicate $P \subseteq \mathtt{St}$ then we have to enhance P to obtain a predicate on $\mathtt{St} + \mathtt{St} \times \Pi_2$. The idea is to define the enhancement of $P \subseteq \mathtt{St}$ by incorporating in P all those pairs $\langle s, \rho\rangle \in \mathtt{St} \times \Pi_2$ such that all computations denoted by ρ started at s terminate and satisfy P.

7.5.1. Definition. For $P \subseteq \mathtt{St}$ define the *enhanced predicates* P^{tc} for total correctness as the least subset of $\mathtt{St} + \mathtt{St} \times \Pi_2$ satisfying the equation

$$X \;=\; P \cup \{\langle s, \rho\rangle \in \mathtt{St} \times \Pi_2 \mid s \in \rho(X)\}.$$

The enhancement of P for partial correctness is defined as the greatest subset P^{pc} of $\mathtt{St} + \mathtt{St} \times \Pi_2$ satisfying the above equation.

In Definition 7.3.13 we have given the enhancement of P for total correctness in terms of its approximants. Similarly, the enhancement of $P \subseteq \mathtt{St}$ for partial correctness can be defined by

$$P^{pc} = \bigcap_n P^{pc}_n \quad \text{where} \quad \left\{ \begin{array}{lcl} P^{pc}_0 & = & \mathtt{St} + \mathtt{St} \times \Pi_2 \\ P^{pc}_{n+1} & = & P \cup \{\langle s, \rho\rangle \mid s \in \rho(P^{pc}_n)\}. \end{array} \right.$$

The next theorem shows that the enhancement for total and partial correctness of a predicate P is appropriate for the study of partial and total correctness properties in the domain Π_2.

7.5.2. Theorem. *Let* $\pi \in \Pi_2$. *Then*
(i) $\lambda P \subseteq \mathtt{St}.\pi(P^{tc}) \in PT_T(\mathtt{St},\mathtt{St})$;
(ii) $\lambda P \subseteq \mathtt{St}.\pi(P^{pc}) \in PT_P(\mathtt{St},\mathtt{St})$.

Proof: (i) By Corollary 6.2.13 it is enough to prove that the predicate transformer $\lambda P.\pi(P^{tc})$ is Scott-continuous and preserves finite non-empty intersections.

To prove Scott-continuity, let $\mathcal{V}$ be a directed set of subsets of $\mathtt{St}$. We first show that for all $n \geq 0$,

$$(\bigcup \mathcal{V})^{tc}_n = \bigcup\{P^{tc}_n \mid P \in \mathcal{V}\}. \tag{7.6}$$

We proceed by induction on n. For $n = 0$, the Equation (7.6) is obviously true. Assume it holds for $n = k$, then

$$\begin{aligned}
&(\bigcup \mathcal{V})^{tc}_{k+1}\\
&= \bigcup \mathcal{V} \cup \{\langle s,\rho\rangle \mid s \in \rho((\bigcup \mathcal{V})^{tc}_k)\}\\
&= \bigcup \mathcal{V} \cup \{\langle s,\rho\rangle \mid s \in \rho(\bigcup\{P^{tc}_k \mid P \in \mathcal{V}\})\} && \text{[induction hypothesis]}\\
&= \bigcup \mathcal{V} \cup \bigcup\{\{\langle s,\rho\rangle \mid s \in \rho(P^{tc}_k)\} \mid P \in \mathcal{V}\} && \text{[Lemma 7.3.14, } \rho \text{ continuous]}\\
&= \bigcup\{P^{tc}_{k+1} \mid P \in \mathcal{V}\}.
\end{aligned}$$

Hence $(\bigcup \mathcal{V})^{tc} = \bigcup\{P^{tc} \mid P \in \mathcal{V}\}$, from which it follows that $\lambda P.\pi(P^{tc})$ is Scott continuous. Preservation of binary intersections follows similarly because, for $P, Q \subseteq \mathtt{St}$ it holds that

$$(P \cap Q)^{tc}_n = P^{tc}_n \cap Q^{tc}_n,$$

for all n. The above can be proved by induction on n.

(ii) Since predicate transformers in Π_2 are top-preserving (because they preserve arbitrary intersections), $\mathtt{St}^{pc} = \mathtt{St} + \mathtt{St} \times \Pi_2$. Hence $\pi(\mathtt{St}^{pc}) = \mathtt{St}$. Also, if $\mathcal{V}$ is a non empty set of predicates on $\mathtt{St}$ then

$$(\bigcap \mathcal{V})^{pc} = \bigcap\{P^{pc} \mid P \in \mathcal{V}\}$$

follows immediately from the characterization of the enhancement for partial correctness as the countable intersection of its approximants. Since π preserves arbitrary intersections we obtain that $\lambda P.\pi(P^{pc})$ is a partial correctness predicate transformer. □

Not only partial and total correctness are encoded in a predicate transformer in Π_2. Also their combination is present in the sense of Equation (3.9).

7.5.3. Theorem. *For every* $\pi \in \Pi_2$ *and* $P \subseteq \mathtt{St}$,

$$\pi(P^{tc}) = \pi(\mathtt{St}^{tc}) \cap \pi(P^{pc}).$$

Proof: The inclusion from left to right follows because π is monotone, $P \subseteq \mathtt{St}$ implies $P^{tc} \subseteq \mathtt{St}^{tc}$, and $P^{tc} \subseteq P^{pc}$. To prove the other direction we first show that for all $n \geq 0$,

$$P^{tc}_n \supseteq \mathtt{St}^{tc}_n \cap P^{pc}. \tag{7.7}$$

We proceed by induction on $n \geq 0$. In case $n = 0$ then both P_0^{tc} and $\mathtt{St}_0^{tc}$ are the empty set. Hence (7.7) holds. Assume now (7.7) holds for $n = k$ and let $x \in \mathtt{St}_{k+1}^{tc} \cap P^{pc}$. There are two cases. If $x \in \mathtt{St}$ then $x \in P$ by definition of P^{pc}. Otherwise $x = \langle s, \rho \rangle \in \mathtt{St} \times \Pi_2$. Since $\langle s, \rho \rangle \in \mathtt{St}_{k+1}^{tc}$, $s \in \rho(\mathtt{St}_k^{tc})$. Also, since $\langle s, \rho \rangle \in P^{pc}$, $s \in \rho(P^{pc})$. Because ρ preserves intersections and by the induction hypothesis,

$$s \in \rho(\mathtt{St}_k^{tc}) \cap \rho(P^{pc}) = \rho(\mathtt{St}_k^{tc} \cap P^{pc}) \subseteq \rho(P_k^{tc}).$$

By definition of P_{k+1}^{tc} it follows that $\langle s, \rho \rangle \in P_{k+1}^{tc}$. Hence (7.7) holds for every $n \geq 0$. As a consequence

$$P^{tc} \supseteq \mathtt{St}^{tc} \cap P^{pc}$$

from which we can conclude $\pi(P^{tc}) \subseteq \pi(\mathtt{St}^{tc} \cap P^{pc}) = \pi(\mathtt{St}^{tc}) \cap \pi(P^{pc})$. □

The enhancement of a predicate for total correctness preserves the semantical operators corresponding to choice and to sequential composition. We cannot have a similar result for the parallel composition since it would allow for a compositional semantics for $\mathcal{L}_2$ by means of total correctness predicate transformers.

7.5.4. Lemma. *Let $\pi_1, \pi_2 \in \Pi_2$ and $P \subseteq \mathtt{St}$. Then*
(i) $(\pi_1 \;\Box\; \pi_2)(P^{tc}) = \pi_1(P^{tc}) \cap \pi_2(P^{tc})$;
(ii) $(\pi_1 \,;\, \pi_2)(P^{tc}) = \pi_1(\pi_2(P^{tc})^{tc})$.

Proof: (i) Immediate from Definition 7.3.2.
(ii) To begin with, we prove by induction on n that

$$(\pi_1 \,;\, \pi_2)(P_{n+1}^{tc}) \subseteq \pi_1(\pi_2(P_n^{tc})_{n+1}^{tc}). \tag{7.8}$$

For $n = 0$ we have

$$\begin{aligned}
&(\pi_1 \,;\, \pi_2)(P_1) \\
&= \pi_1(\{s \mid \langle s, \pi_2 \rangle \in P_1\} \cup \{\langle s, \rho \rangle \mid \langle s, \rho \,;\, \pi_2 \rangle \in P_1\}) \quad \text{[Definition 7.3.2]} \\
&= \pi_1(\{s \mid s \in \pi_2(P_0^{tc})\} \cup \{\langle s, \rho \rangle \mid s \in (\rho \,;\, \pi_2)(P_0^{tc})\}) \quad \text{[Definition 7.3.13]} \\
&= \pi_1(\pi_2(P_0^{tc}) \cup \{\langle s, \rho \rangle \mid s \in \rho(P_0^{tc})\}) \quad [P_0^{tc} = \emptyset,\ (\rho \,;\, \pi_2)(\emptyset) = \rho(\emptyset)] \\
&= \pi_1(\pi_2(P_0^{tc}) \cup \{\langle s, \rho \rangle \mid s \in \rho(\pi_2(P_0^{tc})_0^{tc})\}) \quad [\pi_2(P_0^{tc})_0^{tc} = P_0^{tc} = \emptyset] \\
&= \pi_1(\pi_2(P_0^{tc})_1). \quad \text{[Definition 7.3.13]}
\end{aligned}$$

Assume now (7.8) holds for $n = k$ and we prove it for $n = k + 1$.

$$\begin{aligned}
&(\pi_1 \,;\, \pi_2)(P_{k+2}^{tc}) \\
&= \pi_1(\{s \mid \langle s, \pi_2 \rangle \in P_{k+2}^{tc}\} \cup \{\langle s, \rho \rangle \mid \langle s, \rho ; \pi_2 \rangle \in P_{k+2}^{tc}\}) \quad \text{[Definition 7.3.2]} \\
&= \pi_1(\{s \mid s \in \pi_2(P_{k+1}^{tc})\} \cup \{\langle s, \rho \rangle \mid s \in (\rho ; \pi_2)(P_{k+1}^{tc})\}) \quad \text{[Definition 7.3.13]} \\
&\subseteq \pi_1(\pi_2(P_{k+1}^{tc}) \cup \{\langle s, \rho \rangle \mid s \in \rho(\pi_2(P_k^{tc})_{k+1}^{tc})\}) \quad \text{[induction, } \pi_1 \text{ monotone]} \\
&\subseteq \pi_1(\pi_2(P_{k+1}^{tc}) \cup \{\langle s, \rho \rangle \mid s \in \rho(\pi_2(P_{k+1}^{tc})_{k+1}^{tc})\}) \quad [P_k^{tc} \subseteq P_{k+1}^{tc}] \\
&= \pi_1(\pi_2(P_{k+1}^{tc})_{k+2}^{tc}). \quad \text{[Definition 7.3.13]}
\end{aligned}$$

Since π_1, π_2, and $\pi_1 ; \pi_2$ preserve directed unions of opens, and all P_n^{tc}'s are opens (Lemma 7.3.14) we obtain

$$\begin{aligned}
&(\pi_1 ; \pi_2)(P^{tc}) \\
&= (\pi_1 ; \pi_2)(\bigcup_n P_n^{tc}) \\
&= (\pi_1 ; \pi_2)(\bigcup_n P_{n+1}^{tc}) \quad [P_0^{tc} = \emptyset] \\
&= \bigcup_n (\pi_1 ; \pi_2)(P_{n+1}^{tc}) \\
&\subseteq \bigcup_n \pi_1(\pi_2(P_n^{tc})_{n+1}^{tc}) \\
&= \pi_1(\bigcup_{\langle n,m\rangle} \pi_2(P_m^{tc})_{n+1}^{tc}) \\
&= \pi_1(\pi_2(\bigcup_m P_m^{tc})^{tc}) \\
&= \pi_1(\pi_2(P^{tc})^{tc}).
\end{aligned}$$

In order to prove the converse, we first prove by induction on n that

$$\pi_1(\pi_2(P_m^{tc})_n^{tc}) \subseteq (\pi_1 ; \pi_2)(P^{tc}), \tag{7.9}$$

for all $m \geq 0$. The base case is immediate: for $n = 0$ and $m \geq 0$,

$$\begin{aligned}
&\pi_1(\pi_2(P_m^{tc})_0^{tc}) \\
&= \pi_1(\emptyset) \\
&= (\pi_1 ; \pi_2)(\emptyset) \\
&\subseteq (\pi_1 ; \pi_2)(P^{tc}).
\end{aligned}$$

Assume now (7.9) holds for $n = k$. Then, for every $m \geq 0$,

$$\begin{aligned}
&\pi_1(\pi_2(P_m^{tc})_{k+1}^{tc}) \\
&= \pi_1(\pi_2(P_m^{tc}) \cup \{\langle s, \rho\rangle \mid s \in \rho(\pi_2(P_m^{tc})_k^{tc})\}) \quad [\text{definition } \pi_2(P_m^{tc})_{k+1}^{tc}] \\
&= \pi_1(\{s \mid \langle s, \pi_2\rangle \in P_{m+1}^{tc}\} \cup \{\langle s, \rho\rangle \mid s \in \rho(\pi_2(P_m^{tc})_k^{tc})\}) \quad [\text{definition } P_{m+1}^{tc}] \\
&\subseteq \pi_1(\{s \mid \langle s, \pi_2\rangle \in P_{m+1}^{tc}\} \cup \{\langle s, \rho\rangle \mid s \in (\rho ; \pi_2)(P^{tc})\}) \quad [\text{induction}] \\
&\subseteq \pi_1(\{s \mid \langle s, \pi_2\rangle \in P^{tc}\} \cup \{\langle s, \rho\rangle \mid s \in (\rho ; \pi_2)(P^{tc})\}) \quad [P_{m+1}^{tc} \subseteq P^{tc}] \\
&= \pi_1(\{s \mid \langle s, \pi_2\rangle \in P^{tc}\} \cup \{\langle s, \rho\rangle \mid \langle s, \rho ; \pi_2\rangle \in P^{tc}\}) \quad [\text{Definition 7.5.1}] \\
&= (\pi_1 ; \pi_2)(P^{tc}). \quad [\text{Definition 7.3.2}]
\end{aligned}$$

By (7.9) we can conclude

$$\begin{aligned}
&\pi_1(\pi_2(P^{tc})^{tc}) \\
&= \pi_1(\bigcup_n \pi_2(\bigcup_m P_m^{tc})_n^{tc}) \\
&= \bigcup_n \bigcup_m \pi_1(\pi_2(P_m^{tc})_n^{tc}) \\
&\subseteq (\pi_1 ; \pi_2)(P^{tc}).
\end{aligned}$$

Hence we obtain $\pi_1(\pi_2(P^{tc})^{tc}) = (\pi_1 ; \pi_2)(P^{tc})$. □

Similarly one can prove that for $\pi_1, \pi_2 \in \Pi_2$ and $P \subseteq \mathtt{St}$,

$$\begin{aligned}
(\pi_1 \,\Box\, \pi_2)(P^{pc}) &= \pi_1(P^{pc}) \cap \pi_2(P^{pc}); \\
(\pi_1 ; \pi_2)(P^{pc}) &= \pi_1(\pi_2(P^{pc})^{pc}).
\end{aligned}$$

So far we compared the semantic domain Π_2 of metric predicate transformers for concurrency to the semantic domains $PT_T(\mathtt{St},\mathtt{St})$, $PT_P(\mathtt{St},\mathtt{St})$, and $PT_N(\mathtt{St},\mathtt{St})$ of predicate transformers for total and partial correctness. Another enterprise is to compare the metric semantics $Wp_2[\![\cdot]\!]$ with the partial order semantics $Wp_0[\![\cdot]\!]$ and $Wlp_0[\![\cdot]\!]$. Since the language $\mathcal{L}_2$ was introduced at the beginning of this chapter as an extension of the sequential non-deterministic language $\mathcal{L}_0$ of Chapter 3, we expect that both the $Wp_0[\![\cdot]\!]$ and the $Wlp_0[\![\cdot]\!]$ semantics can be retrieved from the $Wp_2[\![\cdot]\!]$ semantics.

There are two main aspects to be considered: declarations of procedure variables in $\mathcal{L}_0$ need not to be guarded as for those in $\mathcal{L}_2$, and the semantics $Wp_2[\![\cdot]\!]$ is defined as the unique fixed point of a contractive function, whereas the semantics $Wp_0[\![\cdot]\!]$ and $Wlp_0[\![\cdot]\!]$ are defined, respectively, as the least and the greatest fixed point of a monotone function.

As for the first aspect, define for every declaration $d \in Decl_0$ a new declaration $d' \in Decl_0$ by

$$d'(x) = true\!\rightarrow\, ;\, d(x)$$

where $x \in PVar$ and $true \in BExp$ such that $\mathcal{B}_\mathcal{V}(true) = \mathtt{St}$. It is immediate to see that if $\langle d, S\rangle \in \mathcal{L}_0$ then $\langle d', S\rangle \in \mathcal{L}_0 \cap \mathcal{L}_2$. Moreover,

$$Wp_0[\![\langle d, S\rangle]\!] = Wp_0[\![\langle d', S\rangle]\!] \quad \text{and} \quad Wlp_0[\![\langle d, S\rangle]\!] = Wlp_0[\![\langle d', S\rangle]\!].$$

Therefore below we will consider, without loss of generality, only programs $\langle d, S\rangle$ in $\mathcal{L}_0$ which are also in $\mathcal{L}_2$. The second aspect requires more attention and it is formally treated in the theorem below.

7.5.5. Theorem. *Let $\langle d, S\rangle \in \mathcal{L}_0 \cap \mathcal{L}_2$ and $P \subseteq \mathtt{St}$. Then*

$$Wp_0[\![\langle d, S\rangle]\!](P) = Wp_2[\![\langle d, S\rangle]\!](P^{tc}).$$

Proof: The proof consists of two parts. In the first part we prove the inclusion from left to right whereas in the second part we prove the converse.

1. Define $\hat{F} : (\mathcal{L}_0 \cap \mathcal{L}_2) \to PT_T(\mathtt{St},\mathtt{St})$ by

$$\hat{F}(\langle d, S\rangle)(P) \quad = \quad Wp_2[\![\langle d, S\rangle]\!](P^{tc}). \tag{7.10}$$

In Chapter 3 we introduced the weakest precondition semantics $Wp_0[\![\cdot]\!]$ as the least fixed point of the monotone function Ψ_T defined in Lemma 3.3.3. Below we prove, by structural induction on S, that for $\langle d, S\rangle \in \mathcal{L}_0 \cap \mathcal{L}_2$ and $P \subseteq \mathtt{St}$,

$$\Psi_T(\hat{F})(\langle d, S\rangle)(P) \quad = \quad \hat{F}(\langle d, S\rangle)(P), \tag{7.11}$$

from which it follows that

$$Wp_0[\![\langle d, S\rangle]\!](P) \quad \subseteq \quad Wp_2[\![\langle d, S\rangle]\!](P^{tc}). \tag{7.12}$$

We expand two typical sub-cases.

$$
\begin{array}{ll}
[x] & \hat{F}(\langle d, x\rangle)(P) \\
& = Wp_2[\![\langle d, x\rangle]\!](P^{tc}) \quad \text{[Equation (7.10)]} \\
& = Wp_2[\![\langle d, d(x)\rangle]\!](P^{tc}) \quad \text{[Definition 7.3.11]} \\
& = \hat{F}(\langle d, d(x)\rangle)(P) \quad \text{[Equation (7.10)]} \\
& = \Psi_T(\hat{F})(\langle d, x\rangle)(P). \quad \text{[Lemma 3.3.3]} \\
[S_1\,;\,S_2] & \hat{F}(\langle d, S_1\,;\,S_2\rangle)(P) \\
& = Wp_2[\![\langle d, S_1\,;\,S_2\rangle]\!](P^{tc}) \quad \text{[Equation (7.10)]} \\
& = (Wp_2[\![\langle d, S_1\rangle]\!]\,;\,Wp_2[\![\langle d, S_2\rangle]\!])(P^{tc}) \quad \text{[Definition 7.3.11]} \\
& = Wp_2[\![\langle d, S_1\rangle]\!](Wp_2[\![\langle d, S_2\rangle]\!](P^{tc})^{tc}) \quad \text{[Lemma 7.5.4]} \\
& = \hat{F}(\langle d, S_1\rangle)(\hat{F}(\langle d, S_2\rangle)(P)) \quad \text{[Equation (7.10)]} \\
& = \Psi_T(\hat{F})(\langle d, S_1\rangle)(\Psi_T(\hat{F})(\langle d, S_2\rangle)(P)) \quad \text{[induction]} \\
& = \Psi_T(\hat{F})(\langle d, S_1\,;\,S_2\rangle)(P). \quad \text{[Lemma 3.3.3]}
\end{array}
$$

2. Next we claim that for $\langle d, S\rangle \in \mathcal{L}_0 \cap \mathcal{L}_2$, $P \subseteq \mathtt{St}$, and $n \geq 0$,

$$Wp_2[\![\langle d, S\rangle]\!](P_n^{tc}) \subseteq Wp_0[\![\langle d, S\rangle]\!](P). \tag{7.13}$$

From the above claim it follows that

$$
\begin{array}{l}
Wp_2[\![\langle d, S\rangle]\!](P^{tc}) \\
= Wp_2[\![\langle d, S\rangle]\!](\bigcup_n P_n^{tc}) \quad \text{[Definition 7.3.13]} \\
= \bigcup_n Wp_2[\![\langle d, S\rangle]\!](P_n^{tc}) \quad \text{[Lemma 7.3.14, } Wp_2[\![\cdot]\!] \text{ open-continuous]} \\
\subseteq Wp_0[\![\langle d, S\rangle]\!](P). \quad \text{[Equation (7.13)]}
\end{array}
$$

The above and Equation (7.12) imply

$$Wp_2[\![\langle d, S\rangle]\!](P^{tc}) = Wp_0[\![\langle d, S\rangle]\!](P).$$

It remains to prove the claim (7.13). We prove it by induction on $n \geq 0$. If $n = 0$ then $P_n^{tc} = \emptyset$. It is easy to see by induction on $wgt_2(S)$ that (7.13) holds. We treat two simple cases as illustration.

$$
\begin{array}{ll}
[x] & Wp_2[\![\langle d, x\rangle]\!](\emptyset) \\
& = Wp_2[\![\langle d, d(x)\rangle]\!](\emptyset) \quad \text{[Definition 7.3.11]} \\
& \subseteq Wp_0[\![\langle d, d(x)\rangle]\!](P) \quad \text{[induction, } wgt_2(d(x)) < wgt_2(x)\text{]} \\
& = Wp_0[\![\langle d, x\rangle]\!](P). \quad \text{[Lemma 3.3.3]} \\
[S_1\,;\,S_2] & Wp_2[\![\langle d, S_1\,;\,S_2\rangle]\!](\emptyset) \\
& = (Wp_2[\![\langle d, S_1\rangle]\!]\,;\,Wp_2[\![\langle d, S_2\rangle]\!])(\emptyset) \quad \text{[Definition 7.3.11]} \\
& = Wp_2[\![\langle d, S_1\rangle]\!](\emptyset) \quad \text{[Definition 7.3.2]} \\
& = Wp_2[\![\langle d, S_1\rangle]\!](Wp_2[\![\langle d, S_2\rangle]\!](P)_0^{tc}) \quad [Wp_2[\![\langle d, S_2\rangle]\!](P)_0^{tc} = \emptyset] \\
& \subseteq Wp_0[\![\langle d, S_1\rangle]\!](Wp_2[\![\langle d, S_2\rangle]\!](P)) \quad \text{[induction, (on } wgt_2(S)\text{))]}
\end{array}
$$

$$= \; Wp_0[\![\langle d, S_1 \,;\, S_2\rangle]\!](P). \qquad \text{[Lemma 3.3.3]}$$

Assume now (7.13) holds for $n = k$ and we prove it for $n = k+1$. As before we proceed by induction on $wgt_2(S)$. We expand two typical sub-cases.

$$\begin{array}{lll}
[x] & & Wp_2[\![\langle d, x\rangle]\!](P^{tc}_{k+1}) \\
& = & Wp_2[\![\langle d, d(x)\rangle]\!](P^{tc}_{k+1}) \quad \text{[Definition 7.3.11]} \\
& \subseteq & Wp_0[\![\langle d, d(x)\rangle]\!](P) \quad \text{[induction, } wgt_2(d(x)) < wgt_2(x)\text{]} \\
& = & Wp_0[\![\langle d, x\rangle]\!](P). \quad \text{[Lemma 3.3.3]} \\
[S_1 \,;\, S_2] & & Wp_2[\![\langle d, S_1 \,;\, S_2\rangle]\!](P^{tc}_{k+1}) \\
& = & (Wp_2[\![\langle d, S_1\rangle]\!] \,;\, Wp_2[\![\langle d, S_2\rangle]\!])(P^{tc}_{k+1}) \quad \text{[Definition 7.3.11]} \\
& \subseteq & Wp_2[\![\langle d, S_1\rangle]\!](Wp_2[\![\langle d, S_2\rangle]\!](P^{tc}_k)^{tc}_{k+1}) \quad \text{[Equation (7.8)]} \\
& \subseteq & Wp_2[\![\langle d, S_1\rangle]\!](Wp_0[\![\langle d, S_2\rangle]\!](P)^{tc}_{k+1}) \quad \text{[induction, } k < k+1\text{]} \\
& \subseteq & Wp_0[\![\langle d, S_1\rangle]\!](Wp_0[\![\langle d, S_2\rangle]\!](P)) \quad \text{[induction (on } wgt_2(S)\text{)]} \\
& = & Wp_0[\![\langle d, S_1 \,;\, S_2\rangle]\!](P). \quad \text{[Lemma 3.3.3]} \qquad \Box
\end{array}$$

In a similar way one can prove that, for $\langle d, S\rangle \in \mathcal{L}_0 \cap \mathcal{L}_2$ and $P \subseteq \mathtt{St}$,

$$Wlp_0[\![\langle d, S\rangle]\!](P) = Wp_2[\![\langle d, S\rangle]\!](P^{pc}).$$

Hence both the weakest precondition semantics $Wp_0[\![\cdot]\!]$ and the weakest liberal precondition semantics $Wlp_0[\![\cdot]\!]$ are encoded in the metric predicate transformer semantics $Wp_2[\![\cdot]\!]$.

We conclude this section with an example of the use of the metric predicate transformer semantics $Wp_2[\![\cdot]\!]$ for calculating a total correctness property of a concurrent program. We treat the accumulator example (stemming from. [US93]). Consider the program

$$v := v + 2^0 \parallel v := v + 2^1 \parallel \ldots \parallel v := v + 2^n.$$

Under the assumption of atomic execution of the assignments $v{:=}v{+}2^i$, we want to calculate the weakest precondition for $P = \{s \mid s(v) = 2^{n+1}\}$. As discussed, we first have to enhance the predicate P for total correctness yielding

$$P^{tc} \;\; = \;\; \{s \mid s(v) = 2^{n+1}\} \cup \{\langle s, \rho\rangle \mid s \in \rho(P^{tc})\}.$$

Let, for convenience, $\pi_i = Wp_2[\![\langle d, v := v + 2^i\rangle]\!]$. So, we are heading, under the notational conventions given in the examples at the end of Section 7.3, for $\pi_{0,..,n}(P^{tc})$. First we establish, for $0 \le i_1 < \ldots < i_k \le n$, the equation

$$\pi_{i_1,..,i_k}(P^{tc}) \;\; = \;\; \{s \mid s(v) + \sum_{r=0}^{k} 2^{i_r} = 2^{n+1}\}, \tag{7.14}$$

by induction on k. We leave the base case, $k = 1$, to the reader. For the induction step, $k+1$ we will employ equation (7.4).

$$
\begin{aligned}
&\pi_{i_1,..,i_{k+1}}(P^{tc}) \\
&\quad = \bigcap_{r=1}^{k+1} \pi_{i_r}(\{s \mid \langle s, \pi^r_{i_1,..,i_k}\rangle \in P^{tc}\} \cup \{\langle s, \rho\rangle \mid \langle s, \hat{\pi}^r_{i_1,..,i_k}\rangle \in P^{tc}\}) \quad [(7.4)] \\
&\quad = \bigcap_{r=1}^{k+1} \{s \mid \langle s[s(v) + 2^{i_r}/v], \pi^r_{i_1,..,i_k}\rangle \in P^{tc}\} \quad [\text{definition } \pi_{i_r}] \\
&\quad = \bigcap_{r=1}^{k+1} \{s \mid s[s(v) + 2^{i_r}/v] \in \pi^r_{i_1,..,i_k}(P^{tc})\} \quad [\text{definition } P^{tc}] \\
&\quad = \bigcap_{r=1}^{k+1} \{s \mid (s(v)+2^{i_r})+(\textstyle\sum_{q=0,q\neq r}^{k+1} 2^{i_q}) = 2^{n+1}\} \quad [\text{induction hypothesis}] \\
&\quad = \bigcap_{r=1}^{k+1} \{s \mid s(v) + 2^{i_r}) + (\textstyle\sum_{q=0}^{k+1} 2^{i_q}) = 2^{n+1}\} \\
&\quad = \{s \mid s(v) + 2^{i_r}) + (\textstyle\sum_{q=0}^{k+1} 2^{i_q}) = 2^{n+1}\}.
\end{aligned}
$$

From the above Equation (7.14) we immediate derive

$$
\begin{aligned}
&\pi_{1,..,n}(P^{tc}) \\
&\quad = \{s \mid s(v) + (\textstyle\sum_{i=0}^{n} 2^i) = 2^{n+1}\} \\
&\quad = \{s \mid s(v) = (2^{n+1} - 1) = 2^{n+1}\} \\
&\quad = \{s \mid s(v) = 1\}.
\end{aligned}
$$

Hence only if we start the accumulator in a state where $v = 1$ can we guarantee that it terminates in a state where $v = 2^{n+1}$.

Enhancement and abstraction

Next we use the isomorphism $\eta : PT_N(\mathtt{St}, \mathtt{St}) \to ST_E(\mathtt{St}, \mathtt{St})$ given in Equation (3.10), the isomorphism $pt : \Sigma_2 \to \Pi_2$, and the enhancement for partial and total correctness to relate the metric domain Σ_2 of De Bakker and Zucker with the domain of Egli-Milner state transformers given in Chapter 3.

We begin by defining a divergence predicate on $\mathtt{St} \times \Sigma_2$. The idea is that a state transformer $\sigma \in \Sigma_2$ diverges in a state $s \in \mathtt{St}$ if it fails to terminate. Termination can be easily expressed in terms of predicate transformers using a total correctness predicate: a state transformer $\sigma \in \Sigma_2$ terminates at input $s \in \mathtt{St}$ if and only if $s \in pt(\sigma)(\mathtt{St}^{tc})$. This fact leads us to the following definition.

7.5.6. Definition. Let $\sigma \in \Sigma_2$ and $s \in \mathtt{St}$. Define the *divergence predicate* $\Uparrow$ on $\mathtt{St} \times \Sigma_2$ to be the complement of the *convergence predicate* $\Downarrow$. The latter is defined by

$$\langle s, \sigma\rangle \Downarrow \text{ if and only if } \exists n \geq 0 \colon \langle s, \sigma\rangle \Downarrow_n$$

where

$$
\begin{aligned}
&\langle s, \sigma\rangle \Downarrow_0 && \text{if and only if } \sigma(s) = \emptyset, \\
&\langle s, \sigma\rangle \Downarrow_{n+1} && \text{if and only if } \langle t, \tau\rangle \in \sigma(s) \Rightarrow \langle t, \tau\rangle \Downarrow_n .
\end{aligned}
$$

The above definition agrees with the intuition about termination in terms of predicate transformers.

7.5.7. Lemma. *For* $s \in \mathtt{St}$ *and* $\sigma \in \Sigma_2$,

$$\langle s, \sigma \rangle \Downarrow \quad \textit{if and only if} \quad s \in pt(\sigma)(\mathtt{St}^{tc}).$$

Proof: We prove by induction on $n \geq 0$ that

$$s \in pt(\sigma)(\mathtt{St}^{tc}_n) \quad \text{if and only if} \quad \langle s, \sigma \rangle \Downarrow_n . \tag{7.15}$$

If $n = 0$ Equation (7.15) holds because

$$pt(\sigma)(\mathtt{St}^{tc}_0) = pt(\sigma)(\emptyset) = \{s \mid \sigma(s) = \emptyset\}.$$

Assume now (7.15) holds for $n = k$. Then

$$\begin{array}{ll} & s \in pt(\sigma)(\mathtt{St}^{tc}_{k+1}) \\ \Leftrightarrow & s \in pt(\sigma)(\mathtt{St} \cup \{\langle s, \rho \rangle \mid s \in \rho(\mathtt{St}^{tc}_k)\}) \quad [\text{Definition } \mathtt{St}^{tc}_{k+1}] \\ \Leftrightarrow & (t \in \sigma(s) \;\Rightarrow\; t \in \mathtt{St}) \;\&\; (\langle t, \tau \rangle \in \sigma(s) \;\Rightarrow\; t \in pt(\tau)(\mathtt{St}^{tc}_k)) \\ & \qquad [\text{Definition } pt] \\ \Leftrightarrow & \langle t, \tau \rangle \in \sigma(s) \;\Rightarrow\; \langle t, \tau \rangle \Downarrow_k . \quad [\text{induction}] \end{array}$$

Therefore (7.15) holds and we can immediately conclude that $\langle s, \sigma \rangle \Downarrow$ if and only if $s \in pt(\sigma)(\mathtt{St}^{tc})$. □

The set of outcomes of the terminating computations of $\sigma \in \Sigma_2$ can be expressed in terms of the corresponding predicate transformer $pt(\sigma)$ and the enhancement for partial correctness, or, more directly, via a flattening function.

7.5.8. Definition. Define the *flattening operator* $|\cdot| : \Sigma_2 \to ST_E(\mathtt{St}, \mathtt{St})$ by

$$|\sigma|(s) = \bigcup_n |\sigma|_n(s)$$

where

$$\begin{array}{lcl} |\sigma|_0(s) & = & \emptyset \\ |\sigma|_{n+1}(s) & = & \{t \in \mathtt{St} \mid t \in \sigma(s) \text{ or } (\langle t', \tau \rangle \in \sigma(s) \;\&\; t \in |\tau|_n(t'))\}, \end{array}$$

for all $\sigma \in \Sigma_2$ and $s \in S$.

We have the following characterization of the flattening operator.

7.5.9. Lemma. *For every* $\sigma \in \Sigma_2$,

$$|\sigma| \;=\; \omega^{-1}(\lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{pc})).$$

Proof: By definition $|\sigma|$ is a Hoare state transformer in $ST_H(\mathtt{St},\mathtt{St})$. Since $\omega^{-1} : PT_P(\mathtt{St},\mathtt{St}) \to ST_H(\mathtt{St},\mathtt{St})$ is part of an isomorphism preserving the opposite order, it is enough to prove that, for all $\sigma \in \Sigma_2$ and $n \geq 0$,

$$|\sigma|_n \quad = \quad \omega^{-1}(\overline{pt(\sigma)}_n), \tag{7.16}$$

where $\overline{pt(\sigma)}_n$ is a shorthand for the predicate transformer $\lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{pc}_n)$. We prove the above equation by induction on $n \geq 0$.

If $n = 0$ then $P^{pc}_0 = \mathtt{St}{+}\mathtt{St}{\times}\Pi_2$. Since $pt(\sigma)$ is top preserving, $pt(P^{pc}_0) = \mathtt{St}$. Hence $\overline{pt(\sigma)}_0(P) = \mathtt{St}$ for all $P \subseteq \mathtt{St}$. It follows that

$$\begin{aligned}
&\omega^{-1}(\lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{pc}_0))(s) \\
&\quad = \ \omega^{-1}(\overline{pt(\sigma)}_0)(s) \qquad [\text{our convention: } \overline{pt(\sigma)}_0 = \lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{pc}_0)] \\
&\quad = \ \bigcap\{P \subseteq \mathtt{St} \mid s \in \overline{pt(\sigma)}_0(P)\} \qquad [\text{definition } \omega^{-1}] \\
&\quad = \ \bigcap\{P \mid P \subseteq \mathtt{St}\} \qquad [\overline{pt(\sigma)}_0(P) = \mathtt{St}] \\
&\quad = \ \emptyset \\
&\quad = \ |\sigma|_0.
\end{aligned}$$

Assume now (7.16) holds for $n = k$. We first note that

$$\begin{aligned}
&pt(\sigma)(P^{pc}_{k+1}) \\
&\quad = \ pt(\sigma)(P \cup \{\langle t, \rho\rangle \mid t \in \rho(P^{pc}_k)\}) \qquad [\text{Definition } P^{pc}_{k+1}] \\
&\quad = \ \{s \in \mathtt{St} \mid (t \in \sigma(s) \ \Rightarrow \ t \in P) \ \& \ (\langle t, \tau\rangle \in \sigma(s) \ \Rightarrow \ t \in pt(\tau)(P^{pc}_k))\} \\
&\qquad\qquad [\text{Definition } pt] \\
&\quad = \ \{s \in \mathtt{St} \mid (t \in \sigma(s) \ \Rightarrow \ t \in P) \ \& \ (\langle t, \tau\rangle \in \sigma(s) \ \Rightarrow \ t \in \overline{pt(\tau)}_k(P))\} \\
&\qquad\qquad [\text{our convention: } \overline{pt(\tau)}_k = \lambda P \subseteq \mathtt{St}.pt(\tau)(P^{pc}_k)] \\
&\quad = \ \{s \in \mathtt{St} \mid (t \in \sigma(s) \ \Rightarrow \ t \in P) \ \& \\
&\qquad\qquad\qquad (\langle t, \tau\rangle \in \sigma(s) \ \Rightarrow \ \omega^{-1}(\overline{pt(\tau)}_k)(t) \subseteq P)\} \qquad [\text{Lemma 3.3.5}] \\
&\quad = \ \{s \in \mathtt{St} \mid (t \in \sigma(s) \ \Rightarrow \ t \in P) \ \& \ (\langle t, \tau\rangle \in \sigma(s) \ \Rightarrow \ |\tau|_k(t) \subseteq P)\} \\
&\qquad\qquad [\text{induction}] \\
&\quad = \ \{s \in \mathtt{St} \mid |\sigma|_{k+1} \subseteq P\} \qquad [\text{definition } |\sigma|_{k+1}] \\
&\quad = \ \omega(|\sigma|_{k+1})(P). \qquad [\text{definition } \omega]
\end{aligned}$$

Since ω and ω^{-1} form an isomorphism, we can conclude that Equation (7.16) holds also for $n = k + 1$. $\square$

We use Lemma 7.5.7 and Lemma 7.5.9 to obtain an Egli-Milner state transformer from $\sigma \in \Sigma_2$. We proceed as follows.

1. By the duality Theorem 7.4.7 we have that $pt(\sigma)$ is a predicate transformer in Π_2.

2. By the correctness Theorem 7.5.2, $\lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{tc})$ is a total correctness predicate transformer in $PT_T(\mathtt{St}, \mathtt{St})$, and $\lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{pc})$ is a partial correctness predicate transformer in $PT_P(\mathtt{St}, \mathtt{St})$.

3. By Theorem 7.5.3, the pair $\langle \lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{tc}), \lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{pc})\rangle$ is a Nelson predicate transformer in $PT_N(\mathtt{St}, \mathtt{St})$.

4. By the duality Theorem 3.3.14,
$$\begin{aligned} &\eta^{-1}(\langle \lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{tc}), \lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{pc})\rangle)(s) \\ &\quad = \omega^{-1}(\lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{pc}))(s) \cup \{\perp \mid s \notin pt(\sigma)(\mathtt{St}^{tc})\} \end{aligned}$$
is an Egli-Milner state transformer in $ST_E(\mathtt{St}, \mathtt{St})$.

5. By Lemma 7.5.7, $s \notin pt(\sigma)(\mathtt{St}^{tc})$ if and only if $\langle s, \sigma\rangle \Uparrow$.

6. By Lemma 7.5.9, $|\sigma|(s) = \omega^{-1}(\lambda P \subseteq \mathtt{St}.pt(\sigma)(P^{pc}))(s)$.

This yields an *abstraction function* $abs_E : \Sigma_2 \to ST_E(\mathtt{St}, \mathtt{St})$ by putting
$$abs_E(\sigma)(s) = |\sigma| \cup \{\perp \mid \langle s, \sigma\rangle \Uparrow\}$$
for $\sigma \in \Sigma_2$ and $s \in \mathtt{St}$. Moreover, using the function $E_H : ST_E(\mathtt{St}, \mathtt{St}) \to ST_H(\mathtt{St}, \mathtt{St})$ and $E_S : ST_E(\mathtt{St}, \mathtt{St}) \to ST_S(\mathtt{St}, \mathtt{St})$ defined in Section 3.2 we obtain two other abstraction functions
$$\begin{aligned} abs_H &= E_H \circ abs_E : \Sigma_2 \to ST_H(\mathtt{St}, \mathtt{St}) \\ abs_S &= E_S \circ abs_E : \Sigma_2 \to ST_S(\mathtt{St}, \mathtt{St}). \end{aligned}$$
Thus the three basic denotational models for sequential non-deterministic languages can be encoded into the forward metric semantics with resumptions.

7.5.10. Theorem. *Let $\sigma \in \Sigma_2$ and $P \subseteq \mathtt{St}$. Then*
(i) $\eta(abs_E(\sigma))(P) = \langle pt(\sigma)(P^{tc}), pt(\sigma)(P^{pc})\rangle$;
(ii) $\omega(abs_H(\sigma))(P) = pt(\sigma)(P^{pc})$;
(iii) $\omega(abs_S(\sigma))(P) = pt(\sigma)(P^{tc})$.

Proof: The first item follows immediately by definition of abs_E and because η^{-1} is the inverse of η. The other two items can be proved simultaneously as follows.
$$\begin{aligned} &\langle pt(\sigma)(P^{tc}), pt(\sigma)(P^{pc})\rangle \\ &\quad = \eta(abs_E(\sigma))(P) \quad \text{[by the above item (i)]} \\ &\quad = \langle \omega(E_S(abs_E(\sigma)))(P), \omega(E_H(abs_E(\sigma)))(P)\rangle \quad \text{[Equation (3.10)]} \end{aligned}$$

$$= \langle \omega(abs_S(\sigma))(P), \omega(abs_H(\sigma))(P)\rangle. \qquad \text{[Definition of } abs_S \text{ and } abs_H\text{]} \quad \Box$$

An immediate consequence of the above Theorem, Lemma 7.5.4, Lemma 7.4.8, and Lemma 3.3.7 is that all the three abstraction functions abs_H, abs_S and abs_E preserve both the union function '$\Box$' and the composition function ';'.

7.6 Temporal properties

We conclude this chapter by showing (without going into details) how (linear) temporal properties of programs can be treated within the metric predicate transformer semantics $Wp_2[\![-]\!]$. Branching temporal properties could be studied in a similar way.

Let $\mathtt{St}^\infty$ be the set of all finite and infinite sequence of states in $\mathtt{St}$. A sequence of states in $\mathtt{St}$ can be thought of as the juxtaposition of the states in which a computation may result when executing a program. A predicate $P \subseteq \mathtt{St}^\infty$ is said to be *linear time*. For example, if $P \subseteq \mathtt{St}$ then the predicate

$$always(P) \quad = \quad \{w \in \mathtt{St}^\infty \mid \forall u, v \in \mathtt{St}^\infty\, \forall s \in \mathtt{St}\colon w = usv \;\Rightarrow\; s \in P\}$$

is linear time. Informally, a program S satisfies the linear predicate $always(P)$ if the predicate P holds in any state of any computation of S.

For every linear time predicate $P \subseteq \mathtt{St}^\infty$, define the *truncated predicate* $trunc(P)$ and the *first state predicate* $first(P)$ respectively by

$$\begin{aligned} trunc(P) \quad &= \quad \{w \in \mathtt{St}^\infty \mid \exists s \in \mathtt{St}\colon sw \in P\} \\ first(P) \quad &= \quad \{s \in \mathtt{St} \mid \exists w \in \mathtt{St}^\infty\colon sw \in P\}. \end{aligned}$$

Informally, if a linear predicate P holds for a computation then $trunc(P)$ holds for the rest of the computation after its first atomic step, and $first(P)$ holds for the first state of the computation. The truncated predicate and the first state predicate are used for defining the linear enhancement of $P \subseteq \mathtt{St}^\infty$ as the least subset $P^{lin} \subseteq \mathtt{St} + \mathtt{St} \times \Pi_2$ such that

$$P^{lin} \quad = \quad first(P) \cup \{\langle s, \rho\rangle \mid s \in first(P) \;\&\; s \in \rho(trunc(P)^{lin})\}. \tag{7.17}$$

Hence P^{lin} consists of all strings of length one of P and of all those pairs $\langle s, \rho\rangle$ for which s is appropriate in the sense that it satisfies the first state of the predicate P and it will lead to $trunc(P)^{lin}$ following ρ. By taking the greatest solution of the above equation we obtain both a weakest and a weakest liberal linear semantics in the style of Lukkien [Luk91, Luk94].

For example consider the program $\langle d, v_1 := 1 \,;\, v_2 := 2\rangle$ in $\mathcal{L}_2$, and let $\pi_1 = Wp_2[\![\langle d, v_1{:=}1\rangle]\!]$ and $\pi_2 = Wp_2[\![\langle d, v_2{:=}2\rangle]\!]$. We want to calculate the semantics of the above program for the linear enhancement P^{lin} of the linear time predicate

$$P \quad = \quad \{s_1 s_2 \in \mathtt{St}^\infty \mid s_1(v_1) = s_2(v_1) = 1 \;\&\; s_2(v_2) = 2\}.$$

Intuitively P says that after the first atomic step the value of v_1 is 1, and immediately after the value of v_2 is 2 while the value of v_1 remains the same. We have

$$\begin{aligned}
& Wp_2[\![\langle d, v_1 := 1 \,;\, v_2 := 2\rangle]\!](P^{lin}) \\
&= (\pi_1 \,;\, \pi_2)(P^{lin}) \\
&= \pi_1(\{s \mid \langle s, \pi_2\rangle \in P^{lin}\} \cup \{\langle s, \rho\rangle \mid \langle s, \rho \,;\, \pi_2\rangle \in P^{lin}\}) \\
&= \{s \mid \langle s[1/v_1], \pi_2\rangle \in P^{lin}\} \\
&= \{s \mid s[1/v_1] \in \mathit{first}(P) \;\&\; s[1/v_1] \in \pi_2(\mathit{trunc}(P)^{lin})\} \\
&= \{s \mid s[1/v_1](v_1) = 1 \;\&\; s[1/v_1][2/v_2] \in \mathit{trunc}(P)^{lin}\} \\
&= \{s \mid s[1/v_1](v_1) = 1 \;\&\; s[1/v_1][2/v_2](v_1) = 1 \;\&\; s[1/v_1][2/v_2](v_2) = 2\} \\
&= \mathtt{St}
\end{aligned}$$

which is indeed the result to be expected. From the above it follows that if we take P to be the linear time predicate

$$\{s_1 w \in \mathtt{St}^\infty \mid s_1(v_1) \neq 1 \;\&\; w \in \mathtt{St}^\infty\}$$

then $Wp_2[\![\langle d, v_1 := 1 \,;\, v_2 := 2\rangle]\!](P^{lin}) = \emptyset$.

Returning to the predicate $\mathit{always}(P)$ for $P \subseteq \mathtt{St}$, it is immediate to see that $\mathit{trunc}(\mathit{always}(P)) = \mathit{always}(P)$ and $\mathit{first}(\mathit{always}(P)) = P$. Hence the linear extension of $\mathit{always}(P)$ according to the Equation (7.17) is the least subset P^{alw} of $\mathtt{St} + \mathtt{St} \times \Pi_2$ such that

$$P^{alw} = P \cup \{\langle s, \rho\rangle \mid s \in P \;\&\; s \in \rho(P^{alw})\}. \tag{7.18}$$

As for P^{tc}, we can give a characterization of P^{alw} in terms of its approximants. Define, for $n \geq 0$, P^{alw}_n inductively by

$$\begin{aligned}
P^{alw}_0 &= \emptyset \text{ and} \\
P^{alw}_{n+1} &= P \cup \{\langle s, \rho\rangle \mid s \in P \;\&\; s \in \rho(P^{alw}_n)\}.
\end{aligned}$$

Using a proof similar to that of Lemma 7.3.14 we can show that for all $n \geq 0$, P^{alw}_n is open in the metric topology of $\mathtt{St} + \mathtt{St} \times \frac{1}{2} \cdot \Pi_2$. From this fact it follows immediately that

$$P^{alw} = \bigcup_n P^{alw}_n.$$

Informally, this means that $\pi(P^{alw})$ holds exactly for those input states s such that all computations of the program denoted by π started in s terminate and every state that is reached satisfies the predicate P. By taking the greatest solution of the Equation (7.18) we obtain a more liberal version in which termination is not required.

7.7 Concluding notes

The language we considered in this chapter assumes a global shared state for all parallel components, and it does not have a synchronization operator. Hence the domain of 'sequences of pairs' of [BHR94] could have been used in order to obtain a fully abstract model. We opted for a branching domain because it gives a finer equivalence on processes and supports both linear and branching time properties. In fact the domain Π_2 is internally fully abstract with respect to bisimulation [BKV95], that is, two predicate transformers $\pi, \rho \in \Pi_2$ are equal if and only if the computations they denote are bisimilar.

Synchronization by shared variables can be implemented in our language using, for example, semaphores [Dij68]. This requires a simple form of atomization for sequential and non-deterministic statements [BKV95].

In the last section we focused on two classes of properties of programs, namely classes of properties based on partial and total correctness. In the area of parallel programming many other properties are of importance as well. The metric predicate transformer domain Π_2 supports reasoning about both linear and branching time predicates. We briefly studied the linear time predicate 'always P' (during an execution the predicate P always holds). It would be interesting to investigate other more specific linear predicates like 'eventually P' (during an execution the predicate P will holds), or 'P leads-to Q' (during an execution if P holds then at some point later Q will hold').

Part III

A logical perspective

Chapter 8

Topological spaces and observation frames

In this last part of the thesis we make an abstraction step towards an axiomatic approach to the semantics and the specification of programming languages. We consider predicates as elements of an abstract algebra. For example, we may think of an abstract algebra as stemming from a logical system: elements correspond to equivalence classes of formulae which are provably equivalent, and the order corresponds to the entailment relation.

In order to show that the axioms of the class of algebras we consider capture exactly the collection of predicates we have in mind, a representation theorem is necessary. A representation theorem is a correspondence between an abstract algebra and its set-theoretical model. The first representation theorem is due to Cayley [Ca878] showing that every abstract group is isomorphic to a concrete group of permutations. A representation theorem for the algebra of all predicates was first proved by Lindenbaum and Tarski [Tar35]. They proved that a Boolean algebra is isomorphic to the collection of all subsets of some set if and only if it is complete and atomic. This general result restricts the class of Boolean algebras for which a concrete representation exists. It was Stone [Sto36] who first saw a connection between algebra and topology. He constructed from a Boolean algebra a set of points using prime ideals which can be made into a topological space in a natural way. Conversely, using a topology on a set of points he was able to construct a Boolean algebra. For certain topological spaces (later called Stone spaces) these constructions give an isomorphism. In a later paper [Sto37], Stone generalized this correspondence from Stone spaces to spectral spaces and from Boolean algebras to distributive lattices. Hofmann and Keimel [HK72] described the Stone representation theorem in a categorical framework showing a duality between the category of Boolean algebras and a sub-category of topological spaces. A representation theorem for Boolean algebras with operators has been considered by Jónsson and Tarski [JT51, JT52]. By means of an extension theorem they proved that operators on a Boolean algebra can be naturally extended to completely additive operators on a complete and atomic Boolean algebra.

Stone's representation theorem leaves open the problem of finding an abstract characterization of topological spaces. For every topological space, its lattice of open sets forms a frame. This fact leads Papert and Papert [PP58] to a representation theorem between spatial frames and sober spaces. Even further, Isbell [Isb72a] gives an adjunction between the category of topological spaces with continuous functions and the opposite category of frames with frame homomorphisms. This adjunction yields a duality between the category of sober spaces and the category of spatial frames.

The importance of Stone-like dualities in a mathematical context is shown in a book of Johnstone [Joh82a], and in the context of domain theory in [GHK$^+$80]. Abramsky [Abr87] applied Stone duality to get logics of domains, as used in denotational semantics. He argues that Stone duality is the bridge between denotational and axiomatic semantics.

In this chapter we consider a topological space as a function from its frame of open sets to its completely distributive lattice of saturated sets. More abstractly this structure is an observation frame. In the light of our discussion in Chapter 5, an observation frame is a map from affirmative predicates to specifiable predicates preserving the geometric logic of the affirmative predicates. We construct topological spaces from observation frames by taking as points special kinds of prime elements. In this way we obtain a duality between observation frames and $\mathcal{T}_0$ spaces. We also give a logic of observation frames with arbitrary conjunctions and disjunctions. This is done by the introduction of M-topological systems, which are a generalization of the topological systems of Vickers [Vic89]. Finally we consider some examples of interesting $\mathcal{T}_0$ spaces which need not to be sober.

8.1 Observation frames

Complete lattices are closed under infinite meets and infinite joins. However these operations do not need to represent infinite conjunctions and infinite disjunctions even if the elements of the lattice are subsets of some set. For example, the lattice of open sets of a topological space X is complete since an arbitrary union of open sets is open, but the meet of an arbitrary collection of open sets S is given by the interior of the intersection of S, which does not, in general, coincide with the intersection of S.

In this section we introduce a mathematical structure which represents abstract topological spaces and which supports both the arbitrary conjunctions and the arbitrary disjunctions of the abstract open sets.

Our starting point is the fact that the lattice of open sets of a topological space X can be embedded in the lattice of saturated sets of X. The saturated sets are closed under arbitrary unions and intersections. Since every open set is saturated, the logic of the affirmative predicates is preserved.

8.1.1. Definition. An *observation frame* is a frame morphism $\alpha : F \to L$ from a frame F to a completely distributive lattice L such that every element q in L is saturated , that is,

$$q \quad = \quad \bigsqcap\{\alpha(x) \mid q \sqsubseteq \alpha(x)\}.$$

For clarity we adopt the following convention for an observation frame $\alpha : F \to L$. The order, the meet and the join in F are denoted by $(\leq, \bigwedge, \bigvee)$, respectively, while the order, the meet and the join in L are denoted by $(\sqsubseteq, \bigsqcap, \bigsqcup)$. In case F (or L) is a subset of $\mathcal{P}(X)$ for some set X we use the standard $(\subseteq, \bigcap, \bigcup)$ whenever these coincide with the order, the meet or the join in F (or L)

The identity function $id_L : L \to L$, for L a completely distributive lattice, is an observation frame. If L is the two point completely distributive lattice $2 = \{\bot, \top\}$ with $\bot \sqsubseteq \top$, we refer to the observation frame $id_L : L \to L$ simply by **2**. Another example of an observation frame is given by the inclusion map $\mathcal{O}(X) \hookrightarrow \mathcal{Q}(X)$, where X is a topological space, $\mathcal{O}(X)$ is the frame of opens and $\mathcal{Q}(X)$ is the completely distributive lattice of the saturated sets. We denote this observation frame by $\Omega(X)$.

Next we organize observation frames into a category. We need an appropriate notion of morphism for observation frames. It can be obtained by adapting Definition 6.2.1.

8.1.2. Definition. A *morphism* from an observation frame $\alpha : F \to L$ to an observation frame $\beta : G \to K$ consists of a frame morphism $\phi : F \to G$ which is M-multiplicative, that is, for all subsets S and T of F

$$\bigsqcap \alpha(S) \sqsubseteq \bigsqcap \alpha(T) \quad \text{implies} \quad \bigsqcap \beta(\phi(S)) \sqsubseteq \bigsqcap \beta(\phi(T)).$$

This gives a category (with composition as for ordinary functions) which is denoted by **OFrm**.

The idea is that a morphism between observation frames preserves the logic of affirmative predicates, but also takes into account what happens to infinite conjunctions of these affirmative predicates (which are usually outside the frame). The above definition of morphisms between observation frames can be justified by the following example:

Assume that X and Y are two topological spaces and let $f : X \to Y$ be a continuous function (i.e. a map in the category of topological spaces **Sp**). The function f induces a morphism $\Omega(f) : \Omega(Y) \to \Omega(X)$ in **OFrm** defined by its inverse image, i.e., $\Omega(f)(o) = f^{-1}(o) = \{x \in X \mid f(x) \in o\}$ for every

$o \in \mathcal{O}(Y)$. We check the multiplicativity condition. Assume $S, T \subseteq \mathcal{O}(Y)$ with $\bigcap S \subseteq \bigcap T$. Then

$$\begin{array}{rcl} x \in \bigcap \Omega(f)(S) & \Leftrightarrow & \forall o \in S\colon f(x) \in o \\ & \Leftrightarrow & f(x) \in \bigcap S \\ & \Rightarrow & f(x) \in \bigcap T \quad [\bigcap S \subseteq \bigcap T] \\ & \Leftrightarrow & x \in \bigcap \Omega(f)(T). \end{array}$$

Thus we have a functor $\Omega : \mathbf{Sp} \to \mathbf{OFrm}^{op}$. Later it will be shown that Ω has a right adjoint.

The next theorem gives a mathematical justification of the definition of a morphism between observation frames. First we need the following preparatory lemma.

8.1.3. Lemma. *Let $\alpha : F \to L$ be an observation frame, $\beta : G \to H$ be a frame morphism from a frame G to a completely distributive lattice H, and $\phi\colon F \to G$ be a frame morphism. If $\psi\colon L \to H$ is a function preserving arbitrary intersections and such that $\beta \circ \phi = \psi \circ \alpha$, then ψ preserves also arbitrary joins (and hence is a morphism between completely distributive lattices).*

Proof: Let S be a subset of L and consider the following set of sets

$$\mathcal{A} = \{\{\alpha(a) \mid a \in F \ \&\ q \sqsubseteq \alpha(a)\} \mid q \in S\}.$$

Because $\alpha : F \to L$ is an observation frame $q = \bigsqcap\{a \in F \mid q \sqsubseteq \alpha(a)\}$ for all $q \in S$. Hence we have that

$$\begin{array}{rcll} \psi(\bigsqcup S) & = & \psi(\bigsqcup\{\bigsqcap A \mid A \in \mathcal{A}\}) & \\ & = & \psi(\bigsqcap\{\bigsqcup f(\mathcal{A}) \mid f \in \Phi(\mathcal{A})\}) & \text{[complete distributivity]} \\ & = & \bigsqcap\{\psi(\bigsqcup f(\mathcal{A})) \mid f \in \Phi(\mathcal{A})\} & [\psi \text{ preserves arbitrary meets}] \\ & = & \bigsqcap\{\bigsqcup\{\psi(\alpha(a)) \mid \alpha(a) \in f(\mathcal{A})\} \mid f \in \Phi(\mathcal{A})\}, & \end{array} \quad (8.1)$$

where the latter equality holds because for every $f \in \Phi(\mathcal{A})$, $f(\mathcal{A})$ is a subset of the image under α of F (since $f : \mathcal{A} \to \bigcup \mathcal{A}$) and because the commutativity $\beta \circ \phi = \psi \circ \alpha$ implies that ψ preserves all joins $\bigsqcup \alpha(T)$ in L for every $T \subseteq F$.

Consider now the following set of sets

$$\mathcal{B} = \{\{\psi(\alpha(a)) \mid a \in F \ \&\ q \sqsubseteq \alpha(a)\} \mid q \in S\}.$$

We have

$$\begin{array}{rcll} \bigsqcup \psi(S) & = & \bigsqcup\{\psi(\bigsqcap\{\alpha(a) \mid a \in F \ \&\ q \sqsubseteq \alpha(a)\}) \mid q \in S\} & \\ & = & \bigsqcup\{\bigsqcap\{\psi(\alpha(a)) \mid a \in F \ \&\ q \sqsubseteq \alpha(a)\} \mid q \in S\} & \\ & = & \bigsqcup\{\bigsqcap B \mid B \in \mathcal{B}\} & \\ & = & \bigsqcap\{\bigsqcup g(\mathcal{B}) \mid g \in \Phi(\mathcal{B})\}. & \text{[complete distributivity]} \end{array} \quad (8.2)$$

But for every $g \in \Phi(\mathcal{B})$ there exists $f \in \Phi(\mathcal{A})$ such that $f(q) = \alpha(a)$ if $g(q) = \psi(\alpha(a))$. Hence

$$\{\bigsqcup g(\mathcal{B}) \mid g \in \Phi(\mathcal{B})\} \subseteq \bigsqcup\{\psi(\alpha(a)) \mid \alpha(a) \in f(\mathcal{A})\} \mid f \in \Phi(\mathcal{A})\}$$

from which it follows, using the equalities (8.1) and (8.2), that

$$\psi(\bigsqcup S) \sqsubseteq \bigsqcup \psi(S).$$

The converse of the above inequation holds by monotonicity of ψ. $\square$

A frame morphism is a morphism between observation frames if and only if it can be uniquely extended to a morphism between completely distributive lattices. This result is similar to the extension theorem of Jónsson and Tarski [JT51] for Boolean algebras.

8.1.4. Theorem. *Let $\alpha : F \to L$ and $\beta : G \to H$ be two observation frames and $\phi : F \to G$ be a frame morphism. Then ϕ is a morphism in* **OFrm** *if and only if there exists a unique morphism $\widetilde{\phi} : L \to H$ in* **CDL** *such that $\beta \circ \phi = \widetilde{\phi} \circ \alpha$.*

$$\begin{tikzcd} F \arrow[r, "\alpha"] \arrow[d, "\phi"'] \arrow[dr, phantom, "*"] & L \arrow[d, "\widetilde{\phi}"] \\ G \arrow[r, "\beta"'] & H \end{tikzcd} \qquad \begin{tikzcd} L \arrow[d, dashed, "\widetilde{\phi}"] \\ H \end{tikzcd}$$

Proof: Assume $\widetilde{\phi} : L \to H$ is a morphism between completely distributive lattices such that $\beta \circ \phi = \widetilde{\phi} \circ \alpha$. We need to prove that $\phi : F \to G$ is M-multiplicative. Let $S, T \subseteq F$ be such that $\bigsqcap \alpha(S) \sqsubseteq \bigsqcap \alpha(T)$. Then we have:

$$\begin{array}{rcll} \bigsqcap \beta(\phi(S)) & = & \bigsqcap \widetilde{\phi}(\alpha(S)) & [\beta \circ \phi = \widetilde{\phi} \circ \alpha] \\ & = & \widetilde{\phi}(\bigsqcap \alpha(S)) & [\widetilde{\phi} \text{ is meet preserving}] \\ & \sqsubseteq & \widetilde{\phi}(\bigsqcap \alpha(T)) & [\text{monotonicity of } \widetilde{\phi} \text{ and } \bigsqcap \alpha(S) \sqsubseteq \bigsqcap \alpha(T)] \\ & = & \bigsqcap \widetilde{\phi}(\alpha(T)) & [\widetilde{\phi} \text{ is meet preserving}] \\ & = & \bigsqcap \beta(\phi(T)). & [\beta \circ \phi = \widetilde{\phi} \circ \alpha] \end{array}$$

Conversely, assume ϕ is an observation frame morphism and define, for $S \subseteq F$,

$$\widetilde{\phi}(\bigsqcap \alpha(S)) = \bigsqcap \beta(\phi(S)).$$

It is well-defined because ϕ is M-multiplicative: if $\bigsqcap \alpha(S) = \bigsqcap \alpha(T)$ for subsets S and T of F then, $\bigsqcap \beta(\phi(S)) = \bigsqcap \beta(\phi(T))$. By definition, $\widetilde{\phi}$ preserves arbitrary meets and it is the unique function preserving arbitrary meets such that $\beta \circ \phi = \widetilde{\phi} \circ \alpha$. By Lemma 8.1.3 it follows that $\widetilde{\phi}$ preserves also arbitrary joins. $\square$

M-filters and M-prime elements

In this subsection we introduce the notions of M-filter and of M-prime element of an observation frame. They will be used later to construct the points of a topological space associated with an observation frame. Furthermore we prove that completely prime M-filters and the M-prime elements of an observation frame $\alpha : L \to F$ and morphisms from α to $\mathbf{2}$ in **OFrm** are essentially the same. In the next chapter we will state the precise relationships to the standard notions of filter and prime elements of a frame.

8.1.5. Definition. Let $\alpha : F \to L$ be an observation frame. A subset $\mathcal{U}$ of F is an *M-filter* of α if for all $a \in F$,

$$\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \alpha(a) \;\Rightarrow\; a \in \mathcal{U}.$$

An M-filter $\mathcal{U}$ of α is called *completely prime* if for every $S \subseteq F$,

$$\bigvee S \in \mathcal{U} \;\Rightarrow\; \exists s \in S : s \in \mathcal{U}.$$

Notice that a completely prime M-filter $\mathcal{U}$ cannot contain $\perp_F$ because $\perp_F = \bigvee \emptyset$ and hence by the definition above there should be $s \in \emptyset$ such that $s \in \mathcal{U}$.

The condition defining an M-filter can be described as closure under arbitrary meets as distinguished from the closure under finite meets of a filter. It is easy to see that every M-filter $\mathcal{U}$ of an observation frame $\alpha : F \to L$ is a filter of F. Indeed, it is non-empty because $\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \top_L = \alpha(\top_F)$ implies $\top_F \in \mathcal{U}$. It is an upper closed subset of F because if $a \in \mathcal{U}$ and $a \leq b$ then, by monotonicity of α, $\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \alpha(a) \sqsubseteq \alpha(b)$. Hence $b \in \mathcal{U}$. Finally, suppose a and $b \in \mathcal{U}$. Then $\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \alpha(a)$ and $\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \alpha(b)$. Hence $\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \alpha(a) \sqcap \alpha(b) = \alpha(a \wedge b)$, which implies $a \wedge b \in \mathcal{U}$.

The converse is in general not true. This can be shown using a counter-example due to Chellas [Che80] in the context of minimal augmented models for modal logics. Consider the observation frame $id_{\mathcal{P}(\mathbb{R})} : \mathcal{P}(\mathbb{R}) \to \mathcal{P}(\mathbb{R})$, where $\mathbb{R}$ is the set of real numbers ordered as usual. For $r \in \mathbb{R}$ define

$$\mathcal{F}_r = \{ V \subseteq \mathbb{R} \mid \exists s \in \mathbb{R} : r < s \;\&\; (r, s) \subseteq V \}$$

where $(r, s) = \{t \in \mathbb{R} \mid r < t < s\}$. Hence $\mathcal{F}_r$ is the set of all sets of real numbers that include some open interval (r, s) for some $r < s$. It is an easy verification to prove that $\mathcal{F}_r$ is a filter of $\mathcal{P}(\mathbb{R})$. However $\mathcal{F}_r$ is not an M-filter of the observation frame $id_{\mathcal{P}(\mathbb{R})}$ because $\bigcap \mathcal{F}_r$ is the empty set since there is no smallest open interval (r, s) for $r < s$, whereas $\emptyset \notin \mathcal{F}_r$ because (r, s) is not empty for every $r < s$. Therefore $\bigcap \mathcal{F}_r = \emptyset$ but $\emptyset \notin \mathcal{F}_r$. We will return to the relation between filters and M-filters in the next chapter.

8.1.6. Lemma. *Let $\alpha : F \to L$ be an observation frame. The assignment $\mathcal{U} \mapsto {\uparrow}(\bigsqcap \alpha(\mathcal{U}))$ is an isomorphism between M-filters of α and principal filters of L, where ${\uparrow}(\cdot)$ denotes the upper closure with respect to the order of L.*

Proof: The above map is well-defined because, by definition, $\uparrow(\bigsqcap \alpha(\mathcal{U}))$ is a principal filter of L. Assume $\mathcal{U}_1$ and $\mathcal{U}_2$ are two different M-filters of α, say there exists $a \in \mathcal{U}_1$ but $a \notin \mathcal{U}_2$. Then $\alpha(a) \in \uparrow(\bigsqcap \alpha(\mathcal{U}_1))$ but $\bigsqcap \alpha(\mathcal{U}_2) \not\sqsubseteq \alpha(a)$ because otherwise $a \in \mathcal{U}_2$ since $\mathcal{U}_2$ is an M-filter. Hence the above map is injective. Next we show it is also onto: let V be a principal filter of L and consider the set $\mathcal{U} = \{a \in F \mid \alpha(a) \in V\}$. For every $a \in F$, if $\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \alpha(a)$ then $\bigsqcap V \sqsubseteq \alpha(a)$ because $\alpha(\mathcal{U}) = V \cap \alpha(F)$. Since V is a principal filter of L, $\alpha(a) \in V$ and hence $a \in \mathcal{U}$. Therefore $\mathcal{U}$ is a M-filter of α. It remains to prove $V = \uparrow(\bigsqcap \alpha(\mathcal{U}))$. The inclusion from right to left follows because $\alpha(\mathcal{U}) = V \cap \alpha(F)$ and $V = \uparrow \bigsqcap V$, being V a principal filter. Conversely, let $q \in V$. Then $\alpha(a) \in V$ for all $a \in F$ such that $q \sqsubseteq \alpha(a)$ because V is upper closed. Hence $\alpha(a) \in V \cap \alpha(F) = \alpha(\mathcal{U})$, which implies

$$\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \bigsqcap \{\alpha(a) \mid a \in F \;\&\; q \sqsubseteq \alpha(a)\} = q$$

where the latter equality follows because α is an observation frame. Therefore $q \in \uparrow(\bigsqcap \alpha(\mathcal{U}))$. $\square$

In other words, the above lemma says that M-filters of $\alpha: F \to L$ are in bijective correspondence with elements of L. As an immediate consequence we have, for every M-filter $\mathcal{U}$ of $\alpha : F \to L$,

$$\alpha(\mathcal{U}) = \uparrow(\bigsqcap \alpha(\mathcal{U})) \cap \alpha(F).$$

Let X be a topological space and consider the observation frame $\Omega(X)$. Every saturated set $q \in \mathcal{Q}(X)$ induces an M-filter

$$\mathcal{U}(q) = \{o \in \mathcal{O}(X) \mid q \subseteq o\}.$$

Indeed, if $\bigcap \mathcal{U}(q) \subseteq o$ then $q \subseteq o$ because $q = \bigcap \mathcal{U}(q)$. Therefore $o \in \mathcal{U}(q)$.

Also, for every $x \in X$ the set

$$\mathcal{U}_0(x) = \{o \in \mathcal{O}(X) \mid x \in o\}$$

is clearly an M-filter of $\Omega(X)$. Furthermore, $\mathcal{U}_0(x)$ is completely prime since for every $S \subseteq \mathcal{O}(X)$ and $o \in \mathcal{U}_0(x)$ such that $o \subseteq \bigcup S$ we have $x \in o \subseteq \bigcup S$. Hence there exists $s \in S$ such that $x \in s$. Therefore $s \in \mathcal{U}_0(x)$.

8.1.7. Definition. For an observation frame $\alpha : F \to L$, an element $p \in F$ is called *M-prime* if for all $S \subseteq F$,

$$\bigsqcap \alpha(S) \sqsubseteq \alpha(p) \;\Rightarrow\; \exists s \in S : s \leq p.$$

The set of all M-prime elements of α is denoted by $MP(\alpha)$.

Consider the observation frame $\Omega(X)$ of a topological space X. Define for every $x \in X$ the open set

$$o_x = int(X \setminus \{x\}) = \bigcup\{o \in \mathcal{O}(X) \mid x \notin o\} \subseteq X \setminus \{x\},$$

where $int(\cdot)$ is the *interior operator* associated with the topology on X. By definition o_x is the greatest (with respect to subset inclusion) open set not containing x, that is, for an open o', $x \notin o'$ if and only if $o' \subseteq o_x$. It is also an M-prime element. Indeed, for every subset S of $\mathcal{O}(X)$ if $\bigcap S \subseteq o_x$ then $x \notin \bigcap S$ because otherwise we would have $x \in \bigcap S \subseteq o_x$ contradicting $x \notin o_x$. But then there exists an $s \in S$ such that $x \notin s$. Since o_x is the greatest open set not containing x, $s \subseteq o_x$. Thus $o_x \in MP(\Omega(X))$ for every $x \in X$.

Notice that for every $o \in \mathcal{O}(X)$ we have $\bigcap\{o_x \mid x \notin o\} = o$:

($\subseteq$) If $y \in \bigcap\{o_x \mid x \notin o\}$ then $y \in o$ because otherwise $y \in X \setminus o$ and hence $o_y \in \{o_x \mid x \notin o\}$. But this yields $y \in \bigcap\{o_x \mid x \notin o\} \subseteq o_y$, a contradiction.

($\supseteq$) For every $x \in X \setminus o$, $o \subseteq (X \setminus \{x\})$. Hence by idempotency and monotonicity of the interior operator we obtain $o = int(o) \subseteq int(X \setminus \{x\}) = o_x$ for every $x \notin o$. Therefore $o \subseteq \bigcap\{o_x \mid x \notin o\}$.

(For the case when $o = X$ observe that $\{o_x \mid x \notin o\} = \emptyset$ and that $\bigcap \emptyset = X = o$.) Next we show that every M-prime element in $\Omega(X)$ is of the form o_x for some $x \in X$. Indeed, let p be an M-prime element of $\Omega(X)$. Since $p \in \mathcal{O}(X)$ we have just seen that $\bigcap\{o_x \mid x \notin p\} \subseteq p$. But then $o_x \subseteq p$ for some $x \notin p$. The latter yields $p \subseteq o_x$ and hence $p = o_x$. This fact will be crucial later on for obtaining our duality. If X is a $\mathcal{T}_0$ space then clearly every M-prime element of $\Omega(X)$ is of the form o_x for a unique $x \in X$.

The next lemma is the main result of this subsection. It gives isomorphisms between M-filters, M-prime elements of an observation frame $\alpha : F \to L$ and also morphisms from α to $\mathbf{2}$ in **OFRm**.

8.1.8. Lemma. *For an observation frame $\alpha : F \to L$ there are bijective correspondences between*

(i) *morphisms ϕ from α to* $\mathbf{2}$ *in* **OFrm***,*
(ii) *completely prime M-filters $\mathcal{U}$ of α,*
(iii) *M-prime elements p of α.*

The correspondences are given by:

$$
\begin{array}{lrcl}
\text{(i)} \Rightarrow \text{(ii)} & \phi \mapsto \mathcal{U}_\phi & = & \{a \in F \mid \phi(a) = \top\}; \\
\text{(ii)} \Rightarrow \text{(i)} & \mathcal{U} \mapsto \phi_\mathcal{U} & = & \lambda a \in F. \begin{cases} \top & \textit{if } a \in \mathcal{U} \\ \bot & \textit{otherwise;} \end{cases} \\
\text{(ii)} \Rightarrow \text{(iii)} & \mathcal{U} \mapsto p_\mathcal{U} & = & \bigvee\{a \in F \mid a \notin \mathcal{U}\};
\end{array}
$$

$$
\begin{array}{llcl}
\text{(iii)} \Rightarrow \text{(ii)} & p \mapsto \mathcal{U}_p & = & F \setminus (\downarrow p); \\
\text{(iii)} \Rightarrow \text{(i)} & p \mapsto \phi_p & = & \lambda a \in F. \begin{cases} \bot & \textit{if } a \leq p \\ \top & \textit{otherwise;} \end{cases} \\
\text{(i)} \Rightarrow \text{(iii)} & \phi \mapsto p_\phi & = & \bigvee\{a \in F \mid \phi(a) = \bot\}.
\end{array}
$$

Proof: Let $\alpha : F \to L$ be an observation frame, $\phi : \alpha \to \mathbf{2}$ be a morphism in **OFrm**, $\mathcal{U} \subseteq F$ be a completely prime M-filter and $p \in F$ be an M-prime element. We prove only (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (i). The verification of the other correspondences is left to the reader.

- (i) $\Rightarrow$ (ii): We have to prove that $\mathcal{U}_\phi$ is a completely prime M-filter. We start by proving that $\mathcal{U}_\phi$ is an M-filter. For every $x \in F$ such that $\bigsqcap \alpha(\mathcal{U}_\phi) \sqsubseteq \alpha(x)$ we have $\bigwedge\{\phi(a) \mid a \in \mathcal{U}_\phi\} \leq \phi(x)$ since ϕ is a morphism in **OFrm**. But $a \in \mathcal{U}_\phi$ if and only if $\phi(a) = \top$ by definition, hence also $\phi(x) = \top$. Therefore x is in $\mathcal{U}_\phi$.

  It remains to show that $\mathcal{U}_\phi$ is completely prime. Let $S \subseteq F$ and $a \in \mathcal{U}_\phi$ be such that $a \leq \bigvee S$. Then $\phi(\bigvee S) = \top$ because ϕ is a frame morphism and $\top = \phi(a) \leq \phi(\bigvee S) = \bigvee \phi(S)$. Therefore there is $s \in S$ such that $\phi(s) = \top$, that is, there is a $s \in S$ such that $s \in \mathcal{U}_\phi$.

- (ii) $\Rightarrow$ (iii): We have to prove that $p_\mathcal{U}$ is an M-prime element. Let $S \subseteq F$ be such that $\bigsqcap \alpha(S) \sqsubseteq \alpha(p_\mathcal{U})$. Then $\bigsqcap \alpha(S) \sqsubseteq \alpha(\bigvee\{a \in F \mid a \notin \mathcal{U}\}) = \bigsqcup\{\alpha(a) \mid a \notin \mathcal{U}\}$. There must exist $s \in S$ such that $s \notin \mathcal{U}$ because if not, $S \subseteq \mathcal{U}$ would imply $\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \bigsqcap \alpha(S) \sqsubseteq \alpha(\bigvee\{a \in F \mid a \notin \mathcal{U}\})$ and hence $\bigvee\{a \in F \mid a \notin \mathcal{U}\} \in \mathcal{U}$ as $\mathcal{U}$ is an M-filter. Since it is also completely prime we have the contradiction that there exists $a \notin \mathcal{U}$ such that $a \in \mathcal{U}$.

- (iii) $\Rightarrow$ (i): We have to prove that ϕ_p is a morphism in **OFrm**. It is easily verified that it is a frame morphism from F to 2. Hence we concentrate on the proof that ϕ_p is M-multiplicative. Let $S, T \subseteq F$ be such that $\bigsqcap \alpha(S) \sqsubseteq \bigsqcap \alpha(T)$. Assume $\bigwedge \phi_p(S) = \top$ but suppose $\bigwedge \phi_p(T) = \bot$. Then there exists $t \in T$ such that $\phi_p(t) = \bot$ and hence $t \leq p$. Since p is an M-prime element, we have that $\bigsqcap \alpha(S) \sqsubseteq \bigsqcap \alpha(T) \sqsubseteq \alpha(t) \sqsubseteq \alpha(p)$ implies there exists $s \in S$ such that $s \leq p$. Hence $\phi_p(s) = \bot$ contradicting $\bigwedge \phi_p(S) = \top$. □

Completely prime M-filters are preserved by the inverse image of a morphism in **OFrm**: let $\phi\colon(\alpha\colon F \to L) \to (\beta\colon G \to H)$ be a morphism in **OFrm** and let $\mathcal{U}$ be a completely prime M-filter of β. Then $\phi_\mathcal{U}\colon\beta \to \mathbf{2}$ is also a morphism in **OFrm** which hence yields by composition a morphism from α to $\mathbf{2}$, or, equivalently, a completely prime M-filter $\phi^{-1}(\mathcal{U})$ of α.

We conclude this subsection by taking a closer look at Galois connections between posets. Galois connections play an important role in spectral theory (see for example [GHK+80]) and in general in lattice theory. In particular we are interested in those posets which constitute the frame part of an observation frame.

8.1.9. Definition. Let F and G be two posets and $f : F \to G$, $g : G \to F$ be two functions. We say the pair (f, g) is a *Galois connection* between F and G if both f and g are monotone, and, for all $x \in F$ and $y \in G$,

$$f(x) \leq y \quad \Leftrightarrow \quad x \leq g(y).$$

For a Galois connection (f, g) the function g is called *upper (or left) adjoint* and the function f is called *lower (or right) adjoint*. A Galois connection is a special case of adjoint functors, where the posets F and G are seen as categories [ML71, Chapter IV]. Any upper adjoint g preserves all meets in G, while any lower adjoint f preserves all joins in F. More generally we have the following characterization of Galois connections.

8.1.10. Lemma. *Let F, G be two complete lattices.*

(i) *A function $g{:}G \to F$ preserves all meets in G if and only if g is monotone and has a lower adjoint $f : F \to G$ given by $f(x) = \bigwedge\{y \in G \mid x \leq g(y)\}$.*

(ii) *A function $f{:}F \to G$ preserves all joins in F if and only if f is monotone and has an upper adjoint $g : G \to F$ given by $g(y) = \bigvee\{x \in F \mid f(x) \leq y\}$.*

(iii) *A pair of monotone functions (f, g) with $f : F \to G$ and $g : G \to F$ is a Galois connection if and only if $f(g(y)) \leq y$ and $x \leq g(f(x))$ for all $x \in F$ and $y \in G$.*

Proof: See Corollary 0-3.5 and Theorem 0-3.6 in [GHK⁺80]. □

If F and G are frames and $\phi{:}F \to G$ is a frame morphism then, since ϕ preserves arbitrary joins, it has an upper adjoint, say $g{:}\,G \to F$, which preserves arbitrary meets by the above Lemma. Also, the upper adjoint g preserves prime elements because ϕ preserves finite meets [GHK⁺80, Lemma IV-4.5]. If $\phi : F \to G$ is also an observation frame morphism from $\alpha{:}\,F \to L$ to $\beta{:}\,G \to H$ then we have the following.

8.1.11. Lemma. *Assume ϕ is a morphism in* **OFrm** *between the observation frames $\alpha : F \to L$ and $\beta : G \to H$. Then $\phi : F \to G$ has an upper adjoint $g : G \to F$ which preserves arbitrary meets of G, prime elements and also the M-prime elements of β.*

Proof: Since an observation frame morphism ϕ is a frame morphism from F to G it has an upper adjoint $g : G \to F$ which preserves arbitrary meets of G and prime elements of G. Let now $p \in MP(\beta)$ and $S \subseteq F$, then

$$\begin{array}{rcll}
\bigsqcap \alpha(S) \sqsubseteq \alpha(g(p)) & \Rightarrow & \bigsqcap \beta(\phi(S)) \sqsubseteq \beta(\phi(g(p))) & \text{[M-multiplicativity]} \\
& \Rightarrow & \bigsqcap \beta(\phi(S)) \sqsubseteq \beta(p) & \text{[Lemma 8.1.10 (iii)]} \\
& \Leftrightarrow & \exists s \in S{:}\ \phi(s) \leq p & [p \text{ is M-prime}] \\
& \Leftrightarrow & \exists s S{:}\ s \leq g(p) & [(\phi, g) \text{ is a Galois connection}]
\end{array}$$

that is, $g(p) \in MP(\alpha)$. □

For a Galois connection (f, g) there does not seem to be any condition on g alone which implies that f preserves finite meets (see [Joh82a]). However, for lattices in which every element is the meet of all the primes above it, g preserves prime elements if and only if f preserves finite meets (see [GHK$^+$80]).

Duality for $\mathcal{T}_0$ spaces

In this subsection we define a point functor Pt from the opposite of the category of observation frames to the category **Sp** of topological spaces by topologizing the M-prime elements. We show that the functor Pt is a right adjoint to the functor $\Omega : \mathbf{Sp} \to \mathbf{OFrm}^{op}$ and that this adjunction restricts to a duality for $\mathcal{T}_0$ spaces.

In order to define a topology for the set of M-prime elements $MP(\alpha)$ of an observation frame $\alpha : F \to L$, define the 'open set' $\triangle(a)$ by

$$\triangle(a) = MP(\alpha) \setminus \uparrow a = \{p \in MP(\alpha) \mid a \not\leq p\},$$

for all a element of F.

8.1.12. Lemma. *Let $\alpha : F \to L$ be an observation frame. Then:*

$$\begin{array}{rcll} \triangle(\bigvee S) & = & \bigcup\{\triangle(a) \mid a \in S\} & \textit{for all subsets } S \textit{ of } F; \\ \triangle(\bigwedge S) & = & \bigcap\{\triangle(a) \mid a \in S\} & \textit{for all finite subsets } S \textit{ of } F. \end{array}$$

Proof: We prove only the second item. The other one is trivial.

$$\begin{array}{rcl} p \in \bigcap\{\triangle(a) \mid a \in S\} & \Leftrightarrow & p \in MP(\alpha) \text{ and } \forall a \in S \colon a \not\leq p \\ & \overset{*}{\Leftrightarrow} & p \in MP(\alpha) \text{ and } \bigwedge S \not\leq p \\ & \Leftrightarrow & p \in \triangle(\bigwedge S), \end{array}$$

where the implication ($\overset{*}{\Leftarrow}$) is trivial and for ($\overset{*}{\Rightarrow}$) we use that p is an M-prime element: if $\bigwedge S \leq p$ then also $\bigsqcap \alpha(S) \sqsubseteq \alpha(p)$ and hence $a \leq p$ for some $a \in S$. $\square$

The above lemma implies that, for every observation frame $\alpha : F \to L$, the collection of sets of the form $\triangle(a)$, for $a \in F$, forms a topology on $MP(\alpha)$. We denote it by $\mathcal{O}_\triangle(MP(\alpha))$ and the corresponding topological space by $Pt(\alpha)$. Notice that $Pt(\alpha)$ is $\mathcal{T}_0$. Indeed, let p and $q \in MP(\alpha)$ be such that $p \leq_\mathcal{O} q$ and $q \leq_\mathcal{O} p$ in the specialization preorder induced by the topology $\mathcal{O}_\triangle(MP(\alpha))$. Equivalently, $p \in \triangle(a)$ if and only if $q \in \triangle(a)$ for every $a \in F$. Hence, for every $a \in F$, $a \not\leq p$ if and only if $a \not\leq q$, that is $p = q$. Therefore the specialization preorder is a partial order, that is, the topological space $Pt(\alpha)$ is $\mathcal{T}_0$.

8.1.13. Lemma. *Let $\alpha : F \to L$ be an observation frame. The map $\varepsilon : F \to \mathcal{O}_\triangle(MP(\alpha))$ defined by $\varepsilon(a) = \triangle(a)$ for every $a \in F$ is a morphism in* **OFrm** *from α to the observation frame $\Omega(Pt(\alpha))$. It is surjective as a function.*

Proof: By Lemma 8.1.12, ε is a frame morphism which is surjective as a function by definition. It remains to prove that it is M-multiplicative. Let S and T be two subsets of F be such that $\bigsqcap \alpha(S) \sqsubseteq \bigsqcap \alpha(T)$, and take $p \in \bigcap \varepsilon(S)$. From the definition of ε, $p \in MP(\alpha)$ and $s \not\leq p$ for every $s \in S$. We claim that also $t \not\leq p$ for every $t \in T$. If not then there would exist $t \in T$ such that $t \leq p$ and hence $\alpha(t) \sqsubseteq \alpha(p)$. Since $\bigsqcap \alpha(S) \sqsubseteq \bigsqcap \alpha(T) \sqsubseteq \alpha(t) \sqsubseteq \alpha(p)$ there would exist $s \in S$ such that $s \leq p$ (because p is an M-prime element), therefore contradicting the hypothesis. Thus $p \in \triangle(t)$ for every $t \in T$, that is, $p \in \bigcap \varepsilon(T)$. □

We are now ready to formulate the relationship between topological spaces and observation frames.

8.1.14. Theorem. *Let X be a topological space, $\alpha : F \to L$ be an observation frame and ϕ be a morphism in* **OFrm** *from α to $\Omega(X)$. Then there is a unique continuous function $f_\phi : X \to Pt(\alpha)$ in* **Sp** *such that $\Omega(f_\phi) \circ \varepsilon = \phi$.*

$$\begin{tikzcd}
\alpha \arrow[r, "\varepsilon"] \arrow[dr, "\phi"'] & \Omega(Pt(\alpha)) \arrow[d, "\Omega(f_\phi)"] & Pt(\alpha) \\
& \Omega(X) & X \arrow[u, dashed, "f_\phi"]
\end{tikzcd}$$

This extends the assignment $\alpha \mapsto Pt(\alpha)$ to a functor from $\mathbf{OFrm}^{op}$ *to* **Sp** *which is a right adjoint of Ω.*

Proof: Let $a \in F$. In order to obtain the required commutativity we have to prove

$$\begin{array}{rcll}
\forall x \in X \colon x \in \phi(a) & \Leftrightarrow & x \in \Omega(f_\phi)(\varepsilon(a)) & \\
& \Leftrightarrow & f_\phi(x) \in \varepsilon(a) & [\text{definition of } \Omega(f_\phi)] \\
& \Leftrightarrow & f_\phi(x) \in \triangle(a) & [\text{definition of } \varepsilon(a)] \\
& \Leftrightarrow & a \not\leq f_\phi(x). & [\text{definition of } \triangle(a)]
\end{array}$$

This determines $f_\phi(x)$ uniquely as $\bigvee\{b \in F \mid x \notin \phi(b)\}$. Indeed, for all $x \in X$, if $a \not\leq f_\phi(x)$ then $x \in \phi(a)$ because otherwise we would have $a \in \{b \in F \mid x \notin \phi(b)\}$ and hence the contradiction $a \leq \bigvee\{b \in F \mid x \notin \phi(b)\} = f_\phi(x)$.

Conversely, if $x \in \phi(a)$ then $a \not\leq f_\phi(x)$ because otherwise $a \leq f_\phi(x) = \bigvee\{b \in F \mid x \notin \phi(b)\}$ would imply, after applying ϕ,

$$\phi(a) \subseteq \phi(\bigvee\{b \in F \mid x \notin \phi(b)\}) = \bigcup\{\phi(b) \in \mathcal{O}(X) \mid x \notin \phi(b)\}.$$

Since $x \in \phi(a)$ we would get that there exists $b \in F$ such that $x \in \phi(b)$ and $x \notin \phi(b)$.

Next we show that $f_\phi(x)$ is an M-prime element, i.e. $f_\phi(x) \in MP(\alpha)$. Let S be a subset of F such that $\bigsqcap \alpha(S) \sqsubseteq \alpha(f_\phi(x))$. Then from the definition of $f_\phi(x)$ and after applying ϕ we obtain

$$\bigcap \phi(S) \subseteq \phi(\bigvee\{a \in F \mid x \notin \phi(a)\}) = \bigcup\{\phi(a) \in \mathcal{O}(X) \mid x \notin \phi(a)\}.$$

Hence there exists $s \in S$ such that $s \leq f_\phi(x)$ because otherwise for all $s \in S$ we would have $s \not\leq f_\phi(x)$ and hence by the above, $x \in \phi(s)$ for every $s \in S$. But then $x \in \bigcap \phi(S)$ which implies there exists $a \in F$ such that $x \in \phi(a)$ and $x \notin \phi(a)$.

The function f_ϕ is also continuous. Let $a \in F$ and consider the open set $\triangle(a)$ of $Pt(\alpha)$. Then we have:

$$\begin{array}{rcll} f_\phi^{-1}(\triangle(a)) & = & \{x \in X \mid f_\phi(x) \in \triangle(a)\} & \\ & = & \{x \in X \mid a \not\leq f_\phi(x)\} & [\text{definition of } \triangle(a), f_\phi(x) \text{ M-prime}] \\ & = & \{x \in X \mid x \in \phi(a)\} & \\ & = & \phi(a). & \end{array}$$

Since $\phi(a) \in \mathcal{O}(X)$ is open, we have that f_ϕ is continuous. □

The unit of the above adjunction is given by the function η defined in the following Lemma.

8.1.15. Lemma. *Let X be a topological space. Then the unit of the adjunction between $\mathbf{OFrm}^{op}$ and $\mathbf{Sp}$ is given by function $\eta : X \to Pt(\Omega(X))$ defined by $\eta(x) = int(X \setminus \{x\}) = o_x$. It is a continuous surjective function in $\mathbf{Sp}$. Moreover, η is injective and preserves open sets if and only if X is $\mathcal{T}_0$.*

Proof: By Theorem 8.1.14 the unit of the adjunction between $\mathbf{OFrm}^{op}$ and $\mathbf{Sp}$ is uniquely determined by the function f_ϕ, where $\phi: \Omega(X) \to \Omega(X)$ is the identity morphism in $\mathbf{OFrm}^{op}$. Therefore, for every space X, the unit $\eta: X \to Pt(\Omega(X))$ is defined by $\eta(x) = \bigcup\{o \in \mathcal{O}(X) \mid x \notin o\} = o_x$. Next we show η is a continuous surjective function in $\mathbf{Sp}$.

We have already seen that the M-prime elements of $\Omega(X)$ are exactly those of the form $o_x = int(X \setminus \{x\})$ in a topological space X. Hence η is clearly onto. Let us now check it is also continuous. For $o \in \mathcal{O}(X)$ we have:

$$\begin{array}{rcl} \eta^{-1}(\triangle(o)) & = & \{x \in X \mid \eta(x) \in \triangle(o)\} \\ & = & \{x \in X \mid o \not\subseteq \eta(x)\} \\ & = & \{x \in X \mid x \in o\} \\ & = & o. \end{array}$$

This proves also that η is injective as a function and hence an isomorphism between X and $MP(\Omega(X))$. It remains to prove that it is also an open map, i.e. it preserves open sets. For $o \in \mathcal{O}(X)$ we have:

$$\begin{aligned}
\eta(o) &= \{\eta(x) \in MP(\Omega(X)) \mid x \in o\} \\
&= \{\eta(x) \in MP(\Omega(X)) \mid o \not\subseteq \eta(x)\} \quad [\text{by definition of } \eta(x)] \\
&= \{\eta(x) \in MP(\Omega(X)) \mid \eta(x) \in \triangle(o)\} \quad [\text{by definition } \triangle(o)] \\
&= \{p \in MP(\Omega(X)) \mid p \in \triangle(o)\} \quad [\eta \text{ is an isomorphism}] \\
&= \triangle(o),
\end{aligned}$$

which is open in the topology of $Pt(\Omega(X))$. Therefore, if X is a $\mathcal{T}_0$ space then η is an isomorphism in **Sp**.

Finally, if η is injective and open then it forms an isomorphism in **Sp** between X and $Pt(\Omega(X))$. However, for every observation frame $\alpha : F \to L$ the space $Pt(\alpha)$ is $\mathcal{T}_0$. Hence also X is $\mathcal{T}_0$. □

By Lemma 8.1.15 and because $Pt(\alpha)$ is $\mathcal{T}_0$ for every observation frame $\alpha{:}F \to L$, the adjunction of Theorem 8.1.14 restricts to a reflection of $\mathbf{Sp}_0$ into the full image of the functor $\Omega : \mathbf{Sp}_0 \to \mathbf{OFrm}$. Therefore the adjunction between $\mathbf{Sp}_0$ and **OFrm** is Galois. Next we characterize this full sub-category of **OFrm** and hence we prove directly that the adjunction of Theorem 8.1.14 restricts to an equivalence.

A subset X of a complete lattice L is said to be *order generating* in L (or equivalently L is said to be *order generated* by X) if

$$x = \bigwedge(\uparrow x \cap X) = \bigwedge\{y \in X \mid x \leq y\}$$

for every $x \in L$.

8.1.16. Proposition. *For $X \subseteq L$ where L is a complete lattice the following statements are equivalent.*

(i) *X is order generating in L;*
(ii) *every element of L can be written as a meet of a subset of X;*
(iii) *L is the smallest subset containing X closed under arbitrary meets;*
(iv) *whenever $y \not\leq x$, then there is a $p \in X$ with $x \leq p$ but $y \not\leq p$.*

Proof: See Proposition I.3.9 in [GHK+80]. □

For example, for a topological space X the lattice of open sets $\mathcal{O}(X)$ is order generated by $MP(\Omega(X))$: we already know that every M-prime element of $\Omega(X)$ is of the form $o_x = int(X \setminus \{x\})$ for some $x \in X$. Therefore we need to show $o = \bigwedge\{o_x \mid o \subseteq o_x\}$ for every open set o.

Clearly, $o \subseteq \bigwedge\{o_x \mid o \subseteq o_x\}$. To prove the other direction of the inclusion, consider $y \in \bigwedge\{o_x \mid o \subseteq o_x\}$ and assume towards a contradiction that $y \notin o$. Then $o_y = int(X \setminus \{y\})$ is the greatest open set not containing y, so $o \subseteq o_y$. But then $o_y \in \{o_x \mid o \subseteq o_x\}$ and hence $y \in o_y$ because, obviously, $y \in \bigwedge\{o_x \mid o \subseteq o_x\} \subseteq o_y$.

By definition, in an observation frame $\alpha : F \to L$, every element of L can be written as the meet of elements in $\alpha(F)$. The following corollary to the above proposition is then immediate.

8.1.17. Corollary. *A frame morphism $\alpha : F \to L$ from a frame F to a completely distributive lattice L is an observation frame if and only if $\alpha(F)$ is order generating L.* $\square$

Define the category of *spatial observation frames* **SOFrm** to be the full subcategory of **OFrm** with as objects the observation frames $\alpha : F \to L$ in which F is order generated by the set $MP(\alpha)$ of M-prime elements.

The previous example shows that the functor Ω maps every topological space to an object of **SOFrm**. We have already seen that the functor Pt maps every observation frame to an object of $\mathbf{Sp}_0$. Moreover for every $\mathcal{T}_0$ space X the unit of the adjunction is an isomorphism by Lemma 8.1.15. The following lemma gives a similar result for the counit.

8.1.18. Lemma. *Let $\alpha{:}F \to L$ be an observation frame. The counit morphism ε from α to $\Omega(Pt(\alpha))$ is an order isomorphism if and only if $\alpha : F \to L$ is order generated by its M-primes (i.e. it is in* **SOFrm***).*

Proof: (*only if*) Assume $a \not\leq b$ for some $a, b \in F$. Since ε is an order isomorphism (and hence order reflecting), $\varepsilon(a) = \triangle(a) \not\subseteq \triangle(b) = \varepsilon(b)$. Hence, by definition of $\triangle(-)$, there exists $p \in MP(\alpha)$ such that $a \not\leq p$ and $b \leq p$. Thus, by Proposition 8.1.16, F is order generated by $MP(\alpha)$.
(*if*) Define $\varepsilon^{-1}(\triangle(a)) = \bigwedge(MP(\alpha) \setminus \triangle(a))$ for every $a \in F$. Then we have:

$$\begin{aligned}
\varepsilon^{-1}(\varepsilon(a)) &= \varepsilon^{-1}(\triangle(a)) \\
&= \bigwedge(MP(\alpha) \setminus \triangle(a)) \\
&= \bigwedge(MP(\alpha) \setminus (MP(\alpha) \setminus \uparrow a)) \\
&= \bigwedge(MP(\alpha) \cap \uparrow a) \\
&= a.
\end{aligned}$$

Therefore ε is injective. We have already seen in Lemma 8.1.13 that ε is onto, therefore ε is an isomorphism with inverse ε^{-1}. It is also order reflecting because if $a \not\leq b$ for a and b in F, by Proposition 8.1.16, there is a $p \in MP(\alpha)$ such that $a \not\leq p$ and $b \leq a$. Therefore $\varepsilon(a) = \triangle(a) \not\subseteq \triangle(b) = \varepsilon(b)$. $\square$

Now our main result follows.

8.1.19. Corollary. *The adjunction $\mathbf{Sp} \leftrightarrows \mathbf{OFrm}^{op}$ restricts to an equivalence of categories $\mathbf{Sp}_0 \simeq \mathbf{SOFrm}^{op}$. Hence $\mathbf{Sp}_0$ and* **SOFrm** *are each other's duals and the adjunction is Galois.* $\square$

Notice that if $\alpha : F \to L$ is an observation frame such that F is order generated by the M-prime elements, then α is order-reflecting (and hence injective): by Lemma 8.1.18 there is an order isomorphism ε from α to $\Omega(Pt(\alpha))$, and by Theorem 8.1.4 there exists a unique complete distributive lattice morphism $\widetilde{\varepsilon} : L \to \mathcal{Q}(MP(\alpha))$ such that $\widetilde{\varepsilon}(\alpha(a)) = \varepsilon(a)$ for all $a \in F$. Hence, for a and b in F, if $\alpha(a) \sqsubseteq \alpha(b)$ then, by monotonicity of $\widetilde{\varepsilon}$,

$$\varepsilon(a) = \widetilde{\varepsilon}(\alpha(a)) \sqsubseteq \widetilde{\varepsilon}(\alpha(b)) = \varepsilon(b).$$

Since ε is order-reflecting, $a \leq b$. From the monotonicity of α it follows that $a \leq b$ if and only $\alpha(a) \sqsubseteq \alpha(b)$.

8.2 M-topological systems

Topological systems were introduced by Vickers [Vic89] in order to have a framework of which both topological spaces and (ordinary) frames are instances. In a topological system we have a set of subjects (points) and a set of predicates (opens) and a satisfaction relation matching the geometric propositional logic (the logic of finite observations). In this section we generalize these topological systems in order to obtain a satisfaction relation of propositional logic for observation frames (with both infinite conjunctions and disjunctions). Our interest in M-topological systems is justified since they clarify the connection between the infinitary operations of an observation frame $\alpha : F \to L$ (the arbitrary joins $\bigsqcup$ and the arbitrary meets $\bigsqcap$ living in L) and the points of α.

8.2.1. Definition. Let X be a set, let $\alpha : F \to L$ be an observation frame, and let $\models \subseteq X \times L$ be a relation. Then $(X, \models, \alpha)$ is called an *M-topological system* if and only if $\models$ satisfies

$$\begin{array}{lcl} x \models \bigsqcup S & \Leftrightarrow & \exists s \in S \colon x \models s \\ x \models \bigsqcap S & \Leftrightarrow & \forall s \in S \colon x \models s \end{array}$$

for all subsets S of L.

Directly from the above definition we can deduce that

(i) $x \models \top$ for all $x \in X$;
(ii) $x \models \bot$ for no $x \in X$;
(iii) $x \models q_1$ and $q_1 \sqsubseteq q_2$ implies $x \models q_2$ for every q_1 and q_2 in F.

Next we give the two main examples of M-topological systems. Let X be a topological space and define, for every $x \in X$ and $q \in \mathcal{Q}(X)$,

$$x \models q \text{ if and only if } x \in q.$$

Then $\mathcal{T}(X) = (X, \models, \Omega(X))$ is obviously an M-topological system.

Let $\alpha{:}\, F \to L$ be an observation frame and define a relation $\models\, \subseteq MP(\alpha) \times L$ by

$$p \models q \text{ if and only if } \forall a \in F\colon q \sqsubseteq \alpha(a) \;\Rightarrow\; a \not\leq p.$$

Next we show that $\mathcal{S}(\alpha) = (MP(\alpha), \models, \alpha)$ is an M-topological system. Let p be an M-prime element of α, $q \in L$, $\phi_p : F \to 2$ is the morphism in **OFrm** corresponding to the M-prime element $p \in F$ (Lemma 8.1.8) and let $\widetilde{\phi_p} : L \to 2$ be the unique complete distributive lattice morphism such that $\widetilde{\phi_p} \circ \alpha = \phi_p$ (Theorem 8.1.4). We have

$$\begin{aligned}
p \models q \quad &\Leftrightarrow \quad (\forall a \in F\colon q \sqsubseteq \alpha(a) \;\Rightarrow\; a \not\leq p)\\
&\Leftrightarrow \quad (\forall a \in F\colon q \sqsubseteq \alpha(a) \;\Rightarrow\; \phi_p(a) = \top) \quad \text{[Lemma 8.1.8]}\\
&\Leftrightarrow \quad \bigwedge\{\phi_p(a) \mid q \sqsubseteq \alpha(a)\} = \top\\
&\Leftrightarrow \quad \bigwedge\{\widetilde{\phi_p}(\alpha(a)) \mid q \sqsubseteq \alpha(a)\} = \top \quad \text{[Theorem 8.1.4]}\\
&\Leftrightarrow \quad \widetilde{\phi_p}(\textstyle\bigsqcap\{\alpha(a) \mid q \sqsubseteq \alpha(a)\}) = \top \quad [\widetilde{\phi_p} \text{ preserves meets}]\\
&\Leftrightarrow \quad \widetilde{\phi_p}(q) = \top. \quad [q = \textstyle\bigsqcap\{\alpha(a) \mid q \sqsubseteq \alpha(a)\}]
\end{aligned}$$

Since $\widetilde{\phi_p}$ preserves arbitrary joins and arbitrary meets, it is now easy to verify that $\mathcal{S}(\alpha)$ is an M-topological system.

Next we organize M-topological systems in a category for which we introduce the following morphisms.

8.2.2. Definition. Let $\alpha{:}F \to L$ and $\beta{:}G \to H$ be two observation frames, and $D = (X, \models, \alpha)$ and $E = (Y, \models, \beta)$ be two M-topological systems. A *morphism* from D to E consists of a pair (f, ϕ) where f is a function from X to Y and ϕ is a morphism from β to α in **OFrm** such that, for every $x \in X$ and $a \in G$,

$$x \models \alpha(\phi(a)) \text{ if and only if } f(x) \models \beta(a).$$

It is straightforward to check that composition of two morphisms defined as the usual element-wise composition is again a morphism. Hence M-topological systems together with these morphisms form a category to which we refer to as **MTS**.

For a continuous function $f : X \to Y$ in **Sp**, the pair $\mathcal{T}(f) = (f, \Omega(f))$ is a morphism from $\mathcal{T}(X) = (X, \models, \Omega(X))$ to $\mathcal{T}(Y) = (Y, \models, \Omega(Y))$ in **MTS** because

$$\begin{aligned}
x \models \Omega(f)(o) \quad &\Leftrightarrow \quad x \in \Omega(f)(o) \quad [\text{definition of} \models \text{in } \mathcal{T}(X)]\\
&\Leftrightarrow \quad f(x) \in o \quad [\text{definition of } \Omega(f)]\\
&\Leftrightarrow \quad f(x) \models o. \quad [\text{definition of} \models \text{in } \mathcal{T}(Y)]
\end{aligned}$$

It is easy to check that $\mathcal{T}$ is a functor from **Sp** to **MTS**.

Next we show that the adjunction of Theorem 8.1.14 can be split into two parts: one from topological spaces to M-topological systems and one from M-topological systems to observation frames. We start with the first adjunction.

Every M-topological system $D = (X, \models, \alpha : F \to L)$ induces a topology on X by taking as open sets the *extent* of all $a \in F$:

$$ext(a) = \{x \in X \mid x \models \alpha(a)\}.$$

By definition of $\models$ and since α preserves finite meets and arbitrary joins we have that the collection of all extents forms a topology on X. We denote this topological space by $Sp(D)$. Furthermore the assignment $a \mapsto ext(a)$ is a morphism in **OFrm** from α to $id_{\mathcal{P}(X)} : \mathcal{P}(X) \to \mathcal{P}(X)$. Indeed, it is a frame morphism as the collection of all extents forms a topology and it is M-multiplicative because if $\bigsqcap \alpha(S) \sqsubseteq \bigsqcap \alpha(T)$ for subsets S and T of F then

$$\begin{array}{rcll} x \in \bigcap ext(S) & \Leftrightarrow & \forall s \in S\colon x \models \alpha(s) & [\text{definition of } ext(-)] \\ & \Leftrightarrow & x \models \bigsqcap \alpha(S) & [\text{definition of } \models] \\ & \Rightarrow & x \models \bigsqcap \alpha(T) & \\ & \Leftrightarrow & \forall t \in T\colon x \models \alpha(t) & [\text{definition of } \models] \\ & \Leftrightarrow & x \in \bigcap ext(T). & [\text{definition of } ext(-)] \end{array}$$

This shows also that the pair (id_X, ext) is a morphism in **MTS** from $\mathcal{T}(Sp(D))$ to D.

8.2.3. Theorem. *Let $D = (X, \models, \alpha : F \to L)$ be an M-topological system and let Y be a topological space such that there is a morphism (f, ϕ) in* **MTS** *from $\mathcal{T}(Y)$ to D. Then there exists a unique continuous function $g : Y \to Sp(D)$ in* **Sp** *such that the following diagram commutes*

$$\begin{tikzcd} D & \mathcal{T}(Sp(D)) \arrow[l, "(id_X, ext)"'] & & Sp(D) \\ & \mathcal{T}(Y) \arrow[ul, "(f,\phi)"] \arrow[u, "\mathcal{T}(g)"'] & & Y \arrow[u, dashed, "g"] \end{tikzcd}$$

This extends Sp to a functor from **MTS** *to* **Sp** *which is a right adjoint of $\mathcal{T}$.*

Proof: Take $g = f$. It is clearly continuous and the unique one such that $\mathcal{T}(g)$ makes the diagram commute. □

An M-topological system is called *spatial* if it is isomorphic in **MTS** to $\mathcal{T}(X)$ for some topological space X.

Next we give the second adjunction between M-topological systems and observation frames. There is an obvious forgetful functor **MTS** $\to$ $\mathbf{OFrm}^{op}$ which maps every M-topological system $(X, \models, \alpha : F \to L)$ to α and every morphism (f, ϕ) in **MTS** to the observation frame morphism ϕ.

8.2.4. Lemma. *Let $D = (X, \models, \alpha : F \to L)$ be an M-topological system and $p : X \to MP(\alpha)$ be a function which assigns to a concrete point $x \in X$ the abstract point $p(x) = \bigvee\{a \in F \mid x \not\models \alpha(a)\}$. Then $p(x)$ is an M-prime element of α for every $x \in X$ and the pair (p, id_F) is a morphism in* **MTS** *from D to $\mathcal{S}(\alpha)$.*

Proof: We first show that $p(x)$ is M-prime. Let $S \subseteq F$ be such that $\bigsqcap \alpha(S) \sqsubseteq \alpha(p(x))$. There must exist an $s \in S$ such that $s \leq p(x)$, (or, equivalently, $x \not\models \alpha(s)$) because otherwise $x \models \bigsqcap \alpha(S)$. Since $\bigsqcap \alpha(S) \sqsubseteq \alpha(p(x))$, $x \models \alpha(p(x))$. But then $x \models \alpha(\bigvee\{a \in F \mid x \not\models \alpha(a)\}) = \bigsqcup\{a \in F \mid x \not\models \alpha(a)\}$. From the definition of $\models$ this holds if and only if there exists $a \in F$ such that $x \not\models \alpha(a)$ and $x \models \alpha(a)$. Contradiction.

Consider now the pair $(p, id_F) : D \to \mathcal{S}(\alpha)$ where $D = (X, \models, \alpha : F \to L)$. We show that it forms a morphism in **MTS**. By the above it is enough to prove $x \models \alpha(a)$ if and only if $p(x) \models \alpha(a)$, where $p(x) \models \alpha(a)$ means $\alpha(a) \sqsubseteq \alpha(b)$ implies $b \not\leq p(x)$ for all $b \in F$.

$\Rightarrow$) Let $b \in F$ be such that $\alpha(a) \sqsubseteq \alpha(b)$. Then $b \not\leq p(x) = \bigvee\{c \in F \mid x \not\models \alpha(c)\}$ because otherwise $x \models \alpha(a)$ and $\alpha(a) \sqsubseteq \alpha(b)$ implies $x \models \alpha(b)$ and hence also $x \models \alpha(p(x))$. But this leads us to the contradiction that there exists a $c \in F$ such that $x \not\models \alpha(c)$ and $x \models \alpha(c)$.

$\Leftarrow$) If $p(x) \models \alpha(a)$ then $a \not\leq p(x) = \bigvee\{b \in F \mid x \not\models \alpha(b)\}$. Hence $x \models \alpha(a)$ because otherwise $a \in \{b \in F \mid x \not\models \alpha(b)\}$ and hence the contradiction $a \leq p(x)$. □

8.2.5. Theorem. *Let $\alpha : F \to L$ be an observation frame and let D be an M-topological system $(Y, \models, \beta : G \to H)$ such that there is a morphism ϕ from α to β in* **OFrm**. *Then there exists a unique function $g : Y \to MP(\alpha)$ such that (g, ϕ) is a morphism in* **MTS** *from D to $\mathcal{S}(\alpha)$.*

Proof: Define $g(y) = \bigvee\{b \in F \mid x \not\models \beta(\phi(b))\}$ for all $y \in Y$. It is not hard to see that $g(y)$ is an M-prime element of α. We only prove $y \models \beta(\phi(a))$ if and only if $g(y) \models \alpha(a)$ for all $y \in Y$ and $a \in F$.

$\Rightarrow$) If $y \models \beta(\phi(a))$ then also $g(y) \models \alpha(a)$ because otherwise by definition of $\models$ in $\mathcal{S}(\alpha)$ there exists $b \in F$ such that $\alpha(a) \sqsubseteq \alpha(b)$ and $b \leq g(y) = \bigvee\{c \in F \mid y \not\models \beta(\phi(c))\}$. Hence, by M-multiplicativity of ϕ,

$$\begin{aligned} \beta(\phi(a)) &\sqsubseteq \beta(\phi(b)) \\ &\sqsubseteq \beta(\phi(\bigvee\{c \in F \mid y \not\models \beta(\phi(c))\})) \\ &= \bigsqcup\{\beta(\phi(c)) \mid y \not\models \beta(\phi(c))\}. \end{aligned}$$

But $y \models \beta(\phi(a))$ and hence also $y \models \bigsqcup\{\beta(\phi(c)) \mid y \not\models \beta(\phi(c))\}$. Therefore, by definition of $\models$, we get the contradiction that there exists $c \in F$ such that $y \not\models \beta(\phi(c))$ and $y \models \beta(\phi(c))$.

$\Leftarrow$) If $g(y) \models \alpha(a)$ then $a \not\leq g(y) = \bigvee\{b \in F \mid x \not\models \beta(\phi(b))\}$ by definition of $\models$ in $\mathcal{S}(\alpha)$. But then $y \models \beta(\phi(a))$ because otherwise $a \in \{b \in F \mid x \not\models \beta(\phi(b))\}$ and hence $a \leq \bigvee\{b \in F \mid x \not\models \beta(\phi(b))\}$ contradicting $a \not\leq g(y)$.

Clearly g is the unique function with the required property. □

As a consequence of the above theorem, the assignment $\alpha \mapsto \mathcal{S}(\alpha)$ extends to a functor from $\mathbf{OFrm}^{op}$ to $\mathbf{MTS}$ which is a right adjoint of the forgetful functor $\mathbf{MTS} \to \mathbf{OFrm}^{op}$.

An M-topological system is called *observational* if it is isomorphic in **MTS** to $\mathcal{S}(\alpha)$ for some observation frame $\alpha : F \to L$. Clearly the full sub-category of spatial observational M-topological systems is equivalent to the category of $\mathcal{T}_0$ topological spaces and is equivalent to the category of observation frames $\alpha : F \to L$ with F order generated by the M-prime elements of α. The following diagram summarizes the situation.

$$\begin{tikzcd}
\mathbf{Sp} \arrow[r, shift right, "\mathcal{T}"'] \arrow[r, phantom, "\top"] & \mathbf{MTS} \arrow[l, shift right, "Sp"'] \arrow[r, shift right] \arrow[r, phantom, "\top"] & \mathbf{OFrm}^{op} \arrow[l, shift right, "\mathcal{S}"'] \\
\mathbf{Sp}_0 \arrow[u, hook] \arrow[r, phantom, "\simeq"] & \mathbf{MTS_M} \arrow[u, hook] \arrow[r, phantom, "\simeq"] & \mathbf{SOFrm}^{op} \arrow[u, hook]
\end{tikzcd}$$

Next we show that spatiality of an observational M-topological system corresponds to the completeness of the logic. To show this we need the definition of semantic entailment.

8.2.6. Definition. For an observation frame $\alpha : F \to L$ define the relation of *semantic entailment* on F as follows. For all elements a and b of F, $a \vdash_F b$ if and only if for every M-topological system $(X, \models, \alpha)$ and $x \in X$, $x \models \alpha(a)$ implies $x \models \alpha(b)$.

We also define the relation of semantic entailment on L for all q and r in L by putting $q \vdash_L r$ if and only if for every M-topological system $(X, \models, \alpha)$ and $x \in X$, $x \models q$ implies $x \models r$.

The next lemma states that for every observation frame $\alpha : F \to L$ the order on F is contained in the entailment relation. This gives us the soundness for the logic of F. Similarly we have soundness for the logic of L.

8.2.7. Lemma (soundness). *Let $\alpha : F \to L$ be an observation frame. Then*
(i) *$a \leq b$ implies $a \vdash_F b$ for all $a, b \in F$,*
(ii) *$q \sqsubseteq r$ implies $q \vdash_L r$ for all $q, r \in L$.*

Proof: We prove only the first item. If $a \leq b$ for $a, b \in F$ then $\alpha(a) \sqsubseteq \alpha(b)$. Therefore, in every M-topological system $(X, \models, \alpha)$ if $x \models \alpha(a)$ then $x \models \alpha(b)$. Hence $a \vdash_F b$. The proof of the second item is equally simple. □

The entailment relation of F and L is included in the order of F, L respectively, if and only if the observation frame $\alpha : F \to L$ is such that F is order generated by the M-prime elements of α.

8.2.8. Lemma (completeness). *Let $\alpha : F \to L$ be an observation frame. The following statements are equivalent.*

(i) *F is order generated by the M-prime elements of α;*
(ii) *$a \vdash_F b$ implies $a \leq b$ for all a and b in F;*
(iii) *$q \vdash_L r$ implies $q \sqsubseteq r$ for all q and r in L and α is order-reflecting.*

Proof: (i) $\Rightarrow$ (ii). Suppose F is order generated by the M-prime elements of $\alpha : F \to L$ and let $a \vdash_F b$ for some $a, b \in F$. Hence for all M-topological systems $(X, \models, \alpha)$ and $x \in X$ if $x \models \alpha(a)$ then $x \models \alpha(b)$. In particular consider the M-topological system $\mathcal{S}(\alpha)$ and the isomorphism (id_{MP}, ε) in **MTS** from $\mathcal{S}(\alpha)$ to $\mathcal{T}(Pt(\alpha))$. We have

$$\begin{array}{rcll} p \in \triangle(a) & \Leftrightarrow & p \models \triangle(a) & [\text{definition of} \models \text{in } \mathcal{T}(Pt(\alpha))] \\ & \Leftrightarrow & p \models \varepsilon(a) & [\text{definition of } \varepsilon] \\ & \Leftrightarrow & p \models \alpha(a) & [(id_{MP}, \varepsilon) \text{ is a morphism in } \mathbf{MTS}] \\ & \Rightarrow & p \models \alpha(b) & [\text{ because } a \vdash_F b\ ] \\ & \Leftrightarrow & p \in \triangle(b). & \end{array}$$

Hence $\varepsilon(a) = \triangle(a) \subseteq \triangle(b) = \varepsilon(b)$. But ε is an order-preserving isomorphism by Lemma 8.1.18, therefore $a \leq b$.
(ii) $\Rightarrow$ (i). We use the formulation of Proposition 8.1.16(iv). Let $a, b \in F$ be such that $a \not\leq b$. Then $a \not\vdash_F b$, that is, there exists an M-topological system $D = (X, \models, \alpha)$ and an $x \in X$ such that $x \models \alpha(a)$ but $x \not\models \alpha(b)$. Consider the morphism (p, id_F) in **MTS** from D to $\mathcal{S}(\alpha)$. Then $x \models \alpha(a)$ if and only if $p(x) \models \alpha(a)$, where $p(x) \in MP(\alpha)$. Hence, by definition of $\models$ in $\mathcal{S}(\alpha)$ we have found an M-prime element $p(x)$ such that $a \not\leq p(x)$ but $b \leq p(x)$. Therefore, by Proposition 8.1.16 we have that F is order generated by $MP(\alpha)$.
(ii) $\Rightarrow$ (iii). Suppose $a \vdash_F b$ implies $a \leq b$ for all $a, b \in F$ and let $q, r \in L$ be such that $q \vdash_L r$. Then for all M-topological systems $(X, \models, \alpha)$ and $x \in X$ we have that $x \models q$ implies $x \models r$. But $q = \bigsqcap\{\alpha(a) \mid a \in F \ \&\ q \sqsubseteq \alpha(a)\}$ and also $r = \bigsqcap\{\alpha(b) \mid b \in F \ \&\ r \sqsubseteq \alpha(b)\}$, hence, by definition of $\models$, $x \models \alpha(a)$ for all $a \in F$ such that $q \sqsubseteq \alpha(a)$, implies $x \models \alpha(b)$ for all $b \in F$ such that $r \sqsubseteq \alpha(b)$. But this means that $a \vdash_F b$ for $a, b \in F$ such that $q \sqsubseteq \alpha(a)$ and $r \sqsubseteq \alpha(b)$. Hence $a \leq b$ (which implies $\alpha(a) \sqsubseteq \alpha(b)$) for all $a, b \in F$ such that $q \sqsubseteq \alpha(a)$ and $r \sqsubseteq \alpha(b)$. Therefore $q \sqsubseteq r$. It is easy to see that α reflects the order: assume $\alpha(a) \sqsubseteq \alpha(b)$, but $a \not\leq b$. Then $a \not\vdash_F b$ and hence $\alpha(a) \not\vdash_L \alpha(b)$.

This is a contradiction.
(iii) $\Rightarrow$ (ii). If $a \vdash_F b$, then $\alpha(a) \vdash_L \alpha(b)$, so $\alpha(a) \sqsubseteq \alpha(b)$ and thus $a \leq b$ since α reflects the order. $\square$

8.3 Some further equivalences

In this section we restrict our attention to sub-categories of **Sp**. We consider topological spaces which are not, in general, sober. For these spaces we give a duality by restricting the adjunction of Theorem 8.1.14. Of special interest is a duality for the category **PoSet**. We derive a pointless version of the (directed) ideal completion of posets.

$\mathcal{T}_1$ spaces

An observation frame $\alpha : F \to L$ will be called *atomic* if for all M-prime elements p and q of α, if $p \leq q$ then $p = q$. The full sub-category of **OFrm** whose objects are atomic observation frames is denoted by $\mathbf{OFrm_A}$, whereas the full sub-category of **SOFrm** whose objects are atomic observation frames is denoted by $\mathbf{SOFrm_A}$.

8.3.1. Lemma. *The functors $\Omega : \mathbf{Sp} \to \mathbf{OFrm}^{op}$ and $Pt : \mathbf{OFrm}^{op} \to \mathbf{Sp}$ restrict to an adjunction between the category of $\mathcal{T}_1$ spaces $\mathbf{Sp}_1$ and the category of atomic observation frames ${\mathbf{OFrm_A}}^{op}$. Hence we have a duality between $\mathbf{Sp}_1$ and $\mathbf{SOFrm_A}$.*

Proof: If a space X is $\mathcal{T}_1$ then the specialization preorder is the equality. Moreover, since every $\mathcal{T}_1$ space is $\mathcal{T}_0$, we have that points are M-prime elements $o_x = \bigcup\{o \in \mathcal{O}(X) \mid x \notin o\}$. Therefore, for every o_x and o_y in $MP(\Omega(X))$ of a given $\mathcal{T}_1$ space X, if $o_x \subseteq o_y$ then $x \leq y$ and hence $x = y$, i.e. $o_x = o_y$.

Conversely, let $\alpha : F \to L$ be an atomic observation frame and take $p, q \in MP(\alpha)$ with $p \neq q$. This implies $p \not\leq q$ or $q \not\leq p$. Suppose $p \not\leq q$ then clearly q is in the open $\triangle(p) = \{r \in MP(\alpha) \mid p \not\leq r\}$ but p is not. The other case can be treated similarly. Hence $Pt(\alpha)$ is a $\mathcal{T}_1$ space. $\square$

Notice that for an atomic observation frame $\alpha : F \to L$ with F order generated by the M-prime elements, there can be no element different from the $\top$ which is above some other M-prime element. This means that the M-prime elements of α are exactly the *co-atoms* of F (that is, maximal elements which differ from the top).

Locally open compact spaces

Denote by $\mathbf{OKSp}_0$ the full sub-category of $\mathbf{Sp}_0$ whose objects are locally open compact spaces. Let $\mathbf{SOFrm}_{Alg}$ denote the full sub-category of **SOFrm** whose objects are observation frames $\alpha : F \to L$ such that F is an algebraic lattice.

8.3.2. Lemma. *The functors* $\Omega : \mathbf{Sp} \to \mathbf{OFrm}^{op}$ *and* $Pt : \mathbf{OFrm}^{op} \to \mathbf{Sp}$ *restrict to a duality between* $\mathbf{OKSp}_0$ *and* $\mathbf{SOFrm}_{Alg}$.

Proof: It is enough to prove that a space X is open compact if and only if $\mathcal{O}(X)$ is an algebraic complete lattice. Let X be an open compact space and let $o \in \mathcal{O}(X)$. For every $x \in o$, since X is open compact, there exists a compact open u such that $x \in u \subseteq o$. Hence $o \subseteq \bigcup\{u \in \mathcal{KO}(X) \mid u \subseteq o\}$. The reverse inclusion is clear, and hence $\mathcal{O}(X)$ is algebraic.

Conversely, if $\mathcal{O}(X)$ is algebraic then for every open set o we have $o = \bigcup\{u \in \mathcal{KO}(X) \mid u \subseteq o\}$. Hence for every $x \in X$, if $x \in o$ then there exists a compact open $u \in \mathcal{KO}(X)$ such that $x \in u \subseteq o$, that is, X is open compact. $\square$

Posets and complete lattices

Let $\mathbf{AlSp}_0$ denote the full sub-category of $\mathbf{Sp}$ whose objects are $\mathcal{T}_0$ spaces X in which open sets are closed under arbitrary intersection (i.e. they form the Alexandrov topology). The full and faithful functor from the category **PoSet** (posets and monotone functions) to $\mathbf{Sp}_0$ which maps a poset $(X, \leq)$ to the underlying set X equipped with the Alexandrov topology, determines an equivalence of categories between **PoSet** and $\mathbf{AlSp}_0$.

8.3.3. Lemma. *The functors* $\Omega : \mathbf{Sp} \to \mathbf{OFrm}^{op}$ *and* $Pt : \mathbf{OFrm}^{op} \to \mathbf{Sp}$ *restrict to an equivalence between* $\mathbf{AlSp}_0$ *and* $\bigwedge$-$\mathbf{SOFrm}^{op}$ *, the full sub-category of* $\mathbf{SOFrm}^{op}$ *whose objects are observation frames* $\alpha : F \to L$ *for which* α *preserves arbitrary meets and* F *is order generated by the M-prime elements of* α.

Proof: It is enough to prove for every $\alpha : F \to L$ in $\bigwedge$-**SOFrm** that $\bigcap \triangle(A) = \triangle(\bigwedge A)$ for every $A \subseteq F$.

$$\begin{aligned} p \in \bigcap\{\triangle(a) \in a \in A\} \quad &\Leftrightarrow \quad p \in MP(\alpha) \text{ and } \forall a \in A\text{: } a \not\leq p \\ &\overset{*}{\Leftrightarrow} \quad p \in MP(\alpha) \text{ and } \bigwedge A \not\leq p \\ &\Leftrightarrow \quad p \in \triangle(\bigwedge A) \end{aligned}$$

where the implication ($\overset{*}{\Leftarrow}$) is trivial and for ($\overset{*}{\Rightarrow}$) we use that $p \in MP(\alpha)$ and the following contradiction: if $\bigwedge A \leq p$ then also $\alpha(\bigwedge A) = \bigsqcap \alpha(A) \sqsubseteq \alpha(p)$ and hence $a \leq p$ for some $a \in A$. $\square$

Let now **CLat** be the category whose objects are complete lattices and whose morphisms are functions preserving arbitrary joins and arbitrary meets. Given a complete lattice L, an element $p \in L$ is called *completely prime* if $\bigwedge A \leq p$ for $A \subseteq L$ implies there exists $a \in A$ such that $a \leq p$.

8.3.4. Lemma. *The category* $\bigwedge$-**SOFrm** *is equivalent to* **SCDL**, *the full sub-category of* **CLat** *whose objects are completely distributive lattices order generated by the completely prime elements.*

Proof: Let $\alpha : F \to L$ in $\bigwedge$-**SOFrm**. Then α is order-reflecting, preserving arbitrary meets and arbitrary joins. Since every element of L is the meet of elements of $\alpha(F)$, α is an isomorphism between the complete lattices F and L. Therefore F is a completely distributive lattice. It is not hard to see that a morphism in $\bigwedge$-**SOFrm** between two observation frames preserves arbitrary meets and arbitrary joins.

Conversely, if L is a complete distributive lattice order generated by its completely prime elements, then $id_L : L \to L$ is clearly an object in $\bigwedge$-**SOFrm**. Moreover, every morphism between completely distributive lattices is a morphism between the corresponding observation frames. Since the two constructions are each other's inverse, we obtain the required equivalence of categories. $\square$

Since complete rings of sets are closed under arbitrary unions and intersections, they are in one-to-one correspondence with posets taken with the Alexandrov topology. By combining Lemma 8.3.3 and Lemma 8.3.4 it follows that a complete lattice L is isomorphic with a complete ring of sets if and only if every element of L is the meet of a set of completely prime elements. A very similar result is due to Raney [Ran52]: A complete lattice L is isomorphic to a complete ring of sets if and only if every element of L is the join of a set of *completely join-irreducible elements* (an element $x \in L$ is called completely join-irreducible if for every $S \subseteq L$ such that $x \leq \bigvee S$ there exists $y \in S$ such that $x \leq y$).

The above result suggests that we can give a sharper representation theorem for the category **PoSet** in terms of algebraic completely distributive lattices.

8.3.5. Corollary. *The category* **PoSet** *is dual to the category* **AlgCDL** *(the full sub-category of* **CDL** *whose objects are algebraic completely distributive lattices). This duality is given by the functor* $\mathcal{O}_{Al}(-) : \mathbf{PoSet} \to \mathbf{AlgCDL}^{op}$ *which assigns to a poset its Alexandrov topology and by* $MP(-)$ *which assigns to every algebraic completely distributive lattice* L *the set of all its completely prime elements ordered as in* L^{op} *(the specialization order induced by the topological space* $Pt(id_L : L \to L)$*).*

Proof: Every poset X in the Alexandrov topology is an open compact space. Hence by Lemma 8.3.2 and Lemma 8.3.4 we have that every complete lattice L which is order generated by its M-prime elements is an algebraic complete lattice. Since the Alexandrov topology is a complete ring of sets, it is also a completely distributive lattice. Therefore $\mathcal{O}_{Al}(X)$ is an algebraic completely distributive lattice.

Conversely, it is enough to prove that every algebraic completely distributive lattice L is order generated by its completely prime elements. We begin by showing, using Lemma 8.1.16, that L is order generated by its completely irreducible elements, where $p \in L$ is called completely irreducible if for all $S \subseteq L$ such that $p = \bigwedge S$ there exists $s \in S$ such that $p = s$ (notice that if p is completely prime then p is completely irreducible, but the converse, in general, is not true). Let $x, y \in L$ such that $x \not\leq y$. Since L is algebraic by [GHK⁺80][Theorem 4.22, page 93] there exists a completely irreducible element $p \in L$ such that $x \leq p$ but $y \not\leq p$. We need to prove now that p is completely prime. Let $S \subseteq L$ be such that $p \leq \bigwedge S$. Then $p \vee \bigwedge S = p$ and hence by complete distributivity of L we obtain $\bigwedge\{p \vee s \mid s \in S\} = p$. Since p is completely irreducible there exists $s \in S$ such that $p \vee s = p$, that is $s \leq p$. Therefore p is completely prime and by Lemma 8.1.16 we have that L is order generated by its completely prime elements. □

By combining the above result with Lemma 8.3.1 we obtain that the category of sets **Set** is dual to the category of atomic algebraic completely distributive lattices.

In [GHK⁺80, AJ94] it is shown that the category **AlgCDF** of algebraic completely distributive frames with frame morphisms is dual to the category **AlgPos** of algebraic dcpo's and Scott continuous functions. This duality is given by the functor $\mathcal{O}_{Sc}(-) : \mathbf{AlgPos} \to \mathbf{AlgCDF}^{op}$ which assigns to every algebraic dcpo its Scott topology and by *Spec*$(-)$ which assigns to every algebraic completely distributive frame F the set of all its prime elements *Spec*(F) with the inherited opposite order. Notice that the categories **AlgCDF** and **AlgCDL** differ only in the morphisms: they preserve finite meets and arbitrary joins in the first category and both arbitrary meets and joins in the second one. Hence, we have an inclusion functor $i : \mathbf{AlgCDL} \hookrightarrow \mathbf{AlgCDF}$.

Next we give the pointless version of a result by Hoffmann [Hof79]: the soberification of a poset in its Alexandrov topology equals the ideal completion in its Scott topology. Alternatively we can see it as the ideal completion of a poset without considering points and even without considering ideals but working only on the lattice-side of the duality. The function which maps every poset P to the set of its directed ideals *Idl*(P) ordered by subset inclusion, extends to a functor *Idl*$(-) : \mathbf{PoSet} \to \mathbf{AlgPos}$ which is a left adjoint of the forgetful functor $U : \mathbf{AlgPos} \to \mathbf{PoSet}$ (see for example [Plo81a]).

8.3.6. Lemma. *The inclusion functor* $i : \mathbf{AlgCDL} \hookrightarrow \mathbf{AlgCDF}$ *has a left adjoint* j *that is given by assigning to every algebraic complete lattice* L *the Alexandrov topology of the set of all prime elements Spec*(L) *(with the opposite* *of the inherited order). Moreover the two squares below commute.*

$$\begin{tikzcd}
\mathbf{PoSet} \arrow[d, shift right=2, "Idl"'] \arrow[d, phantom, "\dashv"] \arrow[r, phantom, "\simeq"] & \mathbf{AlgCDL}^{op} \arrow[d, hook, shift right=2, "i^{op}"'] \arrow[d, phantom, "\dashv"] \\
\mathbf{AlgPos} \arrow[u, shift right=2, "U"'] \arrow[r, phantom, "\simeq"] & \mathbf{AlgCDF}^{op} \arrow[u, shift right=2, "j^{op}"']
\end{tikzcd}$$

Proof: Let us first notice that the inclusion functor i^{op} is naturally isomorphic to the functor given by the composition $\mathcal{O}_{Sc} \circ Idl \circ MP(-)$. Indeed for every algebraic complete lattice L we have $\mathcal{O}_{Sc}(Idl(MP(L))) \cong \mathcal{O}_{Al}(MP(L)) \cong L = i^{op}(L)$, where the latter isomorphism holds by Corollary 8.3.5. Naturality follows from the fact that the functor $Idl\colon \mathbf{PoSet} \to \mathbf{AlgPos}$ is faithful.

Since the functor $\mathcal{O}_{Sc} \circ Idl \circ MP(-)$ has a right adjoint, namely $\mathcal{O}_{Al} \circ U \circ Spec(-) = \mathcal{O}_{Al} \circ Spec(-) = j^{op}(-)$, we have that $j^{op}(-)$ is also a right adjoint of $i^{op}(-)$. Therefore $j : \mathbf{AlgCDF} \to \mathbf{AlgCDL}$ is a left adjoint of $i(-)$. Commutativity of the diagram is immediate from the definition of $j(-)$. $\Box$

The above lemma implies that the completely distributive lattice of the Alexandrov opens of a dcpo X is free over the frame of the Scott opens of X. In the next chapter we will generalize the above results to sober spaces.

8.4 Concluding notes

In this chapter we studied topological spaces in terms of the inclusion of the lattice of open sets into the lattice of saturated sets. Our representation theorem differs significantly from previous representation theorems of (some) topological spaces in algebraic terms because observation frames are not algebras in the traditional sense. However, in the next chapter we will prove that observation frames are also algebraic structures, even if they are not set-based.

Our work on observations frames is strictly related to the work of Jónsson and Tarski on Boolean algebras [JT51, JT52]. In particular we generalize the notion of perfect extension of a Boolean algebra to that of observation frame (a completely distributive lattice which 'perfectly extends' a frame). Both the extension theorems for Boolean algebras and for Boolean algebras with operators in [JT51] are generalized to frames (in Theorem 9.1.5 and Theorem 8.1.4, respectively).

Our main application for the theory we developed so far is that it can be used to extend a finitary logic based on a topological model to an infinitary one without changing the model. We will treat an example of such an extension in Chapter 10.

Further research is needed to describe limits, colimits, monos, and epis in the category of observation frames. A related question is whether the category of observation frames is any good for doing some form of pointless topology, as in the category of frames [Joh91].

Finally, we mention one more point which needs to be exploited further: it is possible to find an algebraic representation of general (non-sober) directed complete partial orders with the Scott topology. In fact the category of dcpo's is fully and faithfully embedded into $\mathbf{Sp}_0$, and hence into some full sub-category of **OFrm**.

Chapter 9

Frames and observation frames

Traditionally, topological spaces are studied in an abstract way by considering the lattice of open sets. In this case, it is convenient to regard open sets as elements of a frame. Frames are infinitary algebras which can be presented by a set together with a proper class of operations (finite meets and arbitrary joins) which satisfy some suitable axioms. In particular finite meets distribute over infinite joins. For any set X the free frame over X exists, and every frame F can be presented as the free frame over a set of generators modulo some (proper class of) equations. The category of frames is of interest because it is the 'right' place to do pointless topology [Joh91]. Indeed, the (opposite of the) category of frames is related by an adjunction to the category of topological spaces. In particular, a full sub-category of frames is dual to the category of sober topological spaces [Joh82a].

In this chapter we continue our study of topological spaces as functions mapping the lattice of open sets into the lattice of saturated sets. We relate these two abstract approaches to topology by constructing the free observation frame over a given frame. Abstract points are preserved in constructing the free observation frame. As a consequence we obtain a characterization of sober spaces in terms of its lattice of saturated sets, which will be the key mathematical tool in the next chapter in which we give a conservative extension of Abramsky's finitary domain logic of transition systems [Abr91b] to an infinitary domain logic. We also study the relationship between ordinary filters and M-filters.

Observation frames are also shown to be (categorical) algebras over onto functions between two sets, that is, the forgetful functor from the category of observation frames to the category of onto functions between sets (with the evident commutative squares as morphisms) is monadic. Using results on the presentation of frames and of completely distributive lattices, a universal presentation of observation frames is given.

9.1 Two infinitary algebraic theories

In this section we investigate frames and completely distributive lattices as infinitary algebras. We show that for the theory of frames, as well as for the theory of completely distributive lattices, every presentation always presents an algebra.

Presenting frames

The algebraic theory of frames has a proper class of operators, $\bigwedge_I$ (the I-ary meet for each finite set I) and $\bigvee_J$ (the J-ary join for each set J), and a proper class of equations:

(i)	$x \wedge y = y \wedge x$	(commutativity)
(ii)	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$	(associativity)
(iii)	$x \wedge x = x$	(idempotency)
(iv)	$x \wedge \top = x$	(unit)
(v)	$x_k \wedge \bigvee\{x_j \mid j \in J\} = x_k$ for $k \in J$	(join absorption)
(vi)	$x \wedge \bigvee\{x_j \mid j \in J\} = \bigvee\{x \wedge x_j \mid j \in J\}$	(infinite distributivity)

where $\wedge$ is the binary meet and $\top$ is the 0-ary meet. The category of algebras of the above algebraic theory is equivalent to the category **Frm**.

First we construct the *free frame* over a set X. Let $Fr(X)$ denote the set of all lower-closed subsets of $(\mathcal{P}_{fin}(X), \supseteq)$ ordered by subset inclusion, where $\mathcal{P}_{fin}(X)$ is the collection of all finite subsets of X. The poset $Fr(X)$ is a complete lattice with arbitrary intersections as arbitrary joins and finite unions as finite meets. Hence the equations (i) - (v) hold. Because $Fr(X)$ is a sub-lattice of $\mathcal{P}(\mathcal{P}_{fin}(X))$, the infinite distributivity law (vi) also holds.

The set X can be mapped into the frame $Fr(X)$ by the function $\eta_X : X \to Fr(X)$ defined, for every every $x \in X$, by

$$\eta_X(x) \quad = \quad \{A \subseteq_{fin} X \mid x \in A\}.$$

The above construction is universal in the following sense.

9.1.1. Theorem. *Let X be a set and F be an arbitrary frame. For any function $f : X \to F$ there exists a unique frame morphism $f^\sharp : Fr(X) \to F$ such that $f^\sharp \circ \eta_X = f$.*

$$\begin{tikzcd}
X \arrow[r, "\eta_X"] \arrow[dr, "f"'] & Fr(X) \arrow[d, "f^\sharp"] \\
& F
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
Fr(X) \arrow[d, dashed, "f^\sharp"] \\
F
\end{tikzcd}$$

Proof: For every element I of $Fr(X)$, it holds that

$$I = \bigcup\{\bigcap\{\eta_X(x) \mid x \in A\} \mid A \in I\}.$$

Since $f^\sharp : Fr(X) \to F$ must preserve arbitrary joins and finite meets, and $f^\sharp \circ \eta_X = f$, its only possible definition is given, for $I \in Fr(X)$, by

$$f^\sharp(I) = \bigvee_{A \in I} \bigwedge_{x \in A} f(x).$$

The function $f^\sharp$ preserves all joins and all finite meets [Joh82a, Lemma 4.4 and Theorem 1.2]. □

By Proposition 2.1.1 it follows that the assignment $X \mapsto Fr(X)$ can be extended to a functor $Fr : \mathbf{Set} \to \mathbf{Frm}$ which is a left adjoint to the forgetful functor $\mathbf{Frm} \to \mathbf{Set}$. By Proposition 2.1.3, $\mathbf{Frm}$ has all small limits and all small colimits.

Next we come to the presentation of frames. For a set G of generators, an expression e formed from generators in G by applying the frame operators is said to be in *frame-normal form* if

$$e = \bigvee_I \bigwedge_J g_{i,j}, \tag{9.1}$$

for some set I, finite set J and $g_{i,j} \in G$. An expression formed from elements of G by applying the frame operators can always be reduced to an equivalent expression in frame normal form using the equations of the algebra of frames.

9.1.2. Corollary. *In the theory of frames, any presentation $Fr\langle G \mid R\rangle$ by a set of generators G and a set of relations R presents a frame.*

Proof: Construct the free frame $Fr(G)$ and assume, without loss of generality, that every expression in the relations of R is in frame normal form. There are two functions $f_l, f_r : R \to Fr(G)$ defined by

$$f_l(\langle e_l, e_r\rangle) = \bigcup_I \bigcap_J \eta_G(g_{i,j})$$

for $e_l = \bigvee_I \bigwedge_J g_{i,j}$, and similarly for f_r. By Theorem 9.1.1 we can extend the above functions to the frame morphisms $f_l^\sharp, f_r^\sharp : Fr(R) \to Fr(G)$. Let F be the coequalizer in $\mathbf{Frm}$ of this diagram and let $h : Fr(G) \to F$ be the canonical frame morphism associated with this colimit. The frame F together with the function $h \circ \eta_G : G \to F$ is a model for the presentation, and by universality of the construction, F presents $Fr\langle G \mid R\rangle$. □

Also the converse of the above corollary holds: every frame F has a presentation. For each $x \in F$, let $\widehat{x}$ be a generator, and take as relations

$$\langle \bigwedge_{x \in S} \widehat{x}, \widehat{\bigwedge_{x \in S} x}\rangle \quad \text{for every finite subset } S \text{ of } F$$

$$\langle \bigvee_{x \in S} \widehat{x}, \widehat{\bigvee_{x \in S} x}\rangle \qquad \text{for every subset } S \text{ of } F.$$

Notice that for each frame we use only set-many relations.

Presenting completely distributive lattices

Next we turn to the theory of completely distributive lattices . It has a proper class of operators, $\bigsqcap_I$ (the I-ary meet for each set I) and $\bigsqcup_J$ (the J-ary join for each set J), and a proper class of equations:

(i)	$x \sqcap y = y \sqcap x$	(commutativity)
(ii)	$x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z$	(associativity)
(iii)	$x \sqcap x = x$	(idempotency)
(iv)	$x \sqcap \top = x$	(unit)
(v)	$x \sqcap \bigsqcap\{x_i \mid i \in I\} = \bigsqcap\{x \sqcap x_i \mid i \in I\}$	(meet absorption)
(vi)	$x_k \sqcap \bigsqcup\{x_j \mid j \in J\} = x_k$ for $k \in J$	(join absorption)
(vii)	$\bigsqcap\{\bigsqcup\{x_{i,j} \mid j \in J_i\} \mid i \in I\} =$ $\bigsqcup\{\bigsqcap\{f(i) \mid i \in I\} \mid f \in \Phi(\mathcal{I})\}$	(complete distributivity)

where $\sqcap$ is the binary meet, $\top$ is the 0-ary meet, and $\mathcal{I} = \{\{x_{i,j} \mid j \in J_i\} \mid i \in I\}$. Recall, from Section 2.2 that $\Phi(\mathcal{I})$ denotes the set of all functions $f : \mathcal{I} \to \bigcup \mathcal{I}$ with $f(i)$ in the set $\{x_{i,j} \mid j \in J_i\}$ for all $i \in I$. From the above equations it follows that every completely distributive lattice is also a frame. The category of algebras of the above theory is equivalent to the category **CDL** of completely distributive lattices together with functions preserving arbitrary meets and arbitrary joins.

As before, we begin by constructing the free completely distributive lattice over a set. The construction we present is similar to the free frame construction and differs only slightly from the construction presented (without proof) in [Mar79]. Recall from Definition 4.3.1 that for a set X, $CDL(X)$ denotes the collection of all lower-closed subsets of the poset $(\mathcal{P}(X), \supseteq)$ ordered by subset inclusion. Since $CDL(X)$ is closed under arbitrary unions and arbitrary intersections, it is a complete sub-lattice of $\mathcal{P}(\mathcal{P}(X))$. Hence $CDL(X)$ is a completely distributive lattice.

The set X can be mapped into $CDL(X)$ by the function $\theta_X : X \to CDL(X)$ defined by

$$\theta_X(x) \quad = \quad \{A \subseteq X \mid x \in A\},$$

for every $x \in X$. The above construction is universal.

9.1.3. Theorem. *Let X be a set and L be a completely distributive lattice. For any function $f : X \to L$ there exists a unique morphism $f^\dagger : CDL(X) \to L$ in* **CDL** *such that $f^\dagger \circ \theta_X = f$.*

$$\begin{tikzcd} X \arrow[r, "\theta_X"] \arrow[dr, "f"'] & CDL(X) \arrow[d, "f^\dagger"] \\ & L \end{tikzcd} \qquad \begin{tikzcd} CDL(X) \arrow[d, dashed, "f^\dagger"] \\ L \end{tikzcd}$$

Proof: For every element J in $CDL(X)$, it holds

$$J = \bigcup\{\bigcap\{\theta_X(x) \mid x \in A\} \mid A \in J\}.$$

Since $f^\dagger : CDL(X) \to L$ preserves arbitrary joins and arbitrary meets, and $f^\dagger \circ \theta_X = f$, its only possible definition is given, for $J \in CDL(X)$, by

$$f^\dagger(J) = \bigsqcup\{\bigsqcap\{f(x) \mid x \in A\} \mid A \in J\}.$$

From the form of the above definition it follows that $f^\dagger$ preserves arbitrary joins. So it remains to prove that $f^\dagger$ preserves all meets. Let $J_i \in CDL(X)$ for all i in an arbitrary set I, and let $\phi : \mathcal{P}(X) \to L$ be the function mapping every subset A of X to $\bigsqcap\{f(x) \mid x \in A\}$. It is not hard to see that ϕ preserves arbitrary meets. Moreover $f^\dagger(J) = \bigsqcup\{\phi(A) \mid A \in J\}$. We have

$$\begin{aligned}
\bigsqcap\{f^\dagger(J_i) \mid i \in I\} &= \bigsqcap\{\bigsqcup\{\phi(A) \mid A \in J_i\} \mid i \in I\} \\
&= \bigsqcup\{\bigsqcap\{\phi(g(i)) \mid i \in I\} \mid g \in \Phi(I)\} \quad \text{[complete distributivity]} \\
&= \bigsqcup\{\phi(\bigsqcap\{g(i) \mid i \in I\}) \mid g \in \Phi(I)\} \quad [\phi \text{ preserves meets}] \\
&= \bigsqcup\{\phi(A) \mid A \in \bigcap\{J_i \mid i \in I\}\} \quad [\text{all } J_i\text{'s are lower sets}] \\
&= f^\dagger(\bigcap\{J_i \mid i \in I\}),
\end{aligned}$$

where $\Phi(I)$ is the set of all functions $g : I \to \bigcup_I J_i$ such that $g(i) \in J_i$. $\square$

By Proposition 2.1.1 the assignment $X \mapsto CDL(X)$ can be extended to a functor $CDL : \mathbf{Set} \to \mathbf{CDL}$ which is a left adjoint to the forgetful functor $\mathbf{CDL} \to \mathbf{Set}$. Moreover, by Proposition 2.1.3, **CDL** has all small limits and all small colimits.

For a set G of generators, an expression e formed from generators in G by applying the operators of the theory of completely distributive lattices is said to be in *cdl-normal form* if

$$e = \bigsqcup_I \bigsqcap_J g_{i,j}, \tag{9.2}$$

for some sets I and J, and generators $g_{i,j} \in G$. Every expression formed from elements of G by applying the operators of the theory of completely distributive lattices can always be reduced to an equivalent expression in cdl-normal form by using the equations of the algebra of frames and the dual completely distributive law as given in Section 2.2.

9.1.4. Corollary. *In the theory of completely distributive lattices, any presentation $CDL\langle G \mid R\rangle$ by a set of generators G and a set of relations R presents a completely distributive lattice.*

Proof: Similar to that of Corollary 9.1.2. $\square$

As for the case of frames, also the converse of the above corollary holds: every completely distributive lattice L has a presentation. For each x in L, let $\widehat{x}$ be a generator, and take as relations

$$\begin{array}{ll} \langle \bigsqcap_{x\in S} \widehat{x}, \widehat{\bigsqcap_{x\in S} x}\rangle & \text{for every subset } S \text{ of } L \\ \langle \bigsqcup_{x\in S} \widehat{x}, \widehat{\bigsqcup_{x\in S} x}\rangle & \text{for every subset } S \text{ of } L. \end{array}$$

Notice that also for completely distributive lattices we use set-many relations.

We conclude this section by constructing the free completely distributive lattice over a frame, using the results on the presentation of frames and completely distributive lattices.

9.1.5. Theorem. *The forgetful functor* **CDL** $\rightarrow$ **Frm** *has a left adjoint denoted by* $\overline{(\cdot)}$: **Frm** $\rightarrow$ **CDL**. *Moreover, for every frame F, the unit of the adjunction, $\zeta_F : F \rightarrow \overline{F}$ defines an observation frame.*

Proof: Let F be a frame presented by $Fr\langle G \mid R\rangle$ and let $[\![-]\!]^F_G : G \rightarrow F$ be its associated function. Say $\overline{F}$ is the completely distributive lattice which presents $CDL\langle G|R\rangle$ (notice that we have used the same generators and relations as in the frame presentation of F). Because $\overline{F}$ is a model for $CDL\langle G|R\rangle$, it comes equipped with a canonical function $[\![-]\!]^{\overline{F}}_G : G \rightarrow \overline{F}$. Since $\overline{F}$ is also a frame and, by construction a model for $Fr\langle G \mid R\rangle$, Corollary 9.1.2 gives us a unique frame morphism $\zeta_F : F \rightarrow \overline{F}$ such that $\zeta_F([\![g]\!]^F_G) = [\![g]\!]^{\overline{F}}_G$ for all generators g in G, i.e. ζ_F maps each generator of F to the corresponding generator of $\overline{F}$. The morphism ζ_F is the unit of the adjunction between the category of frames and that of the completely distributive lattices.

Moreover, for every other completely distributive lattice L, if $f : F \rightarrow L$ is a frame morphism then L together with the function $f \circ [\![-]\!]^F_G : G \rightarrow L$ is a model of $Fr\langle G \mid R\rangle$ and also of $CDL\langle G \mid R\rangle$ (only finite meets are present in the relations of R). By Corollary 9.1.4, there exists a unique morphism $h : \overline{F} \rightarrow L$ in **CDL** such that $h([\![g]\!]^{\overline{F}}_G) = f([\![g]\!]^F_G)$ for all generators $g \in G$. Since $[\![-]\!]^{\overline{F}}_G = \zeta_F([\![-]\!]^F_G)$, f and ζ_F are frame morphisms, and every element in F is the join of finite meets of elements in $[\![G]\!]^F_G$, we have that $h : \overline{F} \rightarrow L$ is the unique morphism in **CDL** such that $h \circ \zeta_F = f$. Thus, by Proposition 2.1.1, the first statement of the theorem follows.

Next we prove that the frame morphism $\zeta_F : F \rightarrow \overline{F}$ mapping each generator of F to the corresponding generator of $\overline{F}$, defines an observation frame. By construction, each element in $\overline{F}$ is the join of the meets of elements in $[\![G]\!]^{\overline{F}}_G$. Using the dual of the complete distributive law (which holds for every completely distributive lattice [Ran52]) we obtain that, for every element $x \in \overline{F}$,

$$x = \bigsqcap_I \bigsqcup_J [\![g_{i,j}]\!]^{\overline{F}}_G$$

for some sets I and J, and generators $g_{i,j}$ of F. But $[\![g_{i,j}]\!]_G^{\overline{F}} = \zeta_F([\![g_{i,j}]\!]_G^F)$, the unit ζ_F preserves arbitrary joins, and for all $i \in I$, $\bigvee_J [\![g_{i,j}]\!]_G^F$ is an element in F. Hence every element in $\overline{F}$ is the meet of elements in $\zeta_F(F)$, that is, by Proposition 8.1.16, $\zeta_F(F)$ is order generating $\overline{F}$. By applying Corollary 8.1.17 we conclude that $\zeta_F : F \to \overline{F}$ is an observation frame. □

Free objects are unique up to isomorphism, and hence the above theorem implies that $\overline{Fr(X)}$ is isomorphic in the category **CDL** to $CDL(X)$ for every set X. But $\zeta_{Fr(X)} : Fr(X) \to \overline{Fr(X)}$ is an observation frame, hence also the unique frame morphism $\theta_X^\sharp : Fr(X) \to CDL(X)$ such that $\theta_X^\sharp \circ \eta_X = \theta_X$ (given in Theorem 9.1.1) defines an observation frame.

9.2 Observation frames as algebras

Using the results of the previous section, we are now in a position to view observation frames as algebras. Since an observation frame is a function between two ordinary algebras (frames and completely distributive lattices, respectively), we expect it to be an algebra over a function between sets.

Let $f : X \to Y$ be a function in **Set**. By Theorem 9.1.1 and because every completely distributive lattice is a frame there exists a unique frame morphism $(\theta_Y \circ f)^\sharp : Fr(X) \to CDL(Y)$ such that $(\theta_Y \circ f)^\sharp \circ \eta_X = \theta_Y \circ f$. Let $f^\odot$ denote $(\theta_Y \circ f)^\sharp$. Using the definition of θ_Y and Theorem 9.1.1, $f^\odot : Fr(X) \to CDL(Y)$ can be characterized directly, for all $I \in Fr(X)$, by

$$f^\odot(I) = \bigcup_{A \in I} \bigcap_{x \in A} \{B \subseteq Y \mid f(x) \in B\}. \tag{9.3}$$

In general, $f^\odot : Fr(X) \to CDL(Y)$ will not be an observation frame, as can be seen by taking, for example, f to be the inclusion of a one-element set X into a two-element set Y. The following lemma gives a necessary condition on $f : X \to Y$ in order that $f^\odot : Fr(X) \to CDL(Y)$ be an observation frame.

9.2.1. Lemma. *Let $f : X \to Y$ be a function between two sets X and Y. If f is onto then the function $f^\odot : Fr(X) \to CDL(Y)$ is an observation frame.*

Proof: Because $f^\odot$ is a frame morphism, we only need that every element in $CDL(Y)$ is the intersection of elements in the image under $f^\odot$ of $Fr(X)$. Since $\theta_Y^\sharp : Fr(Y) \to CDL(Y)$ is an observation frame, and $Fr(Y) = Fr(f(X))$, f being onto, every element in $CDL(Y)$ is the meet of elements in $\theta_Y^\sharp(Fr(f(X)))$. Since $Fr(f(X))$ is the free frame over $Y = f(X)$, every element in $Fr(f(X))$ is the join of finite meets of elements in $\eta_Y(f(X))$, where $\eta_Y : Y \to Fr(f(X))$ is the unit of the adjunction in Theorem 9.1.1. But $\theta_Y^\sharp$ preserves arbitrary joins and finite meets, and, for all $x \in X$,

$$\begin{aligned}
\theta^\sharp_Y(\eta_Y(f(x))) &= \theta_Y(f(x)) \quad [\text{defining property of } \theta^\sharp_Y] \\
&= (\theta_Y \circ f)^\sharp(\eta_X(x)) \quad [\text{defining property of } (\theta_Y \circ f)^\sharp] \\
&= f^\odot(\eta_X(x)). \quad [\text{definition of } f^\odot]
\end{aligned}$$

Since every element in $Fr(f(X))$ is the join of finite meets of elements in $\eta_Y(f(X))$ and $f^\odot$ preserves both arbitrary joins and finite meets, we have that $\theta^\sharp_Y(Fr(f(X))) = f^\odot(Fr(X))$. Therefore every element in $CDL(Y)$ is the meet of elements in $f^\odot(Fr(X))$. By Corollary 8.1.17, it follows that $f^\odot : Fr(X) \to CDL(Y)$ is an observation frame. $\square$

Consider now the full sub-category of $\mathbf{Set^2}$ whose objects are onto functions, and denote it by $\mathbf{Set}^{epi}$. Thus objects in $\mathbf{Set}^{epi}$ are onto functions $f : X \to Y$ between sets, and morphisms from $f : X \to Y$ to $g : X' \to Y'$ are pairs of functions $\langle h : X \to X', k : Y \to Y' \rangle$ such that $k \circ f = g \circ h$. Composition is defined componentwise. The category $\mathbf{Set}^{epi}$ has all small limits and all small colimits (which are constructed as in category $\mathbf{Set^2}$). There is a functor $U : \mathbf{OFrm} \to \mathbf{Set}^{epi}$ mapping every observation frame $\alpha : F \to L$ to the onto function $U(\alpha) = \alpha : F \to \alpha(F)$ (the co-restriction of α to its image). The functor U is defined on morphisms as follows:

$$U(\phi) \;=\; \langle \phi : F \to G, \widetilde{\phi} : \alpha(F) \to \beta(G) \rangle,$$

where ϕ is a morphism from the observation frame $\alpha : F \to L$ to the observation frame $\beta : G \to H$, and $\widetilde{\phi} : \alpha(F) \to \beta(G)$ is the restriction and co-restriction of the unique morphism form L to G in $\mathbf{CDL}$ such that $\widetilde{\phi} \circ \alpha = \beta \circ \phi$ (see Theorem 8.1.4). By the above commutativity, $U(\phi)$ is well-defined.

9.2.2. Theorem. *Assume that $f : X \to Y$ is an onto function between the sets X and Y and let $\beta : G \to H$ be an observation frame. For every morphism $\langle h : X \to G, k : Y \to \beta(G) \rangle$ in $\mathbf{Set}^{epi}$ from f to $U(\beta)$, the frame morphism $h^\sharp : Fr(X) \to G$ is the only morphism in $\mathbf{OFrm}$ from $f^\odot$ to β such that $U(h^\sharp) \circ \langle \eta_X, \theta_Y \rangle = \langle h, k \rangle$ in $\mathbf{Set}^{epi}$.*

$$\begin{tikzcd}
 & Y \arrow[r, "\theta_Y"] \arrow[dr, "k"] & f^\odot(Fr(X)) \arrow[d, "\widetilde{h^\sharp}"] \\
X \arrow[ur, "f"] \arrow[r, "\eta_X"] \arrow[dr, "h"'] & Fr(X) \arrow[ur, "f^\odot"] \arrow[d, "h^\sharp"] & \beta(G) \\
 & G \arrow[ur, "\beta"] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & CDL(Y) \arrow[d, "\widetilde{h^\sharp}"] \\
Fr(X) \arrow[ur, "f^\odot"] \arrow[d, "h^\sharp"] & H \\
G \arrow[ur, "\beta"] &
\end{tikzcd}$$

Proof: By Theorem 9.1.1, $h = h^\sharp \circ \eta_X$. Hence $\beta \circ h^\sharp \circ \eta_X = \beta \circ h$. Also $k^\dagger \circ f^\odot \circ \eta_X = \beta \circ h$ because

$$\begin{aligned} k^\dagger \circ f^\odot \circ \eta_X &= k^\dagger \circ (\theta_Y \circ f)^\sharp \circ \eta_X \quad [\text{definition of } f^\odot] \\ &= k^\dagger \circ \theta_Y \circ f \quad [\text{by Theorem 9.1.1 } (\theta_Y \circ f)^\sharp \circ \eta_X = \theta_Y \circ f] \\ &= k \circ f \quad [\text{by Theorem 9.1.3 } k^\dagger \circ \theta_Y = k] \\ &= \beta \circ h. \quad [\langle h, k\rangle \text{ is a morphism in } \mathbf{Set}^{epi}] \end{aligned}$$

Therefore, by Theorem 9.1.1, $\beta \circ h^\sharp = k^\dagger \circ f^\odot$. Using Theorem 8.1.4 it follows that $h^\sharp$ is a morphism in **OFrm** from $f^\odot$ to β with $\widetilde{h^\sharp} = k^\dagger$. Moreover, by Theorem 9.1.1 and Theorem 9.1.3,

$$\begin{aligned} U(h^\sharp) \circ \langle \eta_X, \theta_Y\rangle &= \langle h^\sharp, \widetilde{h^\sharp}\rangle \circ \langle \eta_X, \theta_Y\rangle \\ &= \langle h^\sharp, k^\dagger\rangle \circ \langle \eta_X, \theta_Y\rangle \\ &= \langle h^\sharp \circ \eta_X, k^\dagger \circ \theta_Y\rangle \\ &= \langle h, k\rangle. \end{aligned}$$

Uniqueness of $h^\sharp$ follows immediately from Theorem 9.1.1, Theorem 9.1.3 and Theorem 8.1.4. □

The above theorem and Proposition 2.1.1 imply that the assignment

$$(f : X \to Y) \mapsto (f^\odot : Fr(X) \to CDL(Y))$$

extends to a functor from $\mathbf{Set}^{epi}$ to **OFrm** which is a left adjoint to the functor $U : \mathbf{OFrm} \to \mathbf{Set}^{epi}$. Next we want to prove that the functor U is monadic. As a consequence we have that the category of observation frames **OFrm** is equivalent to the category of algebras induced by the monad $U \circ (-)^\odot$. We need the following three lemmas.

9.2.3. Lemma. *The functor $U : \mathbf{OFrm} \to \mathbf{Set}^{epi}$ reflects isomorphisms.*

Proof: Assume that $\alpha : F \to L$ and $\beta : G \to H$ are observation frames such that $U(\alpha)$ is isomorphic to $U(\beta)$ in $\mathbf{Set}^{epi}$. Then $F \cong G$ and $\alpha(F) \cong \beta(G)$. Since every element in L is the meet of elements in $\alpha(F)$, the isomorphism $\alpha(F) \cong \beta(G)$ can be extended to an isomorphism between L and G. Hence F and G are isomorphic as frames while L and G are isomorphic as completely distributive lattices, that is α and β are isomorphic as observation frames. □

9.2.4. Lemma. *The category* **OFrm** *has all coequalizers.*

Proof: Assume that ϕ_1 and ϕ_2 are morphisms in **OFrm** from the observation frame $\alpha : F \to L$ to the observation frame $\beta : G \to H$. By Theorem 8.1.4, $\phi_1 : F \to G$ induces uniquely a morphism $\widetilde{\phi_1} : L \to H$ between completely distributive lattices such that $\widetilde{\phi_1} \circ \alpha = \beta \circ \phi_1$. Similarly, $\phi_2 : F \to G$ induces uniquely a morphism $\widetilde{\phi_2} : L \to H$ between completely distributive lattices such that $\widetilde{\phi_2} \circ \alpha = \beta \circ \phi_2$.

Let $(G', c_1 : G \to G')$ be the coequalizer in **Frm** of ϕ_1 and ϕ_2, and let $(H', c_2 : H \to H')$ be the coequalizer in **CDL** of $\widetilde{\phi_1}$ and $\widetilde{\phi_2}$. Notice that

$$c_2 \circ \beta \circ \phi_1 = c_2 \circ \widetilde{\phi_1} \circ \alpha = c_2 \circ \widetilde{\phi_2} \circ \alpha = c_2 \circ \beta \circ \phi_2.$$

By the defining property of coequalizers, there exists a unique frame morphism $\gamma : G' \to H'$ such that $\gamma \circ c_1 = c_2 \circ \beta$. We claim γ is an observation frame. Indeed, since $q = \bigsqcap\{\beta(x) \mid q \sqsubseteq \beta(x)\}$ for all $q \in H$ and c_2 preserves arbitrary meets, we have

$$\begin{array}{rcll} c_2(q) & = & \bigsqcap\{c_2(\beta(x)) \mid q \sqsubseteq \beta(x)\} & \\ & \sqsupseteq & \bigsqcap\{c_2(\beta(x)) \mid c_2(q) \sqsubseteq c_2(\beta(x))\} & [c_2 \text{ is monotone}] \\ & = & \bigsqcap\{\gamma(c_1(x)) \mid c_2(q) \sqsubseteq \gamma(c_1(x))\}. & [\text{defining property of } \gamma] \end{array}$$

The reverse of the above inequality holds by the defining property of meets. Hence $\gamma : G' \to H'$ is an observation frame. Notice that $\gamma \circ c_1 = c_2 \circ \beta$ implies $c_2 = \widetilde{c_1}$ by Theorem 8.1.4. One can now easily verify that γ is indeed the coequalizer in **OFrm** of ϕ_1 and ϕ_2. □

9.2.5. Lemma. *The functor $U : \mathbf{OFrm} \to \mathbf{Set}^{epi}$ preserves all coequalizers*

Proof: Assume ϕ_1 and ϕ_2 are two morphisms in **OFrm** from the observation frame $\alpha : F \to L$ to the observation frame $\beta : G \to H$. By Theorem 8.1.4, $\phi_1 : F \to G$ induces uniquely a morphism $\widetilde{\phi_1} : L \to H$ between completely distributive lattices such that $\widetilde{\phi_1} \circ \alpha = \beta \circ \phi_1$. Similarly, $\phi_2 : F \to G$ induces uniquely a morphism $\widetilde{\phi_2} : L \to H$ between completely distributive lattices such that $\widetilde{\phi_2} \circ \alpha = \beta \circ \phi_2$.

Let $(\gamma : G' \to H', c_1 : G \to G')$ be the coequalizer in **OFrm** of ϕ_1 and ϕ_2 as described in Lemma 9.2.4. Then $(G', c_1 : G \to G')$ is the coequalizer in **Frm** of ϕ_1 and ϕ_2, and $(H', c_2 : H \to H')$ is the coequalizer in **CDL** of $\widetilde{\phi_1}$ and $\widetilde{\phi_2}$, where c_2 is the unique morphism in **CDL** such that $\gamma \circ c_1 = c_2 \circ \beta$. We need to prove that $(U(\gamma), U(c_1) = \langle c_1, c_2\rangle)$ is the coequalizer in $\mathbf{Set}^{epi}$ of $U(\phi_1) = \langle \phi_1, \widetilde{\phi_1}\rangle$ and $U(\phi_2) = \langle \phi_2, \widetilde{\phi_2}\rangle$.

Assume there is an object $f : X \to Y$ in $\mathbf{Set}^{epi}$ and a morphism $\langle h_1, h_2\rangle$ from $U(\beta)$ to $f : X \to Y$ such that $U(\phi_1) \circ \langle h_1, k\rangle = U(\phi_2) \circ \langle h_1, h_2\rangle$. Define the functions $k_1 : G' \to X$ and $k_2 : \gamma(G') \to Y$ as follows:

$$k_1(c_1(x)) = h_1(x) \quad \text{and} \quad k_2(c_2(y)) = h_2(y)$$

for all $x \in G$ and $y \in \beta(G)$. By the standard theory on congruences, the above maps are well-defined and they are the unique ones such that $k_1 \circ c_1 = h_1$ and

$k_2 \circ c_2 = h_2$. It remains only to prove that $\langle k_1, k_2 \rangle$ is a morphism in $\mathbf{Set}^{epi}$, that is, $k_2 \circ \gamma = f \circ k_1$. For all $c_1(x) \in G'$,

$$\begin{aligned} k_2(\gamma(c_1(x))) &= k_2(c_2(\beta(x)) \\ &= h_2(\beta(x)) \quad [\text{definition } k_2] \\ &= f(h_1(x)) \quad [\langle h, k \rangle \text{ is morphism in } \mathbf{Set}^{epi}] \\ &= f(k_1(c_1(x))). \quad [\text{definition } k_1] \end{aligned}$$

□

9.2.6. Theorem. *The functor $U : \mathbf{OFrm} \to \mathbf{Set}^{epi}$ is monadic.*

Proof: Apply Proposition 2.1.2 using the above Lemmas 9.2.3, 9.2.4, and 9.2.5. □

Since $\mathbf{Set}^{epi}$ has small limits and monadic functors create limits [ML71], the category **OFrm** has all small limits too.

Presenting observation frames

Consider a frame $F = Fr\langle G_1 \mid R_1 \rangle$ and a completely distributive lattice $L = CDL\langle G_2 \mid R_2 \rangle$. Let

$$[\![-]\!]_{G_1} : G_1 \to F \quad \text{and} \quad [\![-]\!]_{G_2} : G_2 \to L$$

be the two canonical embeddings of the generators into F and L, respectively. Assume there exists an onto function $f : G_1 \to G_2$ between the generators of the two presentations. If the relations in R_1 and R_2 (in their respective normal forms) satisfy the following commutativity property

$$\langle \bigvee_I \bigwedge_J g_i^j, \bigvee_N \bigwedge_M g_n^m \rangle \in R_1 \quad \Rightarrow \quad \langle \bigsqcup_I \bigsqcap_J f(g_i^j), \bigsqcup_N \bigsqcap_M f(g_n^m) \rangle \in R_2, \tag{9.4}$$

then the function $[\![-]\!]_{G_2} \circ f : G_1 \to L$ makes L a model of the frame presentation $Fr\langle G_1 \mid R_1 \rangle$. Hence, by Corollary 9.1.2, there exists a unique frame morphism $f^\bullet : F \to L$ such that, for all generators $g \in G_1$,

$$f^\bullet([\![g]\!]_{G_1}) = [\![f(g)]\!]_{G_2}.$$

In a way similar to the proof of Lemma 9.2.1, it is possible to show that every element in L is the meet of elements in $f^\bullet(F)$. Hence, by Corollary 8.1.17, $f^\bullet : F \to L$ is an observation frame.

Conversely, every observation frame $\alpha : F \to L$ can be 'presented' as follows. Consider the following two sets of generators

$$G_1 = \{\widehat{x} \mid x \in F\} \text{ and } G_2 = \{\widehat{\alpha(x)} \mid x \in F\},$$

and define the function $f: G_1 \to G_2$ by $f(\widehat{x}) = \widehat{\alpha(x)}$ for all $\widehat{x} \in G_1$. Clearly f is an onto function. Define also the sets of relations

$$\begin{aligned} R_1 &= \{\langle \bigwedge_{x\in S} \widehat{x}, \widehat{\bigwedge_{x\in S} x}\rangle \mid S \subseteq_{fin} F\} \cup \{\langle \bigvee_{x\in S} \widehat{x}, \widehat{\bigvee_{x\in S} x}\rangle \mid S \subseteq F\}, \\ R_2 &= \{\langle \bigsqcap\{\widehat{x} \mid x \in S\}, (\widehat{\bigsqcap S})\rangle \mid S \subseteq \alpha(F) \ \& \ \bigsqcap S \in \alpha(F)\} \\ & \quad \cup \{\langle \bigsqcup\{\widehat{x} \mid x \in S\}, (\widehat{\bigsqcup S})\rangle \mid S \subseteq \alpha(F) \ \& \ \bigsqcup S \in \alpha(F)\}. \end{aligned}$$

The frame F presents $Fr\langle G_1 \mid R_1\rangle$ and the completely distributive lattice L presents $CDL\langle G_2 \mid R_2\rangle$. By universality of the presentation of F it follows that $f^\bullet : F \to L$ is equal to $\alpha : F \to L$.

9.3 Frames and observation frames

In this section we take a closer look at the relationship between frames and observation frames. We start by relating the categories of frames, completely distributive lattices and observation frames.

There is a functor $Dom : \mathbf{OFrm} \to \mathbf{Frm}$ mapping an observation frame $\alpha : F \to L$ to $Dom(\alpha) = F$ and a morphism $\phi : (\alpha : F \to L) \to (\beta : G \to H)$ in $\mathbf{OFrm}$, to the frame morphism $Dom(\phi) = \phi : F \to G$.

9.3.1. Theorem. *The functor $Dom : \mathbf{OFrm} \to \mathbf{Frm}$ has a left adjoint.*

Proof: Let F be a frame and consider the observation frame $\zeta_F : F \to \overline{F}$ given in Theorem 9.1.5. By definition, $Dom(\zeta_F) = F$. Let now $\beta : G \to H$ be another observation frame together with a frame morphism $\phi : F \to Dom(\beta)$. The composition $\beta \circ \phi : F \to H$ is a frame morphism. Hence, by Theorem 9.1.5 there exists a unique morphism $\psi : \overline{F} \to H$ in $\mathbf{CDL}$ such that $\psi \circ \zeta_F = \beta \circ \phi$. By Theorem 8.1.4 this implies that ϕ is a morphism in $\mathbf{OFrm}$. Since $Dom(\phi) = \phi$, by Proposition 2.1.1 we have that the functor $Dom : \mathbf{OFrm} \to \mathbf{Frm}$ has a left adjoint. $\square$

Next we consider the relationship between observation frames and completely distributive lattices. Let $Cod : \mathbf{OFrm} \to \mathbf{CDL}$ be the functor which maps an observation frame $\alpha : F \to L$ to the completely distributive lattice $Cod(\alpha) = L$, and every morphism ϕ in $\mathbf{OFrm}$, between the observation frames $\alpha : F \to L$ and $\beta : G \to H$, to $Cod(\phi) = \widetilde{\phi}$, where $\widetilde{\phi} : L \to H$ is the unique morphism in $\mathbf{CDL}$ such that $\widetilde{\phi} \circ \alpha = \beta \circ \phi$ given in Theorem 8.1.4.

9.3.2. Theorem. *The functor $Cod : \mathbf{OFrm} \to \mathbf{CDL}$ has a right adjoint.*

Proof: For any completely distributive lattice L consider the observation frame $id_L : L \to L$. If $\beta : G \to H$ is another observation frame and $\psi : H \to L$ is a morphism in $\mathbf{CDL}$, then $\phi = \psi \circ \beta : G \to L$ is the unique frame morphism such that $id_L \circ \phi = \psi \circ \beta$. Hence, by Theorem 8.1.4, ϕ is a morphism in **OFrm** and $Cod(\phi) = \widetilde{\phi} = \psi$. From the dual of Proposition 2.1.1 it follows that the functor $Cod : \mathbf{OFrm} \to \mathbf{CDL}$ has a right adjoint. □

It is not difficult to see that an element p of a completely distributive lattice L is completely prime if and only if p is an M-prime element of the observation frame $id_L : L \to L$. By Lemma 8.3.4 it follows that $id_L : L \to L$ is spatial if and only if L is isomorphic in **CDL** to a complete ring of sets.

Filters and M-filters

In this section we investigate, for an observation frame $\alpha : F \to L$, some of the relationships between elements of the completely distributive lattice L, M-filters of the observation frame α and ordinary filters of the frame F. We will use these relationships for a characterization of sober spaces.

We start by showing that points are preserved by the construction of the free observation frame over a frame in which a point in a frame F is a completely prime filter of F, whereas by Lemma 8.1.8 a point in an observation frame $\alpha : F \to L$ is a completely prime M-filter of α.

9.3.3. Theorem. *The collection of all completely prime filters of a frame F is isomorphic to the collection of all completely prime M-filters of the free observation frame $\zeta_F : F \to \overline{F}$.*

Proof: Let $2 = \{\top, \bot\}$ be the two point completely distributive lattice with $\bot \sqsubseteq \top$, and write $\mathbf{2}$ for the observation frame $id_2 : 2 \to 2$. Let $CPF(F)$ be the set of all completely prime filters of F and $CPMF(\zeta_F)$ the set of all completely prime M-filters of $\zeta_F : F \to \overline{F}$. By Lemma 8.1.8, $\mathbf{OFrm}(\zeta_F, \mathbf{2}) \cong CPMF(\zeta_F)$, whereas $\mathbf{Frm}(F, 2) \cong CPF(F)$ (see [Vic89, Proposition 5.4.7] for the latter isomorphism). But $Dom(\mathbf{2}) = 2$ and ζ_F is the free observation frame over F by Theorem 9.3.1. Hence $\mathbf{Frm}(F, 2) \cong \mathbf{OFrm}(\zeta_F, \mathbf{2})$. □

A frame F is called *spatial*, or said to have enough points, if for all x and y in F, if $x \not\leq y$ then there exists a completely prime filter $\mathcal{F}$ of F such that $a \in \mathcal{F}$ and $b \notin \mathcal{F}$. The full sub-category of **Frm** whose objects are spatial frames F is denoted by **SFrm**. If X is any topological space then its lattice of open sets $\mathcal{O}(X)$ is a spatial frame, since we can take $\mathcal{F}$ to be the completely prime filter $\mathcal{F}_x = \{o \in \mathcal{O}(X) \mid x \in o\}$. The following corollary is an immediate consequence of the above theorem.

9.3.4. Corollary. *A frame F is spatial if and only if the observation frame $\zeta_F : F \to \overline{F}$ is spatial. Hence the adjunction of Theorem 9.3.1 restricts to an adjunction between the category of spatial frames* **SFrm** *and the category of spatial observation frames* **SOFrm**. □

In Definition 5.2.3 we have seen that a topological space X is sober if the assignment $x \mapsto \mathcal{F}_x$, for all $x \in X$, defines an isomorphism between X and $CPF(\mathcal{O}(X))$. Given a frame F, we can construct a sober space $Pt_\omega(F)$ by taking the set of all completely prime filters of F, denoted by $CPF(F)$, together with the collection of open sets

$$\mathcal{F}(a) = \{\mathcal{F} \in CPF(F) \mid a \in \mathcal{F}\},$$

for every $a \in F$. This collection forms a topology on $CPF(F)$. The following proposition can be found in [Joh82a, II, Corollary 1.7].

9.3.5. Proposition. *The assignment $F \mapsto Pt_\omega(F)$ defines a functor $\mathbf{Frm}^{op} \to \mathbf{Sp}$ which is a right adjoint of $\mathcal{O}(-) : \mathbf{Sp} \to \mathbf{Frm}^{op}$ (the functor which maps every topological space to its lattice of open sets and every continuous function to its inverse restricted to the open sets). Furthermore we have that*

(i) *the adjunction restricts to a duality between the $\mathbf{SFrm}$ and $\mathbf{Sob}$;*
(ii) *the inclusion $\mathbf{Sob} \hookrightarrow \mathbf{Sp}_0$ has left adjoint $Pt_\omega(\mathcal{O}(-))$;*
(iii) *the inclusion $\mathbf{SFrm} \hookrightarrow \mathbf{Frm}$ has left adjoint $\mathcal{O}(Pt_\omega^{op}(-))^{op}$.* □

The *soberification* of a topological space X is the sober space $Pt_\omega(\mathcal{O}(X))$. Notice that the process of soberification of a topological space X can be replaced by the process of constructing the topological space which best approximates the 'frame part' of the observation frame associated with X, that is,

$$Pt_\omega(\mathcal{O}(X)) = Pt_\omega(Dom(\Omega(X)).$$

Since adjoints are defined uniquely (up to natural isomorphisms), the above implies that the two squares below commute.

$$\begin{tikzcd}
\mathbf{Sp}_0 \arrow[d, shift right=2, "Pt_\omega(\mathcal{O}(-))"'] \arrow[d, phantom, "\dashv"] & \simeq & \mathbf{SOFrm}^{op} \arrow[d, shift right=2, "Dom^{op}"'] \arrow[d, phantom, "\dashv"] \\
\mathbf{Sob} \arrow[u, hook, shift right=2, "i"'] & \simeq & \mathbf{SFrm}^{op} \arrow[u, shift right=2, "\overline{(\cdot)}^{op}"']
\end{tikzcd}$$

The functor $Dom : \mathbf{OFrm} \to \mathbf{Frm}$ can therefore be considered as the pointless soberification of an abstract topological space. Next we use the above results (which generalize Lemma 8.3.6) for a characterization of sober spaces.

9.3.6. Theorem. *A $\mathcal{T}_0$ space X is sober if and only if the completely distributive lattice of saturated sets $\mathcal{Q}(X)$ is free over the frame of open sets $\mathcal{O}(X)$.*

Proof: Assume X is a sober space. By the commutativity of the above diagram it follows that the observation frames $\Omega(X){:}\mathcal{O}(X) \to \mathcal{Q}(X)$ and $\zeta_{\mathcal{O}(X)}{:}\mathcal{O}(X) \to \overline{\mathcal{O}(X)}$ are isomorphic in $\mathbf{OFrm}$. Hence $\overline{\mathcal{O}(X)}$ is isomorphic to $\mathcal{Q}(X)$ in $\mathbf{CDL}$.

But $\overline{\mathcal{O}(X)}$ is the free completely distributive lattice over the frame $\mathcal{O}(X)$, by Theorem 9.1.5.

For the converse, assume X is a $\mathcal{T}_0$ space and $\mathcal{Q}(X)$ is the free completely distributive lattice over the frame $\mathcal{O}(X)$. Then the set of all completely prime M-filter of $\Omega(X)$ coincides with the set of all completely prime M-filters of $\zeta_{\mathcal{O}(X)}$, that is,

$$CPMF(\Omega(X)) = CPMF(\zeta_{\mathcal{O}(X)}).$$

Notice that we have an equality and not an isomorphism because the frame parts of the two observation frames are identical. By combining Lemmas 8.1.8 and 8.1.15, and because X is a $\mathcal{T}_0$ space, the assignment $x \mapsto \{o \in \mathcal{O}(X) \mid x \in o\}$ is an isomorphism between X and $CPMF(\Omega(X))$. On the other hand, $CPMF(\zeta_{\mathcal{O}(X)})$ coincides with $CPF(\mathcal{O}(X))$ by Theorem 9.3.3. Hence X is sober. $\square$

The characterization of sober spaces by means of saturated sets can intuitively be explained as follows. Sober spaces are precisely those spaces whose completely distributive lattice of saturated sets can be presented without infinite meets in its relations. The following corollary is then immediate.

9.3.7. Corollary. *A $\mathcal{T}_0$ space X is sober if and only if the completely prime filters of $\mathcal{O}(X)$ coincide with the completely prime M-filters of $\Omega(X)$.*

Proof: If X is sober then X is isomorphic to the set $CPMF(\Omega(X))$ of all completely prime M-filters of $\Omega(X)$ by Lemma 8.1.15. By Theorem 9.3.6, $CPMF(\Omega(X))$ coincides with the set $CPMF(\zeta_{\mathcal{O}(X)})$ of all completely prime M-filters of $\zeta_{\mathcal{O}(X)}$. The latter set coincides with the set $CPF(\mathcal{O}(X))$ of completely prime filters of $\mathcal{O}(X)$ by Lemma 9.3.3 because $\mathcal{O}(X)$ is a spatial frame.

Conversely, assume $CPMF(\Omega(X))$ coincides with $CPF(\mathcal{O}(X))$. Because X is $\mathcal{T}_0$ the mapping $x \mapsto \{o \in \mathcal{O}(X) \mid x \in o\}$ is an isomorphism between X and $CPMF(\Omega(X))$. Hence, by definition, X is sober. $\square$

Recall from Chapter 5 that a sober space X is said to be *spectral* if finite intersections of compact open sets are still compact, and the compact open sets form a basis; a spectral space X is a *Stone space* if the complement of compact open is compact open. For a $\mathcal{T}_0$ space X, we have the following characterization 'by freeness':

(i) X is a Stone space if and only if its frame of open sets $\mathcal{O}(X)$ is free over the Boolean algebra of compact open sets $\mathcal{KO}(X)$;

(ii) X is a spectral space if and only if its frame of open sets $\mathcal{O}(X)$ is free over the distributive lattice of compact open sets $\mathcal{KO}(X)$;

(iii) X is a coherent space if and only if its frame of open sets $\mathcal{O}(X)$ is free over the distributive lattice of compact saturated sets $\mathcal{KQ}(X)$;

(iv) X is a sober space if and only if the completely distributive lattice of saturated sets $\mathcal{Q}(X)$ is free over the frame of open sets $\mathcal{O}(X)$.
The first three characterizations are standard and can be easily derived from similar results in [Joh82a, Vic89].

We conclude this chapter by establishing the fundamental role of M-filters in the setting of observation frames.

9.3.8. Lemma. *Let $\alpha : F \to L$ be an observation frame. There is an order isomorphism between L and the collection of M-filters $MF(\alpha)$ ordered by superset inclusion.*

Proof: We have already seen in Lemma 8.1.6 that the map $\mathcal{U} \mapsto \uparrow(\bigsqcap \alpha(\mathcal{U}))$ is an isomorphism between M-filters of α and principal filters of L. Since $\uparrow(\bigsqcap \alpha(\mathcal{U}))$ is isomorphic to $\bigsqcap \alpha(\mathcal{U})$ we have that the map

$$\mathcal{U} \mapsto \bigsqcap \alpha(\mathcal{U})$$

is an isomorphism between M-filters and elements of L with as inverse the mapping $q \mapsto \mathcal{U}(q) = \{a \in F \mid q \sqsubseteq \alpha(a)\}$. It is not difficult to see that both mappings are order preserving. □

Let $\alpha : F \to L$ be an observation frame. An element $q \in L$ is said to be *compact with respect to* F if for all directed subsets S of F, $q \sqsubseteq \bigsqcup \alpha(S)$ implies that there exists $s \in S$ such that $q \sqsubseteq \alpha(s)$. For example, for every topological space X, a set $q \in \mathcal{Q}(X)$ is compact with respect to $\mathcal{O}(X)$ if and only if q is compact.

The order-isomorphism of Lemma 9.3.8 restricts to elements of L compact with respect to F and Scott open M-filters of α, where an M-filter $\mathcal{U}$ of α is said to be *Scott open* if it is an open set in the Scott topology on F.

9.3.9. Lemma. *Let $\alpha : F \to L$ be an observation frame.*

(i) *An element q of L is compact with respect to F if and only if $\mathcal{U}(q)$ is a Scott open M-filter of α.*

(ii) *A subset $\mathcal{U}$ of F is a Scott open M-filter of α if and only if $\bigsqcap \alpha(\mathcal{U})$ is compact with respect to F.*

Proof: By Lemma 9.3.8 it is enough to prove only one implication for each item.

(i) Let $q \in L$ be compact with respect to F and let $S \subseteq F$ be a directed set. If $\bigvee S \in \mathcal{U}(q)$ then $q \sqsubseteq \alpha(\bigvee S) = \bigsqcup \alpha(S)$. Since q is compact with respect to F there exists $s \in S$ such that $q \sqsubseteq \alpha(s)$. Hence $s \in \mathcal{U}(q)$, that is, $\mathcal{U}(q)$ is a Scott open subset of F. It is also an M-filter of α because if $\bigsqcap \alpha(\mathcal{U}(q) \sqsubseteq \alpha(a)$ for some $a \in F$, then $q \sqsubseteq \alpha(a)$ and hence $a \in \mathcal{U}(q)$.

(ii) Assume $\mathcal{U}$ is a Scott open M-filter of α and let S be a directed subset of F. If $\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \bigsqcup \alpha(S) = \alpha(\bigvee S)$ then $\bigvee S \in \mathcal{U}$ because $\mathcal{U}$ is an M-filter. But it is also Scott open, hence there exists $s \in S$ such that $s \in \mathcal{U}$. Hence $\bigsqcap \alpha(\mathcal{U}) \sqsubseteq \alpha(s)$, that is, $\bigsqcap \alpha(\mathcal{U})$ is compact with respect to F. □

For an observation frame $\alpha : F \to L$ we can obtain a relationship between completely prime M-filters and elements $q \in L$ such that for every $S \subseteq F$ if $q \sqsubseteq \bigsqcup \alpha(S)$ then there exists an $s \in S$ such that $q \sqsubseteq \alpha(s)$. The proof is as before.

Lemma 9.3.9 is of a more fundamental nature than what is normally called the Hofmann-Mislove theorem (also known as the Scott-open filter theorem) given in Corollary 9.3.11 below. The latter is about Scott-open sets $F \subseteq \mathcal{O}(X)$ of a (sober) space X, which are *ordinary* filters. This theorem is due to Hofmann and Mislove [HM81], and can in our present setting be obtained from the following result. It is stated where the Axiom of Choice is used and when the soberness of the space is needed. We sketch the proof for reasons of completeness. It is very similar to Lemma 8.2.2 in [Vic89].

9.3.10. Lemma. *For a sober space X, a Scott-open set $\mathcal{F} \subseteq \mathcal{O}(X)$ is an M-filter if and only if it is an ordinary filter.*

Proof: The (only-if) part is obvious, so we concentrate on the (if) part. Take a Scott-open filter $\mathcal{F} \subseteq \mathcal{O}(X)$. We have to show

$$\bigcap \mathcal{F} \subseteq o' \quad \Rightarrow \quad o' \in \mathcal{F}.$$

Towards a contradiction, suppose $o' \notin \mathcal{F}$. Then we have to produce an element $x \in X$ with $x \in \bigcap \mathcal{F}$ but $x \notin o'$. Because X is sober it suffices to give a prime-open $p \in \mathcal{O}(X)$ with $p \notin \mathcal{F}$ and $o' \subseteq p$, where we think of p as the directed union $\bigcup\{o \in \mathcal{O}(X) \mid x \notin o\}$. Hence one considers the poset

$$P \quad = \quad \{u \in \mathcal{O}(X) \mid o' \subseteq u \text{ and } u \notin \mathcal{F}\}, \text{ ordered by inclusion.}$$

Every chain in P has an upper bound, so by Zorn's Lemma we get a maximal element $p \in P$. It remains to show that p is prime-open. Towards the contrary, assume

$$o_1 \cap o_2 \subseteq p \quad \text{but} \quad o_1 \not\subseteq p \qquad \text{and also} \qquad o_2 \not\subseteq p.$$

Because p is maximal, both the open sets $o_1' = p \cup o_1$ and $o_2' = p \cup o_2$ are in $\mathcal{F}$ and hence, because $\mathcal{F}$ is a filter, also $o_1' \cap o_2' \in \mathcal{F}$. But $o_1' \cap o_2' = p \cup (o_1 \cap o_2) = p$. Contradiction. □

Finally we obtain the result of Hofmann and Mislove [HM81] as a direct consequence of Lemma 9.3.10 and Lemma 9.3.9.

9.3.11. Corollary (Hofmann-Mislove theorem). *For a sober space X, there is an order isomorphism between the poset $(\mathcal{KQ}(X), \leq_U)$ of compact saturated sets and the poset of Scott-open filters $F \subseteq \mathcal{O}(X)$, ordered by inclusion.* □

In general, even for a sober space X, the set of all M-filters of the observation frame $\Omega(X)$ is strictly included in the set of all ordinary filters of $\mathcal{O}(X)$. We have already seen in the previous chapter an example of filter which is not an M-filter. Alternatively, consider an infinite set X with the discrete topology. There are many filters $\mathcal{F}$ for which $\bigcap \mathcal{F}$ is empty, ranging from the filter of cofinite sets to $\mathcal{P}(X)$ itself, and including all non-principal ultrafilters. But the empty set is a saturated subset of X, and hence, by Lemma 9.3.8, it corresponds to only one M-filter of $\Omega(X)$.

9.4 Concluding notes

The construction of the free completely distributive lattice over a frame which we presented in this chapter is rather indirect. First we construct the free completely distributive lattice $CDL(F)$ over the underlying set of a frame F and then we impose relations on $CDL(F)$ in order to turn it into a model of a presentation of F. It would be nice to characterize the free completely distributive lattice over a frame directly. The intuitively appealing filter completion of a frame in order to add the missing codirected meets unfortunately does not work: by Theorem 9.3.6, it would imply the existence of an isomorphism between filters of open sets and saturated sets of a sober space. Such an isomorphism does not always exists, as can be deduced from the last example after Corollary 9.3.11.

Presentations by generators and relations of observation frames enable us to define new observation frames for old. Much of the theory presented by Vickers [Vic89] for frames can be exported to observation frames. An interesting related question is whether the coproduct of two spatial observation frames gives a spatial observation frame. For ordinary frames coproduct does not need to preserve spatiality [Joh82a, Proposition 2.14].

Chapter 10

An infinitary domain logic for transition systems

The aim of this chapter is to apply the framework of observation frames in order to provide a bridge between the semantics of computations and their logic.

We treat a case study based on the theory developed by Abramsky [Abr87]. Our starting point is Abramsky's domain logic for transition systems [Abr91a]: his logic is equivalent to the Hennessy-Milner logic in the infinitary case, and hence it characterizes bisimulation for every transition system. However in the finitary case it is more satisfactory than the Hennessy-Milner logic in the sense that it characterizes the finitary observable part of bisimulation for every transition system.

The main result of Abramsky [Abr91a] is that the Lindenbaum algebra generated by his finitary logic is a distributive lattice dual to an SFP-domain obtained as a solution of a recursive domain equation. We extend Abramsky's result by proving that the Lindenbaum algebra generated by the infinitary logic is a completely distributive lattice dual to the same SFP-domain. As a consequence soundness and completeness of the infinitary logic is obtained for the class of finitary transition systems.

10.1 Domain theory in logical form

Complete partial orders were originally introduced as a mathematical structure to model computation [Sco70], in particular as domains for denotational semantics [SS71]. Successively, Scott's presentation of domains as information systems [Sco82] suggested a connection between denotational semantics and logics of programs. Abramsky [Abr87, Abr91b] uses Stone duality to relate two views of complete partial orders: one in terms of theories and one in terms of models.

Abramsky's starting point is that for an algebraic cpo P, its compact elements completely determine P, whereas for a logic the Lindenbaum algebra provides a model from which the logic can be recovered. If P is an SFP-domain, then the collection $\mathcal{KO}(P)$ of all Scott compact open subsets of P ordered by subset inclusion forms a distributive lattice, that is, P in its Scott topology is a spectral space. The distributive lattice $\mathcal{KO}(P)$ can be viewed as the Lindenbaum algebra of some logic. Conversely, given a distributive lattice L we can first construct the free frame $Idl(L)$ by ideal completing L, and then we derive a topological space from $Idl(L)$ using using the adjunction given in Proposition 9.3.5.

In this way, the duality between the category of spatial frames **SFrm** and the category of sober spaces **Sob** given in Proposition 9.3.5 cuts down to a duality between the category of distributive lattices **DLat** and the category **Spec** of spectral $\mathcal{T}_0$ spaces and continuous functions which preserve compact open subsets under inverse image ([Joh82a, II.2.11 and II.3.3]).

In his thesis [Abr87] Abramsky provides a method that allows to identify the logic that an SFP-domain generates. He considers a number of basic constructors of domain theory and shows how to generate a logic whose Lindenbaum algebra arises from the application of the constructors to the Lindenbaum algebras dual to the SFP-domains to which the constructors are to be applied. Abramsky's theory applies therefore to all SFP-domains freely generated by the constructors he considered. These constructors include lift, coalesced and separated sum, products, function space, Hoare, Smyth and Plotkin powerdomains, and recursion.

The logics of compact opens considered by Abramsky are weak in expressive power, and inadequate as a general specification formalism [Abr91b]. What we need is a language, with an accompanying semantic framework, which permits to go beyond compact open sets. A first step would be to allow more general open sets by means of infinite disjunctions. Since the spaces considered by Abramsky are spectral, this would not require a major adjustment of the semantic framework. However, for specification purposes we also need the ability to express infinite conjunctions. To this end we can apply the results of the previous two chapters—characterizing sober spaces in terms of the completely distributive lattice of saturated sets—and we can freely extend the finitary logic of compact opens to the infinitary logic of saturated sets. The extension is conservative in the sense that the topological space represented by a finitary logic coincides with the one represented by its infinitary extension.

10.2 Transition systems

We begin by recalling some notions on labelled transition systems (with divergence). In Chapter 4 we have already introduced transition systems as a basic mathematical structure for modeling computations of programming languages (see also [Plo81b]).

10.2.1. Definition. A *labelled transition system* with divergence is a tuple $\langle P, Act, \longrightarrow, \Uparrow \rangle$ where P is a set of processes, Act a set of atomic actions, $\longrightarrow \subseteq P \times Act \times P$ is a transition relation and $\Uparrow$ is a predicate on P. The predicate $\Uparrow$ is called the *divergence predicate*. The *convergence predicate* $\Downarrow$ on P is defined to be the complement of the divergence predicate, that is $\Downarrow = P \setminus \Uparrow$. We use $p \Uparrow$ ($p \Downarrow$) to denote that the process p diverges (converges).

Transition systems can be used to identify processes with the same observable behaviour. One of the well-known behavioral equivalences on processes is bisimulation [Mil80, Par81].

10.2.2. Definition. Given a transition system $\langle P, Act, \longrightarrow, \Uparrow \rangle$, a relation $R \subseteq P \times P$ is called a *partial bisimulation* whenever, if $\langle p, q \rangle \in R$ then for all $a \in Act$

(i) $p \xrightarrow{a} p' \;\Rightarrow\; \exists q' \in P$: $q \xrightarrow{a} q'$ & $\langle p', q' \rangle \in R$;
(ii) $p \Downarrow \Rightarrow\; q \Downarrow$ & $(q \xrightarrow{a} q' \;\Rightarrow\; \exists p' \in P$: $p \xrightarrow{a} p'$ & $\langle p', q' \rangle \in R)$.

We write $p \lesssim^B q$ if there exists a partial bisimulation R with $\langle p, q \rangle \in R$.

Often we will use partial bisimulations to compare processes from different transition systems. This can be formally done by taking the disjoint union of the two systems, and using the above definition of partial bisimulation. Partial bisimulations can also be described in terms of iteration [Par81], but in general one needs to consider a non-countable sequence of relations (in the complete lattice $\mathcal{P}(P \times P)$ ordered by subset inclusion) approximating $\lesssim^B$. By considering only countable approximants of $\lesssim^B$ one obtains the so-called *observable equivalence* $\lesssim^\omega = \bigcap_\omega \lesssim^n$ [Mil80], where

- $\lesssim^0 = P \times P$, and
- $p \lesssim^{n+1} q$ if and only if for all $a \in Act$

  (i) $p \xrightarrow{a} p' \;\Rightarrow\; \exists q' \in P$: $q \xrightarrow{a} q'$ & $p' \lesssim^n q'$;
  (ii) $p \Downarrow \Rightarrow\; q \Downarrow$ & $(q \xrightarrow{a} q' \;\Rightarrow\; \exists p' \in P$: $p \xrightarrow{a} p'$ & $p' \lesssim^n q')$.

In general for a transition system T, $\lesssim^B \subseteq \lesssim^\omega$. However, if T is image-finite then the two notions coincide [HM85], where $T = \langle P, Act, \longrightarrow, \Uparrow \rangle$ is said to be *image-finite* if for all processes $p \in P$ and actions $a \in Act$ the set

$$\{q \mid p \xrightarrow{a} q\}$$

is finite.

A particular example of a transition system is given by the collection of all (finite) *synchronization trees* over an alphabet Act of actions. Define the set $(t \in) ST(Act)$ of finitary synchronization trees over Act by

$$t \quad ::= \quad \Sigma_I\, a_i t_i \mid \Sigma_I\, a_i t_i + \Omega,$$

where I is a finite index set, and all the a_i's are actions in Act for $i \in I$. If $I = \emptyset$ then we write Θ for $\Sigma_I a_i t_i$, and Ω for $\Sigma_I a_i t_i + \Omega$. The set of all finitary synchronization trees can be turned into a transition system $ST(Act) = \langle ST(Act), Act, \longrightarrow, \Uparrow\rangle$, where

- $t \Uparrow$ if and only if Ω is included as a summand of t, and
- $t \xrightarrow{a_i} t_i$ for each summand $a_i t_i$ of t.

Synchronization trees can be used to define a finitary preorder on processes of more general transition systems [Gue81].

10.2.3. Definition. For a transition system $\langle P, Act, \longrightarrow, \Uparrow\rangle$ define the *finitary preorder* $\lesssim^F \subseteq P \times P$ by

$$p \lesssim^F q \text{ if and only if } \forall t \in ST(Act)\colon t \lesssim^B p \;\Rightarrow\; t \lesssim^B q.$$

Since finite synchronization trees are a model for finite processes, the finitary preorder can be considered as the finite observable part of partial bisimulation. For every transition system T, it holds that

$$\lesssim^B \subseteq \lesssim^\omega \subseteq \lesssim^F .$$

In general, these inclusions are strict [Abr91a, pag. 191]. If $p \in P$ is a process of a transition system $\langle P, Act, \longrightarrow, \Uparrow\rangle$ and $t \in ST(Act)$ is a finite synchronization tree it holds that $t \lesssim^B p$ if and only if $t \lesssim^\omega p$ [Abr91a, Lemma 5.10].

Another example of a transition system is given by the SFP-domain $\mathcal{D}$ obtained as the initial (and final) solution in the category **SFP** of the recursive domain equation

$$X \cong (\mathbf{1})_\perp \oplus (\mathcal{P}_c^{co}\left(\sum_{a\in Act} X\right) \setminus \{\emptyset\}),$$

where $\mathbf{1}$ is the one-point cpo, Act is a countable set of actions, $(-)_\perp$ is the lift, $\oplus$ is the coalesced sum, $\sum_{a\in Act}$ is the countable separated sum, and $\mathcal{P}_c^{co}(-) \setminus \{\emptyset\}$ is the Plotkin powerdomain (we use Proposition 6.3.7 to characterize it in terms of Scott-compact and convex sets). Below we will omit the isomorphism pair relating the left and the right hand side of the solution $\mathcal{D}$ of the above domain equation. The SFP-domain $\mathcal{D}$ can be seen as the transition system $\langle \mathcal{D}, Act, \longrightarrow, \Uparrow\rangle$ where

- $d \xrightarrow{a} d'$ if and only if $\langle a, d'\rangle \in d$ and
- $d \Uparrow$ if and only if $\perp \in d$.

The above definition implies, for $d \in \mathcal{D}$, that if $d = \mathbf{1}$ then $d \Downarrow$ and for all $a \in Act$ and $d' \in \mathcal{D}$ there is no transition $d \xrightarrow{a} d'$; whereas if $d = \{\perp\}$ then $d \Uparrow$ and for all $a \in Act$ and $d' \in \mathcal{D}$ there is no transition $d \xrightarrow{a} d'$.

Abramsky's logic for transition systems

Like the Hennessy-Milner logic [HM85], the idea of Abrasmky's infinitary logic $\mathcal{L}_{\infty,\infty}$ for transition systems [Abr91a] is to obtain a suitable characterization of partial bisimulation in terms of a notion of property of processes: $p \lesssim^B q$ if and only if every property satisfied by p is also satisfied by q. However, the finitary restriction of Abramsky's logic differs from the finitary Hennessy-Milner logic in the sense that it characterizes the finitary observable part of partial bisimulation for all transition systems.

10.2.4. Definition. Let $(a \in)Act$ be a set of actions. The language $\mathcal{L}_{\infty,\infty}$ over Act has two sorts: π (processes) and k (capabilities). We write $(\phi \in)\mathcal{L}^{\pi}_{\infty,\infty}$ for the class of formulae of sort π, and $(\psi \in)\mathcal{L}^{k}_{\infty,\infty}$ for the class of formulae of sort k, which are defined inductively as follows:

$$\begin{aligned} \phi &::= \bigvee_I \phi_i \mid \bigwedge_I \phi_i \mid \Box\psi \mid \Diamond\psi \\ \psi &::= \bigvee_I \psi_i \mid \bigwedge_I \psi_i \mid a(\phi), \end{aligned}$$

where I is an arbitrary index set. If $I = \emptyset$ then we write tt for $\bigwedge_I \phi_i$ and $\bigwedge_I \psi_i$, and we write $f\!f$ for $\bigvee_I \phi_i$ and $\bigvee_I \psi_i$.

Before we interpret the language $\mathcal{L}_{\infty,\infty}$ we need the following definitions. For a transition system $\langle P, Act, \longrightarrow, \Uparrow\rangle$ define the set Cap of *capabilities* by

$$Cap = \{\bot\} \cup (Act \times P).$$

The set of capabilities of a process $p \in P$ is given by

$$C(p) \;=\; \{\bot \mid p \Uparrow\} \cup \{\langle a, q\rangle \mid p \xrightarrow{a} q\}.$$

For a transition system $T = \langle P, Act, \longrightarrow, \Uparrow\rangle$, we interpret the language $\mathcal{L}_{\infty,\infty}$ by means of the *satisfaction relations* $\models_\pi \subseteq P \times \mathcal{L}^{\pi}_{\infty,\infty}$ and $\models_k \subseteq Cap \times \mathcal{L}^{k}_{\infty,\infty}$ defined as follows:

$$\begin{array}{lcl} p \models_\pi \bigwedge_I \phi_i & \Leftrightarrow & \forall i \in I \colon p \models_\pi \phi_i \\ p \models_\pi \bigvee_I \phi_i & \Leftrightarrow & \exists i \in I \colon p \models_\pi \phi_i \\ p \models_\pi \Box\phi & \Leftrightarrow & p \Downarrow \text{ and } \forall c \in C(p) \colon c \models_k \phi \\ p \models_\pi \Diamond\phi & \Leftrightarrow & \exists c \in C(p) \colon c \models_k \phi \\ & & \\ c \models_k \bigvee_I \phi_i & \Leftrightarrow & \exists i \in I \colon c \models_k \phi_i \\ c \models_k \bigwedge_I \phi_i & \Leftrightarrow & \forall i \in I \colon c \models_k \phi_i \\ c \models_k a(\phi) & \Leftrightarrow & c = \langle a, q\rangle \text{ and } q \models_\pi \phi. \end{array}$$

For a transition system $T = \langle P, Act, \longrightarrow, \Uparrow\rangle$ and formula ϕ of $\mathcal{L}^{\pi}_{\infty,\infty}$ we write $[\![\phi]\!]^{\pi}_{T}$ for $\{p \in P \mid p \models_\pi \phi\}$. *Assertions* A over the language $\mathcal{L}^{\sigma}_{\infty,\infty}$ are of the form $\phi \leq_\sigma \psi$ or $\phi =_\sigma \psi$ for σ in $\{\pi, k\}$ with ϕ and ψ in $\mathcal{L}^\sigma_{\infty,\infty}$. The satisfaction relation between transition systems T and assertions is defined by

$$\begin{array}{lcl} T \models \phi \leq_\pi \psi & \Leftrightarrow & \forall p \in P\colon p \models_\pi \phi \text{ implies } p \models_\pi \psi \\ T \models \phi =_\pi \psi & \Leftrightarrow & \forall p \in P\colon p \models_\pi \phi \text{ if and only if } p \models_\pi \psi \\ & & \\ T \models \phi \leq_k \psi & \Leftrightarrow & \forall c \in Cap\colon c \models_k \phi \text{ implies } c \models_k \psi \\ T \models \phi =_k \psi & \Leftrightarrow & \forall c \in Cap\colon c \models_k \phi \text{ if and only if } c \models_k \psi. \end{array}$$

As usual, the satisfaction relation can be extended to classes of transition systems $\mathcal{T}$ by

$$\mathcal{T} \models A \quad \Leftrightarrow \quad \forall\ T \in \mathcal{T}\colon T \models A.$$

If $\mathcal{T}$ is the class of all transition systems then we simply write $\models A$.

Let $\mathcal{L}_{\omega,\omega}$ be the sub-language of $\mathcal{L}_{\infty,\infty}$ obtained by the restriction to finite conjunctions and finite disjunctions.

10.2.5. Theorem. *For a transition system $\langle P, Act, \longrightarrow, \Uparrow\rangle$ and p, q in P,*

(i) *$p \lesssim^B q$ if and only if $\forall \phi \in \mathcal{L}^\pi_{\infty,\infty}\colon p \models \phi \;\Rightarrow\; q \models \phi$;*
(ii) *$p \lesssim^F q$ if and only if $\forall \phi \in \mathcal{L}^\pi_{\omega,\omega}\colon p \models \phi \;\Rightarrow\; q \models \phi$.*

Proof: See Theorems 5.6 and 5.8 in [Abr91a]. □

Next we present a proof system for assertions over $\mathcal{L}_{\infty,\infty}$. (We omit the sort subscripts.) The following *logical axioms* give to the language the structure of a large completely distributive lattice.

$$(\leq -\mathit{ref})\ \phi \leq \phi \qquad\qquad (\leq -\mathit{trans})\ \frac{\phi \leq \psi \ \&\ \psi \leq \chi}{\phi \leq \chi}$$

$$(= -I)\ \frac{\phi \leq \psi \ \&\ \psi \leq \phi}{\phi = \psi} \qquad\qquad (= -E)\ \frac{\phi = \psi}{\phi \leq \psi \ \&\ \psi \leq \phi}$$

$$(\wedge - I)\ \frac{\{\phi \leq \psi_i\}_{i \in I}}{\phi \leq \bigwedge_I \psi_i} \qquad\qquad (\wedge - E)\ \textstyle\bigwedge_I \phi_i \leq \phi_k \quad (k \in I)$$

$$(\vee - I)\ \frac{\{\phi_i \leq \psi\}_{i \in I}}{\bigvee_I \phi_i \leq \psi} \qquad\qquad (\vee - E)\ \phi_k \leq \textstyle\bigvee_I \phi_i \quad (k \in I)$$

$$(\wedge - \mathit{dist})\ \textstyle\bigwedge_I \bigvee_{J_i} \phi_{i,j} \quad = \quad \bigvee_{f \in \Phi(I)} \bigwedge_I \phi_{i,f(i)}$$

where $\Phi(I)$ denotes the set of all functions $f : I \to \bigcup_I J_i$ with $f(i) \in J_i$ for all $i \in I$. The dual of the $(\wedge - \mathit{dist})$-axiom is derivable from these logical axioms [Ran52]. The following *modal axioms* relate constructors with the logical structure.

$$(a-\leq)\ \frac{\phi \leq \psi}{a(\phi) \leq a(\psi)}$$

$$(a-\wedge)\ \begin{array}{lll}\text{(i)} & a(\bigwedge_I \phi_i) = \bigwedge_I a(\phi_i) & (I \neq \emptyset) \\ \text{(ii)} & a(\phi) \wedge b(\psi) = F & (a \neq b)\end{array} \qquad (a-\vee)\ a(\bigvee_I \phi_i) = \bigvee_I a(\phi_i)$$

$$(\Box-\leq)\ \frac{\phi \leq \psi}{\Box\phi \leq \Box\psi}$$

$$(\Box-\wedge)\ \Box \bigwedge_I \phi_i = \bigwedge_I \Box\phi_i \quad (I \neq \emptyset) \qquad (\Box-\vee)\ \Box(\phi \vee \psi) \leq \Box\phi \vee \Diamond\psi$$

$$(\Diamond-\leq)\ \frac{\phi \leq \psi}{\Diamond\phi \leq \Diamond\psi}$$

$$(\Diamond-\wedge)\ \Box\phi \wedge \Diamond\psi \leq \Diamond(\phi \wedge \psi) \qquad (\Diamond-\vee)\ \Diamond \bigvee_I \phi_i = \bigvee_I \Diamond\phi_i$$

We write $\mathcal{L}_{\infty,\infty} \vdash A$ if the assertion A of $\mathcal{L}_{\infty,\infty}$ is derivable from the above axioms and rules.

10.2.6. Theorem. *(Soundness) If $\mathcal{L}_{\infty,\infty} \vdash A$ then $\models A$.*

Proof: See Theorem 4.2 in [Abr91a]. □

Next we turn to the finitary logic $\mathcal{L}_{\omega,\omega}$ in order to prove the reverse of the above result for the class of all transition systems.

Define the *syntactic equivalence* $\sim$ on formulae of $\mathcal{L}^{\pi}_{\omega,\omega}$ by

$$\phi_1 \sim \phi_2 \quad \text{if and only if} \quad \mathcal{L}^{\pi}_{\omega,\omega} \vdash \phi_1 =_{\pi} \phi_2.$$

Clearly, $\sim$ is an equivalence relation. Let $[\phi]$ be the equivalence class of ϕ under $\sim$, and denote the set of $\sim$ equivalence classes by $\mathcal{LA}^{\pi}_{\omega,\omega}$. There are natural operations making $\mathcal{LA}^{\pi}_{\omega,\omega}$ a distributive lattice. To show this, we define an order $\leq$ on $\mathcal{LA}^{\pi}_{\omega,\omega}$ by

$$[\phi_1] \leq [\phi_2] \quad \text{if and only if } \mathcal{L}^{\pi}_{\omega,\omega} \vdash \phi_1 \leq_{\pi} \phi_2.$$

It can be checked that $\leq$ is a well-defined order relation. Notice that the poset $\mathcal{LA}^{\pi}_{\omega,\omega}$ has greatest and least elements given by the equivalence classes of all theorems in $\mathcal{LA}^{\pi}_{\omega,\omega}$ and of all non-derivable formulae in $\mathcal{LA}^{\pi}_{\omega,\omega}$, respectively. The next step is to define join and meet in $\mathcal{LA}^{\pi}_{\omega,\omega}$:

$$[\phi_1] \vee [\phi_2] =_{def} [\phi_1 \vee \phi_2] \quad \text{and} \quad [\phi_1] \wedge [\phi_2] =_{def} [\phi_1 \wedge \phi_2].$$

The above operations are well-defined. Moreover, by the logical axioms, it follows that the join and the meet of the poset $\mathcal{LA}^{\pi}_{\omega,\omega}$ are given by the above

$\vee$ and $\wedge$, and that the poset $\mathcal{LA}^{\pi}_{\omega,\omega}$ is in fact a distributive lattice. We call the lattice $\mathcal{LA}^{\pi}_{\omega,\omega}$ the *Lindenbaum algebra* of $\mathcal{L}^{\pi}_{\omega,\omega}$.

Since $\mathcal{LA}^{\pi}_{\omega,\omega}$ is a distributive lattice, it can be represented by a spectral space. The following fundamental result shows that the finitary logic $\mathcal{L}^{\pi}_{\omega,\omega}$ does indeed correspond exactly to the SFP-domain $\mathcal{D}$ defined in the previous section and taken with the Scott topology.

10.2.7. Theorem. *Let $\mathcal{KO}(\mathcal{D})$ be the distributive lattice of Scott compact open sets of $\mathcal{D}$ ordered by subset inclusion. The function $\gamma : \mathcal{LA}^{\pi}_{\omega,\omega} \to \mathcal{KO}(\mathcal{D})$ defined, for ϕ in $\mathcal{L}^{\pi}_{\omega,\omega}$, by*

$$\gamma([\phi]) = [\![\phi]\!]^{\pi}_{\mathcal{D}}$$

is a well-defined order isomorphism.

Proof: See Theorem 4.3 in [Abr91a]. □

As already suggested by Lemma 8.2.8, the proof of the spatiality of the distributive lattice $\mathcal{LA}^{\pi}_{\omega,\omega}$ is equivalent to completeness of the underlying logical system. Indeed, (strong) completeness of $\mathcal{L}^{\pi}_{\omega,\omega}$ is an immediate consequence of the soundness Theorem 8.2.7 and of the above duality result.

10.2.8. Theorem. *(Completeness) Let $\mathcal{T}$ be any class of transition systems containing $\mathcal{D}$. For ϕ_1 and ϕ_2 in $\mathcal{L}^{\pi}_{\omega,\omega}$, $\mathcal{T} \models \phi_1 \leq \phi_2$ if and only if $\mathcal{L}^{\pi}_{\omega,\omega} \vdash \phi_1 \leq \phi_2$.*

Proof: For ϕ_1 and ϕ_2 in $\mathcal{L}^{\pi}_{\omega,\infty}$ we have,

$$\begin{array}{rcll} \mathcal{D} \models_{\pi} \phi_1 \leq_{\pi} \phi_2 & \Leftrightarrow & [\![\phi_1]\!]^{\pi}_{\mathcal{D}} \subseteq [\![\phi_2]\!]^{\pi}_{\mathcal{D}} & \\ & \Leftrightarrow & \gamma([\phi_1]) \subseteq \gamma([\phi_2]) & [\text{definition of } \gamma] \\ & \Leftrightarrow & [\phi_1] \leq [\phi_2] & [\gamma \text{ is an order isomorphism}] \\ & \Leftrightarrow & \mathcal{L}^{\pi}_{\omega,\omega} \vdash \phi_1 \leq_{\pi} \phi_2. & [\text{definition of } \mathcal{LA}^{\pi}_{\omega,\omega}] \end{array}$$

□

We conclude this section by showing that the SFP-domain $\mathcal{D}$ can be used as semantic domain for all transition systems. Let $T = \langle P, Act, \longrightarrow, \Uparrow\rangle$ be a transition system and let $p \in P$. The set

$$TS(p) = \{[\phi] \in \mathcal{LA}^{\pi}_{\omega,\omega} \mid p \models_{\pi} \phi\}$$

is a prime filter of the distributive lattice $\mathcal{LA}^{\pi}_{\omega,\omega}$. Hence, by Theorem 10.2.7, it corresponds uniquely to an element in $\mathcal{D}$. Therefore, the assignment $p \mapsto TS(p)$ defines a function $TS[\![\cdot]\!] : P \to D$ which is unique among all functions $f : P \to D$ such that

$$p \models_{\pi} \phi \quad \text{if and only if} \quad f(p) \models_{\pi} \phi,$$

for all $p \in P$ and $\phi \in \mathcal{L}^{\pi}_{\omega,\omega}$ [Abr91a, Theorem 5.21]. By the characterization Theorem 10.2.5, it follows that p and $TS[\![p]\!]$ are equivalent in the finitary preorder $\lesssim^F$. Hence the function $TS[\![\cdot]\!] : P \to \mathcal{D}$ can be regarded as a syntax-free semantics which is universal because it is defined for every transition system.

10.3 Compactly branching transition systems

Consider properties of a transition system $\langle P, Act, \longrightarrow, \Uparrow\rangle$ like 'the process p converges', 'every a-path starting from p is finite', or 'along every a-path starting from p eventually ψ holds'. The finitary language $\mathcal{L}^{\pi}_{\omega,\omega}$ is too weak to formalize these properties, which however can be expressed in the infinitary language $\mathcal{L}^{\pi}_{\infty,\infty}$ by

- $p \models_\pi \Box \bigvee_{a\in Act} a(tt)$;
- $p \models_\pi \bigvee_{n\in\omega} \phi_n$, where $\begin{cases} \phi_0 & = \mathit{ff} \quad \text{and} \\ \phi_{n+1} & = \Box(a(\phi_n) \vee \bigvee_{Act\setminus\{a\}} b(tt)); \end{cases}$
- $p \models_\pi \bigvee_{n\in\omega} \phi_n$, where $\begin{cases} \phi_0 & = \mathit{ff} \ \text{and} \\ \phi_{n+1} & = \psi \wedge \Diamond a(tt) \vee \Box(a(\phi_n) \vee \bigvee_{Act\setminus\{a\}} b(tt)). \end{cases}$

What we need is a language which allows more general formulae. An example is $\mathcal{L}_{\omega,\infty}$, the sub-language of $\mathcal{L}_{\infty,\infty}$ which allows infinite disjunctions but has only finite conjunctions. Adding expressive power to the finitary logic should not change our main motivation for its introduction: it should characterize the finitary observable part of partial bisimulation. We introduce the following scheme over $\mathcal{L}_{\omega,\infty}$ which restricts the class of transition systems and allows to write any formula in $\mathcal{L}_{\omega,\infty}$ as disjunctions (possibly infinite) of finitary formulae in $\mathcal{L}_{\omega,\omega}$:

$$(BN) \quad \Box \bigvee_I \phi_i \leq \bigvee_{J\in Fin(I)} \Box \bigvee_J \phi_j \quad (\phi_i \in \mathcal{L}_{\omega,\omega})$$

where $Fin(I)$ is the set of all finite subsets of I. The axioms (BN) could equivalently be formulated as: $\Box$ is Scott-continuous (preserves directed joins). This explains that the intuition behind (BN) is that of *bounded non-determinism.* For example, every weakly finitely branching transition system satisfies (BN) (a transition system $\langle P, Act, \longrightarrow, \Uparrow\rangle$ is said to be *weakly finitely branching* if for all $p \in P$,

$$p \Downarrow \ \text{ implies } \ \{q \in P \mid \exists a \in Act\colon p \xrightarrow{a} q\}$$

is finite. We denote the set on the right-hand side by $Br(p)$).

Transition systems satisfying (BN) can be characterized topologically as follows. Let $T = \langle P, Act, \longrightarrow, \Uparrow\rangle$ be a transition system and let $\mathcal{O}(T)$ denote the set of all $[\![\phi]\!]^{\pi}_{T}$ for ϕ in $\mathcal{L}^{\pi}_{\omega,\infty}$. Clearly, $\mathcal{O}(T)$ forms a topology on P. Transition systems together with a topology are introduced in the context of modal logic in [Esa74], where, in a restricted form, the direction from left to right of the next lemma is proved (the proof of the other direction can been found in [BK95]).

10.3.1. Lemma. *A transition system $T = \langle P, Act, \longrightarrow, \Uparrow\rangle$ satisfies (BN) if and only if for all $p \in P$ such that $p \Downarrow$ the set $Br(p)$ is compact in the topology $\mathcal{O}(T)$.*

Proof: Assume T satisfies (BN), take a $p \in P$ with $p \Downarrow$ and $Br(p) \subseteq \bigcup_I [\![\phi_i]\!]^\pi_T$. Then $p \models_\pi \Box \bigvee_I \phi_i$. Hence, by (BN), $p \models_\pi \bigvee_{J \in Fin(I)} \Box \bigvee_J \phi_j$, that is, $Br(p) \subseteq \bigcup_J [\![\phi_j]\!]^\pi_T$ for a finite subset J of I. Hence $Br(p)$ is compact in $\mathcal{O}(T)$.

Conversely, assume that if $p \Downarrow$ then the set $Br(p)$ is compact in the topology $\mathcal{O}(T)$, and let $p \models_\pi \Box \bigvee_I \phi_i$. Since p converges, $Br(p) \subseteq \bigcup_I [\![\phi_i]\!]^\pi_T$. But $Br(p)$ is compact, hence $Br(p) \subseteq \bigcup_J [\![\phi_j]\!]^\pi_T$ for some finite subset J of I. It follows that p satisfies (BN). $\Box$

A transition system is called *compactly branching* if it satisfies all instances of (BN). Using the duality Theorem 10.2.7 and the definition of the Plotkin powerdomain, it is immediate to see that the transition system induced by the SFP-domain $\mathcal{D}$ is compactly branching (and not weakly finite branching).

10.3.2. Lemma. *For every formula ϕ in $\mathcal{L}^\pi_{\omega,\infty}$ there exist formulae $\phi_i \in \mathcal{L}_{\omega,\omega}$ with $i \in I$ such that $\mathcal{L}^\pi_{\omega,\infty} + (BN) \vdash \phi = \bigvee_I \phi_i$.*

Proof: Adapt Lemma 5.17 of [Abr91a]. $\Box$

The above lemma together with the soundness Theorem 8.2.7, the definition of the satisfaction relation, and the characterization Theorem 10.2.5, imply that for compactly branching transition systems $\langle P, Act, \longrightarrow, \Uparrow\rangle$ and processes p, q in P,

$$p \precsim^F q \quad \text{if and only if} \quad \forall \phi \in \mathcal{L}^\pi_{\omega,\infty}\colon p \models \phi \;\Rightarrow\; q \models \phi.$$

The next step is to prove the completeness of the logic $\mathcal{L}^\pi_{\omega,\infty}$ for the class of compactly branching transition systems. We proceed as for the finite case: define the syntactic equivalence $\sim$ on formulae of $\mathcal{L}^\pi_{\omega,\infty}$ by

$$\phi_1 \sim \phi_2 \quad \text{if and only if} \quad \mathcal{L}^\pi_{\omega,\infty} + (BN) \vdash \phi_1 =_\pi \phi_2.$$

Let $\mathcal{LA}^\pi_{\omega,\infty}$ be the Lindenbaum algebra of $\mathcal{L}^\pi_{\omega,\infty}$ with as objects equivalence classes of formulae in $\mathcal{L}^\pi_{\omega,\infty}$ under $\sim$, ordered by

$$[\phi_1] \leq [\phi_2] \quad \text{if and only if } \mathcal{L}^\pi_{\omega,\infty} + (BN) \vdash \phi_1 \leq_\pi \phi_2.$$

The poset $\mathcal{LA}^\pi_{\omega,\infty}$ is a frame with meets and joins defined as expected.

10.3.3. Lemma. *The frame $\mathcal{LA}^\pi_{\omega,\infty}$ is free over the distributive lattice $\mathcal{LA}^\pi_{\omega,\omega}$.*

Proof: For every frame F and function $f : \mathcal{LA}^\pi_{\omega,\omega} \to F$ preserving finite meets and finite joins, define $h : \mathcal{LA}^\pi_{\omega,\infty} \to F$ by $h([\phi]) = \bigvee_I f([\phi_i])$ where, using

Lemma 10.3.2, $[\phi] = \bigvee_I [\phi_i]$ with ϕ_i in $\mathcal{L}^{\pi}_{\omega,\omega}$. By definition, h preserves joins and $h([\phi]) = f([\phi])$ for all ϕ in $\mathcal{L}^{\pi}_{\omega,\omega}$. Moreover, h preserves finite meets:

$$\begin{aligned}
h([\phi] \wedge [\phi']) &= h([\bigvee_I \phi_i] \wedge [\bigvee_J \phi'_j]) \quad \text{[Lemma 10.3.2]}\\
&= h(\bigvee_{I\times J} [\phi_i \wedge \phi'_j]) \quad \text{[distributivity]}\\
&= \bigvee_{I\times J} f([\phi_i \wedge \phi'_j]) \quad [h \text{ preserves joins and commutativity}]\\
&= \bigvee_{I\times J} (f([\phi_i]) \wedge f([\phi'_j])) \quad [f \text{ preserves meets}]\\
&= \bigvee_{I\times J} (h([\phi_i]) \wedge h([\phi'_j])) \quad \text{[commutativity]}\\
&= \bigvee_I h(\phi_i]) \wedge \bigvee_J h([\phi'_j]) \quad \text{[distributivity]}\\
&= h([\phi]) \wedge h([\phi']).
\end{aligned}$$

Hence h is the unique frame morphism such that $h \circ \iota = f$, where $\iota : \mathcal{LA}^{\pi}_{\omega,\omega} \to \mathcal{LA}^{\pi}_{\omega,\infty}$ is the obvious inclusion function. □

We can now draw an interesting consequence of the finitary axiom (BN).

10.3.4. Lemma. *The assignment $[\phi] \mapsto [\![\phi]\!]^{\pi}_{\mathcal{D}}$ defines a unique order isomorphism $\gamma^+ : \mathcal{LA}^{\pi}_{\omega,\infty} \to \mathcal{O}(\mathcal{D})$ such that $\gamma^+([\phi]) = \gamma([\phi])$ for all ϕ in $\mathcal{L}^{\pi}_{\omega,\omega}$.*

Proof: Because $\mathcal{D}$ is an SFP-domain, when taken with its Scott topology it forms a spectral space. Hence the lattice of Scott open sets $\mathcal{O}(\mathcal{D})$ is the free frame over the distributive lattice of Scott compact open sets $\mathcal{KO}(\mathcal{D})$, which, by Lemma 10.2.7, is order isomorphic to the Lindenbaum algebra $\mathcal{LA}^{\pi}_{\omega,\omega}$. But $\mathcal{LA}^{\pi}_{\omega,\infty}$ is the free frame over the distributive lattice $\mathcal{LA}^{\pi}_{\omega,\omega}$ (Lemma 10.3.3), hence $\mathcal{O}(\mathcal{D})$ is order isomorphic to $\mathcal{LA}^{\pi}_{\omega,\infty}$. The isomorphism is given by the unique extension γ^+ of the function $\gamma : \mathcal{LA}^{\pi}_{\omega,\omega} \to \mathcal{KO}(\mathcal{D})$ given in Lemma 10.3.3. Using Lemma 10.3.2 and the soundness Theorem 10.2.6, it can be characterized by

$$\gamma^+([\phi]) = \bigcup_I \gamma([\phi_i]) = \bigcup_I [\![\phi_i]\!]^{\pi}_{\mathcal{D}} = [\![\bigvee_I \psi_i]\!]^{\pi}_{\mathcal{D}} = [\![\phi]\!]^{\pi}_{\mathcal{D}},$$

for all ϕ in $\mathcal{L}^{\pi}_{\omega,\infty}$. □

Soundness of the logical system associated to $\mathcal{L}^{\pi}_{\omega,\infty}$ including the scheme (BN) follows from Theorem 10.2.6 and the definition of compactly branching transition systems. In a way similar to the completeness Theorem 10.2.8, completeness follows from the duality Lemma 10.3.4.

10.3.5. Theorem. *(Completeness) Let $\mathcal{CB}$ be any class of compactly branching transition systems containing $\mathcal{D}$. For ϕ_1 and ϕ_2 in $\mathcal{L}^{\pi}_{\omega,\infty}$, $\mathcal{CB} \models \phi_1 \leq \phi_2$ if and only if $\mathcal{L}^{\pi}_{\omega,\infty} + (BN) \vdash \phi_1 \leq \phi_2$.* □

10.4 Finitary transition systems

The language $\mathcal{L}_{\omega,\infty}$ is more expressive than the finitary language $\mathcal{L}_{\omega,\omega}$. Next we consider the even more expressive language $\mathcal{L}_{\infty,\infty}$. For example, given a transition system $\langle P, Act, \longrightarrow, \Uparrow\rangle$ we can specify in $\mathcal{L}^{\pi}_{\infty,\infty}$ properties like 'there exists an infinite a-path starting from the process p', and 'at any point of any path starting from p an a-transition is always possible', respectively by

- $p \models_\pi \bigwedge_{n\in\omega} \phi_n$, where $\begin{cases} \phi_0 & = tt \text{ and} \\ \phi_{n+1} & = \Diamond a(\phi_n); \end{cases}$
- $p \models_\pi \bigwedge_{n\in\omega} \phi_n$, where $\begin{cases} \phi_0 & = tt \text{ and} \\ \phi_{n+1} & = \Diamond a(\phi_n) \wedge \bigwedge_{Act}(\Box b(\phi_n) \vee \bigvee_{Act\setminus\{b\}} c(tt)). \end{cases}$

This kind of properties cannot be represented by open sets, but they can be expressed by sets which are saturated with respect to the topology $\mathcal{O}(T)$ associated to the transition system T. By using results on observation frames, we can prove a completeness result following the same pattern as for the completeness result of $\mathcal{L}_{\omega,\infty}$.

We begin with the introduction of two finitary axiom schemes over $\mathcal{L}_{\infty,\infty}$. These schemes will restrict the class of transition systems under consideration, and will allow us to write a formula in $\mathcal{L}_{\infty,\infty}$ as a conjunction of disjunctions of finitary formulae in $\mathcal{L}_{\omega,\omega}$:

$$\begin{array}{lll} (BN) & \Box \bigvee_I \phi_i \leq \bigvee_{J\in Fin(I)} \Box \bigvee_J \phi_j & (\phi_i \in \mathcal{L}_{\omega,\omega}) \\ (FA) & \bigwedge_{J\in Fin(I)} \Diamond \bigwedge_J \phi_j \leq \Diamond \bigwedge_I \phi_i & (\phi_i \in \mathcal{L}_{\omega,\omega}), \end{array}$$

where $Fin(I)$ is the set of all finite subsets of I. The axiom scheme (FA) is the dual of (BN). While the axiom (BN) is related to the width of a computation, the axiom (FA) is related to the length of it. It can be understood as a notion of *finite approximation*. For example, every weakly finite branching transition system with no infinite transition sequences satisfies both (BN) and (FA). An example of this kind of transition system is given by the set of finite synchronization trees. In general, a transition system which satisfies all instances of (BN) and (FA) is called *finitary*. The main example of finitary transition systems (which has infinite transition sequences) is given by the transition system induced by the SFP-domain $\mathcal{D}$ (Theorem 5.15 in [Abr91a]).

10.4.1. Lemma. *For each ϕ in $\mathcal{L}^{\pi}_{\infty,\infty}$ there exist formulae $\phi_i \in \mathcal{L}^{\pi}_{\omega,\infty}$, $i \in I$, such that $\mathcal{L}^{\pi}_{\infty,\infty} + (BN) + (FA) \vdash \phi =_\pi \bigwedge_I \phi_i$.*

Proof: See Lemma 5.17 of [Abr91a] and Lemma 10.3.2. □

An immediate consequence of the above lemma is the following characterization property. For a finitary transition system $\langle P, Act, \longrightarrow, \Uparrow\rangle$ and p, q in P,

$$p \lesssim^F q \quad \text{if and only if} \quad \forall \phi \in \mathcal{L}^{\pi}_{\infty,\infty}\colon p \models \phi \;\Rightarrow\; q \models \phi.$$

By Theorem 10.2.5 it follows that for finitary transition systems, $\lesssim^F$ and $\lesssim^B$ coincide, while, by the duality Theorem 10.2.7, it follows that the order of $\mathcal{D}$, which is equivalent to the specialization order of $\mathcal{O}(\mathcal{D})$, coincides with the finitary preorder $\lesssim^F$. Therefore, in $\mathcal{D}$, $d_1 \leq d_2$ if and only if $d_1 \lesssim^B d_2$.

To prove the completeness of the logic $\mathcal{L}^\pi_{\infty,\infty}$ for the class of finitary transition systems, define the syntactic equivalence $\sim$ on formulae of $\mathcal{L}^\pi_{\infty,\infty}$ by

$$\phi_1 \sim \phi_2 \quad \text{if and only if} \quad \mathcal{L}^\pi_{\infty,\infty} + (BN) + (FA) \vdash \phi_1 =_\pi \phi_2.$$

Also, define the Lindenbaum algebra $\mathcal{LA}^\pi_{\infty,\infty}$ to be the set of equivalence classes of formulae in $\mathcal{L}^\pi_{\omega,\infty}$ under $\sim$, ordered by

$$[\phi_1] \leq [\phi_2] \quad \text{if and only if } \mathcal{L}^\pi_{\infty,\infty} + (BN) + (FA) \vdash \phi_1 \leq_\pi \phi_2.$$

The logical axioms say that the poset $\mathcal{LA}^\pi_{\infty,\infty}$ is a completely distributive lattice. By Lemma 10.4.1 and with a proof similar to the proof of Lemma 10.3.3, it is not hard to see that $\mathcal{LA}^\pi_{\infty,\infty}$ enjoys universal properties.

10.4.2. Lemma. *The completely distributive lattice $\mathcal{LA}^\pi_{\infty,\infty}$ is free over the frame $\mathcal{LA}^\pi_{\omega,\infty}$.* □

By Theorem 9.3.1 it follows that the inclusion function

$$\iota : \mathcal{LA}^\pi_{\omega,\infty} \hookrightarrow \mathcal{LA}^\pi_{\infty,\infty}$$

is the free observation frame over $\mathcal{LA}^\pi_{\omega,\infty}$.

10.4.3. Lemma. *The assignment $[\phi] \mapsto [\![\phi]\!]^\pi_\mathcal{D}$ defines the unique order isomorphism $\gamma^\star : \mathcal{LA}^\pi_{\infty,\infty} \to \mathcal{Q}(\mathcal{D})$ such that $\gamma^\star([\phi]) = \gamma([\phi])$ for all $\phi \in \mathcal{L}^\pi_{\omega,\omega}$.*

Proof: Because $\mathcal{D}$ is an SFP-domain, if it is equipped with the Scott topology then it forms a sober space. Hence, by Theorem 9.3.6, the lattice of saturated sets $\mathcal{Q}(\mathcal{D})$ is the free completely distributive lattice over the frame of Scott open sets $\mathcal{O}(\mathcal{D})$, which, by Lemma 10.3.4, is order isomorphic to the Lindenbaum algebra $\mathcal{LA}^\pi_{\omega,\infty}$. But $\mathcal{LA}^\pi_{\infty,\infty}$ is the free completely distributive lattice over the frame $\mathcal{LA}^\pi_{\omega,\infty}$ (Lemma 10.4.2), and hence $\mathcal{Q}(\mathcal{D})$ is order isomorphic to $\mathcal{LA}^\pi_{\infty,\infty}$. The isomorphism is given by the unique extension $\gamma^\star$ of the function $\gamma^+ : \mathcal{LA}^\pi_{\omega,\infty} \to \mathcal{O}(\mathcal{D})$ which can be characterized (using Lemma 10.3.2, the soundness Theorem 10.2.6, and the duality Theorem 10.3.4) by

$$\gamma^\star([\phi]) = \bigcap_I \gamma^+([\phi_i]) = \bigcap_I [\![\phi_i]\!]^\pi_\mathcal{D} = [\![\bigwedge_I \psi_i]\!]^\pi_\mathcal{D} = [\![\phi]\!]^\pi_\mathcal{D},$$

for all ϕ in $\mathcal{L}^\pi_{\infty,\infty}$. □

As before, soundness of the logical system associated with $\mathcal{L}^\pi_{\omega,\infty}$ including both the finitary schemes (BN) and (FA) follows from Theorem 10.2.6 and from the definition of finitary transition systems. In a similar way to the proof of the completeness Theorem 10.2.8, completeness follows from the above duality result.

10.4.4. Theorem. *(Completeness) Let $\mathcal{FT}$ be any class of finitary transition systems containing $\mathcal{D}$. For ϕ_1 and ϕ_2 in $\mathcal{L}^{\pi}_{\infty,\infty}$, $\mathcal{FT} \models \phi_1 \leq \phi_2$ if and only if $\mathcal{L}^{\pi}_{\infty,\infty} + (BN) + (FA) \vdash \phi_1 \leq \phi_2$.* □

10.5 Concluding notes

In this chapter we extended Abramsky's finitary domain logic for transition systems to an infinitary domain logic, the latter being a syntactical representation of the lattice of saturated sets of a particular SFP-domain. The whole lattice of saturated sets however is too big to be represented by finite syntax. The same holds for the lattice of open sets. A better approach would be to extend Abramsky's finitary language with both a greatest fixed point operator and a least fixed point operator to describe some of the saturated sets.

The present chapter does not deal with a formal comparison between Abramsky's logic and Hennessy-Milner logic for transition systems. Such a comparison can be found in [Abr91a], where $\mathcal{L}_{\infty,\infty}$ is proved equivalent to the infinitary Hennessy-Milner logic in the sense that a process of a transition system satisfies a formula of Abramsky's logic if and only if it satisfies the equivalent formula in the Hennessy-Milner logic. As a consequence, the complete distributive lattice induced by the (ordinary) interpretation of the infinitary Hennessy-Milner logic coincides with the one induced by the infinitary Abramsky's logic. Thus the infinitary Hennessy-Milner logic extended with the axiom schemas (BN) and (FA) is isomorphic to the completely distributive lattice of saturated sets of the SFP-domain $\mathcal{D}$.

The techniques we used in this chapter are general and can be applied to every logic based on a topological interpretation. Below we explain how to get an infinitary extension of a finitary logic step by step.

Let $\mathcal{L}$ be a language closed with respect to the binary operations $\wedge$ and $\vee$, and such that both tt and ff are in $\mathcal{L}$. Assume that $\mathcal{L}$ comes equipped with a preorder $\lesssim$ and a set of logical axioms which gives to $(\mathcal{L}, \lesssim)$ the structure of a distributive lattice such that $\wedge$ defines the binary meet, $\vee$ defines the binary join, tt is the top element and ff is the bottom element. It follows that the Lindenbaum algebra $\mathcal{LA}$ of $\mathcal{L}$ is a distributive lattice and hence is isomorphic in **DLat** to the set of compact opens $\mathcal{KO}(X)$ for a spectral space X. This isomorphism defines a canonical interpretation function

$$[\![\cdot]\!]_{\mathcal{L}} : \mathcal{L} \to \mathcal{KO}(X)$$

mapping every element $p \in \mathcal{L}$ to the compact open that is isomorphic to the equivalence class corresponding to p in the Lindenbaum algebra of $\mathcal{L}$.

The goal is to extend $\mathcal{L}$ to a language $\mathcal{L}^E$ closed with respect to the infinitary operations $\bigwedge$ and $\bigvee$ in a way that a 'canonical' interpretation function $[\![\cdot]\!]_{\mathcal{L}^E}$ can be defined from $\mathcal{L}^E$ to the lattice $\mathcal{Q}(X)$ of saturated sets of X such that $[\![\cdot]\!]_{\mathcal{L}^E}$ restricted to $\mathcal{L}$ coincides with $[\![\cdot]\!]_{\mathcal{L}}$.

To this end axioms and rules must be defined on $\mathcal{L}^E$: logical axioms to give $(\mathcal{L}^E, \lesssim)$ the structure of a completely distributive lattice, and structural axioms to relate $\bigwedge$ and $\bigvee$ with the other constructors of $\mathcal{L}$. This last step is the 'creative part' of the enterprise. The criterion is that the resulting Lindenbaum algebra $\mathcal{LA}^E$ of the extended language $\mathcal{L}^E$ must be the free completely distributive lattice over the distributive lattice $\mathcal{LA}$. If this is the case then we can use

1. the characterization of spectral spaces in terms of the free frames of opens, and

2. the characterization of sober spaces in terms of the free completely distributive lattice of saturated sets

to obtain the canonical isomorphism

$$[\![\cdot]\!]_{\mathcal{L}^E} : \mathcal{L}^E \to \mathcal{Q}(X)$$

such that $[\![p]\!]_{\mathcal{L}^E} = [\![p]\!]_{\mathcal{L}}$ for every $p \in \mathcal{L}$.

It is a topic for future work to provide infinitary domain logics for the specific languages of Abramsky for properties, typed terms, and morphisms. They are interpreted, respectively, as compact opens of SFP-domains, elements of SFP-domains, and morphisms between SFP-domains freely generated by means of a language of type expressions.

Bibliography

[Abr87] S. Abramsky. *Domain theory and the logic of observable properties.* PhD thesis, Queen Mary College, University of London, 1987.

[Abr91a] S. Abramsky. A domain equation for bisimulation. *Information and Computation*, 92:161–218, 1991.

[Abr91b] S. Abramsky. Domain theory in logical form. *Annals of Pure and Applied Logic*, 51:1–77, 1991.

[AJ94] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume III - Semantic Structures. Clarendon Press, 1994.

[Ale37] P.S. Alexandrov. Diskrete Räume. *Mat.Sb.*, 1(43):501–519, 1937.

[Ale39] J.W. Alexander. Ordered sets, complexes and the problem of compactifications. *Proceedings of the National Academy of Sciences, USA*, 25:296–298, 1939.

[AN80] A. Arnold and M. Nivat. Metric interpretations of infinite trees and semantics of nondeterministic recursive programs. *Theoretical Computer Science*, 11(2):181–205, 1980.

[AP86] K.R. Apt and G.D. Plotkin. Countable nondeterminism and random assignment. *Journal of the ACM*, 33(4):724–767, October 1986.

[AR89] P. America and J.J.M.M. Rutten. Solving reflexive domain equations in a category of complete metric spaces. *Journal of Computer and System Sciences*, 39(3):343–375, 1989.

[AS85] B. Alpern and F.B. Schneider. Defining liveness. *Information Processing Letters*, 21:181–185, 1985.

[AV93] S. Abramsky and S.J. Vickers. Quantales, observational logic and process semantics. *Mathematical Structures in Computer Science*, 3:161–227, 1993.

[Bac78] R.-J.R. Back. *On the correctness of refinement steps in program development.* PhD thesis, Department of Computer Science, University of Helsinki, 1978. Report A-1978-4.

[Bac80] R.-J.R. Back. *Correctness Preserving Program Refinements: Proof Theory and Applications*, volume 131 of *Mathematical Centre Tracts.* Mathematical Centre, Amsterdam, 1980.

[Bac81] R.-J.R. Back. On correct refinement of programs. *Journal of Computer and System Sciences*, 23(1):49–68, 1981.

[Bac89] R.-J.R. Back. Changing data representation in the refinement calculus. In *Proceedings 21st Hawaii International Conference on System Sciences*, 1989.

[Bak75] J.W. de Bakker. Inleiding bewijsmethoden. In J.W. de Bakker, editor, *Colloquium Programmacorrectheid*, volume 21 of *Mathematical Centre Syllabus*, pages 1–17, 1975. (In Dutch).

[Bak80] J.W. de Bakker. *Mathematical Theory of Program Correctness.* Prentice-Hall, 1980.

[Ban22] S. Banach. Sur les operations dans les ensembles abstraits et leurs applications aux equations integrales. *Fundamenta Mathematicae*, 3:133–181, 1922.

[Bar84] H.P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics.* North-Holland, revised edition, 1984.

[BBR96a] M.M. Bonsangue, F. van Breugel, and J.J.M.M. Rutten. Generalized metric spaces: completion, topology, and powerdomains via the Yoneda embedding. Technical Report CS-R9636, CWI, Amsterdam, 1996.

[BBR96b] M.M. Bonsangue, F. van Breugel, and J.J.M.M. Rutten. Alexandroff and Scott topologies for generalized metric spaces. In S. Andima, R. C. Flagg, G. Itzkowitz, P. Misra, Y. Kong, and R. Kopperman, editors, *Papers on General Topology and Applications: Eleventh Summer Conference at University of Southern Maine*, Annals of the New York Academy of Sciences. New York Academy of Sciences, 1996. To appear.

[Ber63] C. Berge. *Topological Spaces - including a treatment of multi-valued functions, vector spaces and convexity.* Oliver & Boyd, 1963. English translation of ‘Espaces Topologiques: Fonctions Multivoques’ published by Dunod, Paris 1959.

[Bes83] E. Best. Relational semantics of concurrent programs (with some applications). In D. Bjørner, editor, *Proceedings of the IFIP Working Conference on on Formal Description of Programming Concepts - II*, pages 431–452, Garmisch-Partenkirchen, FRG, 1983. North-Holland Publishing Company.

[Bes89] E. Best. Towards compositional predicate transformer semantics for concurrent programs. In *J.W. de Bakker, 25 jaar Semantiek*, pages 111–117, CWI, Amsterdam, 1989.

[BHR94] J.W. de Bakker, E. Horita, and J.J.M.M. Rutten. Fully abstract denotational models for nonuniform concurrent languages. *Information and Computation*, 115(1):125–178, 1994.

[Bir67] G. Birkhoff. *Lattice Theory*, volume 25 of *AMS Colloquium Publications*. American Mathematical Society, 1967. Third edition.

[BJK95] M.M. Bonsangue, B. Jacobs, and J.N. Kok. Duality beyond sober spaces: topological spaces and observation frames. *Theoretical Computer Science*, 151(1):79–124, 1995.

[BK86] J.A. Bergstra and J.W. Klop. Algebra of communicating processes. In J.W. de Bakker, M. Hazewinkel, and J.K. Lenstra, editors, *Proccedings of the CWI symposioum Mathematics and Computer Science* volume 1 of *CWI Monographs*, pages 89–138. North-Holland, 1986.

[BK93a] M.M. Bonsangue and J.N. Kok. Isomorphisms between state and predicate transformers. In A.M. Borzyszkowski and S. Sokolowoski, editors, *Proceedings of MFCS '93, Gdansk, Poland*, volume 711 of *Lecture Notes in Computer Science*, pages 301–310. Springer-Verlag, 1993.

[BK93b] M.M. Bonsangue and J.N. Kok. Semantics, orderings and recursion in the weakest precondition calculus. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Rex Workshop '92 'Semantics: Foundations and Applications'*, volume 666 of *Lecture Notes in Computer Science*, pages 91–109. Springer-Verlag, 1993.

[BK94a] M.M. Bonsangue and J.N. Kok. Relating multifunctions and predicate transformers through closure operators. In Masami Hagiya and John C. Mitchell, editors, *TACS '94, Sendai, Japan*, volume 789 of *Lecture Notes in Computer Science*, pages 822–843. Springer-Verlag, 1994.

[BK94b] M.M. Bonsangue and J.N. Kok. The weakest precondition calculus: recursion and duality. *Formal Aspects of Computing*, 6A:788–800, 1994. Full version in *Formal Aspect of Computing*, 6(E):71–100, 1994. Available through anonymous ftp from ftp.cs.man.ac.uk as the (compressed PostScript) file /pub/fac/FACj_6E_p71.ps.Z.

[BK95] M.M. Bonsangue and M.Z. Kwiatkowska. Re-interpreting the modal μ-calculus. In A. Ponse, M. de Rijke, and Y. Venema, editors, *Modal Logic and Process Algebra*, volume 53 of *CSLI Lecture notes*, pages 65–83, Stanford, 1995. Centre for Study of Languages and Information.

[BKPR93] F.S. de Boer, J.N. Kok, C. Palamidessi, and J.J.M.M. Rutten. On blocks: locality and asynchronous communication. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Rex Workshop '92 'Semantics: Foundations and Applications'*, volume 666 of *Lecture Notes in Computer Science*, pages 73–90. Springer-Verlag, 1993.

[BKV94] M.M. Bonsangue, J.N. Kok, and E. de Vink. Metric predicate transformers: towards a notion of refinement for concurrent programs. Technical Report IR - 371, Vrije Universiteit Amsterdam - Department of Computer Science, 1994.

[BKV95] M.M. Bonsangue, J.N. Kok, and E. de Vink. Metric predicate transformers: towards a notion of refinement for concurrent programs. In I. Lee and S.A. Smolka, editors, *Proceedings of the 6th international conference Concur 95: Concurrency theory*, volume 962 of *Lecture Notes in Computer Science*, pages 363 – 377. Springer-Verlag, 1995.

[BR92] J.W. de Bakker and J.J.M.M. Rutten, editors. *Ten Years of Concurrency Semantics - selected papers of the Amsterdam Concurrency Group*. World Scientific, Singapore, 1992.

[Bre93] F. van Breugel. Relating state transformation semantics and predicate transformer semantics for parallel programs. Technical Report CS-9339, CWI, Amsterdam, 1993.

[Bre94] F. van Breugel. *Topological models in comparative semantics*. PhD thesis, Vrije Universiteit Amsterdam, 1994.

[BV96] J.W. de Bakker and E. de Vink. *Control Flow Semantics*. The MIT Press, 1996.

[BW90a] R.-J.R. Back and J. von Wright. Dualities in specification languages: a lattice theoretical approach. *Acta Informatica*, 27:583–625, 1990.

[BW90b] R.-J.R. Back and J. von Wright. Refinement calculus, part I: sequential nondeterministic programs. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Stepwise Refinement of Distributed Systems: Models, Formalisms, Correctness*, volume 430 of *Lecture Notes in Computer Science*, pages 42–66. Springer-Verlag, 1990.

[BW96] R.-J.R. Back and J. von Wright. *Refinement Calculus: a Systematic Introduction*. Preliminary version of a book submitted for publication, 1996.

[BWP87] M. Broy, M. Wirsing, and P. Pepper. On the algebraic definition of programming languages. *ACM Transactions on Programming Languages and Systems*, 9(1):54–99, 1987.

[BZ82] J.W. de Bakker and J.I. Zucker. Processes and the denotational semantics of concurrency. *Information and Control*, 54:70–120, 1982.

[Ca878] A. Cayley. The theory of groups. *American Journal of Mathematics*, 1:50–52, 1878.

[Che80] B.F. Chellas. *Modal Logic: an introduction*. Cambridge University Press, 1980.

[Chu32] A. Church. A set of postulates for the foundation of logic. *Annals of Mathematics*, 33:346–366, 1932.

[Col54] G.E. Collins. Distributivity and an axiom of choice. *Journal of Symbolic Logic*, 19:275–277, 1954.

[DG86] E.W. Dijkstra and A.J.M. van Gasteren. A simple fixpoint argument without the restriction to continuity. *Acta Informatica*, 23:1–7, 1986.

[Dij68] E.W. Dijkstra. Cooperating sequential processes. In F. Genuys, editor, *Programming Languages*, pages 43–112. Academic Press, London, 1968.

[Dij76] E.W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.

[DP90] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 1990.

[DS90] E.W. Dijkstra and C.S. Scholten. *Predicate Calculus and Program Semantics*. Springer-Verlag, New York, 1990.

[Dug66] J. Dugundji. *Topology*. Allyn and Bacon, inc., 1966.

[EF84] T. Elrad and N. Francez. A weakest precondition semantics for communicating processes. *Theoretical Computer Science*, 29(3):231–250, 1984.

[Egl75] H. Egli. A mathematical model for nondeterministic computations. Technical report, ETH Zurich, Switzerland, 1975.

[EM85] H. Ehrig and B. Mahr. *Fundamentals of Algebraic Specification I*, volume 6 of *EATCS monographs*. Springer-Verlag, 1985.

[Eng89] R. Engelking. *General Topology*, volume 6 of *Sigma Series in Pure Mathematics*. Heldermann Verlag, Berlin, 1989. Revised and completed edition.

[Esa74] L. Esakia. Topological Kripke models. *Soviet Mathematics Doklady*, 15:147–151, 1974.

[Flo67] R.W. Floyd. Assigning meaning to programs. In *Mathematical Aspects of Computer Science* volume 20 of *AMS Colloquium Publications*, pages 19–32, American Mathematical Society, 1967.

[FK96] B. Flagg and R. Kopperman. Continuity spaces: Reconciling domains and metric spaces - part I. To appear in *Theoretical Computer Science*, 1996.

[Fre77] G. Frege. Compound thoughts (Gedankefüge). In P.T. Geach, and R.H. Stoohoff, translators, *Logical Investigation. Gottlob Frege*, pages 55–78, Basil Blackwell, Oxford, 1977.

[FS81] L. Flon and N. Suzuki. The total correctness of parallel programs. *SIAM Journal on Computing*, 10(2):227–246, 1981.

[FS96] B. Flagg and P Sünderhauf. The essence of ideal completion in quantitative form. Draft, 1996.

[FW96] B. Flagg and K. Wagner. A logical approach to quantitative domain theory. Draft, 1996.

[Gai64] H. Gaifman. Infinite Boolean polynomials I. *Fundamenta Mathematicae*, 54:229–250, 1964. Errata in volume 57, page 117 (1965).

[GHK+80] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M.W. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, 1980.

[GM91] P. Gardiner and C. Morgan. Data refinement of predicate transformers. *Theoretical Computer Science*, 87(1):143–162, 1991.

[Gol89] R. Goldblatt. Varieties of complex algebras. *Annals of Pure and Applied Logic*, 44:173–242, 1989.

[Gor79] M. Gordon. *The Denotational Description of Programming Languages*. Springer-Verlag, New York, 1979.

[GR89] R.J. van Glabbeek and J.J.M.M. Rutten. The processes of de Bakker and Zucker represent bisimulation equivalence classes. In *J.W. de Bakker, 25 jaar Semantiek*, pages 243–246, CWI, Amsterdam, 1989.

[Gri81] D. Gries. *The Science of Programming*. Springer-Verlag, Berlin, 1981.

[GTW78] J.A. Goguen, J.W. Thatcher, and E.G. Wagner. An initial algebra approach to the specification, correctness and implementation of abstract data types. In R.T. Yeh, editor, *Current Trends in Programming Methodology IV: Data Structuring*, pages 80–114. Prentice-Hall, 1978.

[GTW+77] J.A. Goguen, J.W. Thatcher, E.G. Wagner, and J.B. Wright. Initial algebra semantics and continuous algebras. *Journal of the ACM*, 24:68–95, 1977.

[Gue81] I. Guessarian. *Algebraic Semantics*, volume 99 of *Lecture Notes in Computer Science*. Springer-Verlag, 1981.

[Haa81] V.H. Haase. Real-time behaviour of programs. *IEEE Transaction on Software Engineering*, SE-7(5):494–501, 1981.

[Hah48] H. Hahn. *Reelle Funktionen*, volume 1: Punktfunktionen. Chelsea Publishing Company, New York, 1948.

[Hal62] P.R. Halmos. *Algebraic Logic*. Chelsea, New York., 1962.

[Hal64] A.W. Hales. On the non-existence of free complete Boolean algebras. *Fundamenta Mathematicae*, 54:45–66, 1964.

[Har80] D. Harel. And/or programs: a new approach to structured programming. *ACM Transactions on Programming Languages and Systems*, 2(1):1–17, 1980.

[Hec91] R. Heckmann. Lower and upper power domain constructions commute on all cpos. *Information Processing Letters*, 40:7–11, 1991.

[Heh79] E.C.R. Hehner. Do considered od: a contribution to programming calculus. *Acta Informatica*, 11:287–304, 1979.

[Hen88] M. Hennessy. *Algebraic Theory of Processes.* The MIT Press, 1988.

[Hes88] W.H. Hesselink. An algebraic calculus of commands. Technical Report CS-8808, University of Groningen, 1988.

[Hes89] W.H. Hesselink. Predicate transformer semantics of general recursion. *Acta Informatica*, 26:309–332, 1989.

[Hes92a] W.H. Hesselink. Processes and formalisms for unbounded choice. *Theoretical Computer Science*, 99:105–119, 1992.

[Hes92b] W.H. Hesselink. *Programs, Recursion and Unbounded Choice*, volume 27 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1992.

[Hes94] W.H. Hesselink. Nondeterminacy and recursion via stacks and games. *Theoretical Computer Science*, 124(2):273–295, 1994.

[HK72] K.H. Hofmann and K. Keimel. A general character theory for partially ordered sets and lattices. *Memories of American Mathematical Society*, 122, 1972.

[HM81] K.H. Hofmann and M.W. Mislove. Local compactness and continuous lattices. In B. Banaschewski and R.-E. Hoffmann, editors, *Continuous lattices - Proceedings Bremen 1979*, volume 871 of *Lecture Notes in Mathematics*, pages 209–248. Springer-Verlag, 1981.

[HM85] M.C. Hennessy and R. Milner. Algebraic laws for non-determinism and concurrency. *Journal of the ACM*, 32:137–161, 1985.

[Hoa69] C.A.R. Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12:576–580, 1969.

[Hoa71] C.A.R. Hoare. Proofs of correctness of data representation. *Acta Informatica*, 1(4):271–281, 1971.

[Hoa78] C.A.R. Hoare. Some properties of predicate transformers. *Journal of the ACM*, 25:461–480, 1978.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes.* Prentice Hall, 1985.

[Hof79] R.-E. Hoffmann. Sobrification of partially ordered sets. *Semigroup Forum*, 17:123–138, 1979.

[HP72] P. Hitchcock and D. Park. Induction rules and termination proofs. In M. Nivat, editor, *Proceedings 1st International Colloquium of Automata, Languages and Programming; Rocquencourt, France*, pages 225–251. North-Holland, 1972.

[HP79] M. Hennessy and G.D. Plotkin. Full abstraction for a simple parallel programming language. In J. Becvar, editor, *Proceedings of the 8th MFCS*, volume 74 of *Lecture Notes in Computer Science*, pages 108–120. Springer-Verlag, 1979.

[HS87] C.A.R. Hoare and J. Sanders. Prespecification in data refinement. *Information Processing Letters*, 25:71–76, 1987.

[Isb72a] J.R. Isbell. Atomless parts of spaces. *Math. Scand.*, 31:5–32, 1972.

[Isb72b] J.R. Isbell. General functorial semantics. *American Journal of Mathematics*, 94:535–596, 1972.

[Joh81] P.T. Johnstone. Scott is not always sober. In B. Banaschewski and R.-E Hoffmann, editors, *Continuous lattices - Proceedings Bremen 1979*, volume 871 of *Lecture Notes in Mathematics*, pages 282–283. Springer-Verlag, 1981.

[Joh82a] P.T. Johnstone. *Stone Spaces.* Cambridge University Press, 1982.

[Joh82b] P.T. Johnstone. The Vietoris monad on the category of locales. In R.-E. Hoffman, editor, *Continuous lattices and related topics*, volume 27, pages 162–179. Mathematik-Arbeitspapiere, University of Bremen, 1982.

[Joh91] P.T. Johnstone. The art of pointless thinking: the category of locales. In H. Herrlich and H.-E. Porst, editors, *Category theory at work*, pages 85–107. Heldermann Verlag Berlin, 1991.

[JS96] A. Jung and P. Sünderhauf. On the duality of compact vs. open. In S. Andima, R. C. Flagg, G. Itzkowitz, P. Misra, Y. Kong, and R. Kopperman, editors, *Papers on General Topology and Applications: Eleventh Summer Conference at University of Southern Maine*, Annals of the New York Academy of Sciences, 1996. To appear.

[JT51] B. Jónsson and A. Tarski. Boolean algebras with operators, part I. *American Journal of Mathematics*, 73:891–939, 1951.

[JT52] B. Jónsson and A. Tarski. Boolean algebras with operators, part II. *American Journal of Mathematics*, 74:127–167, 1952.

[KK89] P. Knijnenburg and J.N. Kok. On the semantics of atomized statements: the parallel-choice option. In Budach, editor, *Proceedings of Fundamentals of Computation Theory '91*, volume 529 of *Lecture Notes in Computer Science*, pages 297–307. Springer-Verlag, 1991.

[Kle52] S.C. Kleene. *Introduction to Meta-Mathematics.* van Nostrand, New York, 1952.

[Kna28] B. Knaster. Un théorème sur les fonctions d'ensembles. *Ann. Soc. Polon. Math.*, 6:133–134, 1928.

[KR90] J.N. Kok and J.J.M.M. Rutten. Contractions in comparing concurrency semantics. *Theoretical Computer Science*, 76 (2/3):179–222, 1990.

[Kun80] K. Kunen. *Set Theory: An Introduction to Independence Proofs.* Volume 102 of *Studies in Logic and the Foundations of Mathematics*, 1980. North-Holland, Amsterdam.

[Kur56] K. Kuratowski. Sur une méthode de métrisation complète des certain espaces d'ensembles compacts. *Fundamenta Mathematicae*, 42:114–138, 1956.

[Kwi91] M.Z. Kwiatkowska. On topological characterization of behavioural properties. In G.M. Reed, A.W. Roscoe, and R.F. Wachter, editors, *Topology and Category Theory in Computer Sciences - Proceedings of the Oxford Topology Symposium, June 1990*, pages 153–177. Oxford Science Publications, 1991.

[Lam77] L. Lamport. Proving the correctness of a multiprocess program. *IEEE Transaction on Software Engineering*, SE-3:125–143, 1977.

[Lam90] L. Lamport. Win and sin: predicate transformers for concurrency. *ACM Transactions on Programming Languages and Systems*, 12(3):396–428, 1990.

[Law73] F.W. Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del Seminario Matematico e Fisico di Milano*, 43:135–166, 1973.

[Law86] F.W. Lawvere. Taking categories seriously. *Revista Colombiana de Matemáticas*, XX:147–178, 1986.

[Lib92] L. Libkin. An elementary proof that upper and lower powerdomain constructions commute. Volume 48 of *Bulletin of EATCS*, pages 175–177, 1992.

[Lin66] F.E.J. Linton. Some aspects of equational categories. In *Proceedings of Conference on Categorical Algebra, La Jolla 1965*, pages 84–94. Springer-Verlag, 1966.

[LNS82] J.-L. Lassez, V.L. Nguyen, and E.A. Sonenberg. Fixed point theorems and semantics: a folk tale. *Information Processing Letters*, 14(3):112–116, 1982.

[LS79] L. van Lamsweerde and M. Sintzoff. Formal derivation of strongly correct concurrent programs *Acta Informatica*, 12(1):1–31, 1979.

[Luk91] J.J. Lukkien. *Parallel program design and generalized weakest preconditions*. PhD thesis, Rijksuniversiteit Groningen, 1991.

[Luk94] J.J. Lukkien. Operational semantics and generalized weakest preconditions. *Science of Computer Programming*, 22:137–155, 1994.

[Man76] E.G. Manes. *Algebraic Theories*, volume 26 of *Graduate Texts in Mathematics*. Springer-Verlag, 1976.

[Mar73] G. Markowsky. *Combinatorial aspects of lattice theory with applications to the enumeration of free distributive lattices*. PhD thesis, Harvard University, 1973.

[Mar79] G. Markowsky. Free completely distributive lattices. *Proceedings of the American Mathematical Society*, 74(2):227–228, 1979.

[Mey85] J.-J.Ch. Meyer. *Programming calculi based on fixed point transformations: semantics and applications*. PhD thesis, Vrije Universiteit, Amsterdam, 1985.

[ML71] S. Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, 1971.

[Mic51] E. Michael. Topologies on spaces of subsets. *Transactions of the American Mathematical Society*, 71:152–182, 1951.

[Mil73] R. Milner. Processes: a mathematical model of computing agents. In H.E. Rose, and J.C. Shepherdson, editors, *Proceedings of Logic Colloquium 73*, pages 157–173. North-Holland, 1973.

[Mil80] R. Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1980.

[Mis91] M.W. Mislove. Algebraic posets, algebraic cpo's and models of concurrency. In G.M. Reed, A.W. Roscoe, and R. Wachter, editors, *Topology and Category Theory in Computer Science*, pages 75–111, Clarendon Press, 1991.

[Mis95] M.W. Mislove. Topology, domain theory and theoretical computer science. Unpublished note presented at Eleventh Summer Conference on Topology and Applications. Available through anonymous ftp from the machine 129.81.96.30 as the (compressed PostScript) file pub/mwn/topandcs.ps.gz.

[MO91] M.W. Mislove and F.J. Oles. A topological algebra for angelic nondeterminism. Technical Report RC 17344, IBM, 1991.

[MO92] M.W. Mislove and F.J. Oles. A simple language supporting angelic nondeterminism and parallel composition. In S. Brookes, M. Main, A. Melton, and M. Mislove editors, *Proceedings of 7th MFPS, Pittsburgh (USA)*, volume 598 of *Lecture Notes in Computer Science*, pages 77–101. Springer-Verlag, 1992.

[Mor87] J.M. Morris. A theoretical basis for stepwise refinement and the programming calculus. *Science of Computer Programming*, 9:287–306, 1987.

[Mor90] C.C. Morgan. *Programming from Specifications.* Prentice-Hall, 1990.

[Mos92] P. Mosses. *Action Semantics.* Cambridge University Press, 1992.

[MRS95] M.W. Mislove, A.W. Roscoe, and S.A. Schneider. Fixed points without completeness. *Theoretical Computer Science*, 138:273–314, 1995.

[MS76] R.E. Milne and C. Strachey. *A Theory of Programming Language Semantics.* Chapman and Hall, 1976.

[Nad78] S.B. Nadler. *Hyperspaces of Sets.* Pure and Applied Mathematics. Marcel Dekker, 1978.

[Nel89] G. Nelson. A generalization of Dijkstra's calculus. *ACM Transactions on Programming Languages and Systems*, 11(4):517–561, 1989.

[Nil82] N.J. Nilsson. *Principles of Artificial Intelligence.* Springer-Verlag, 1982.

[Par81] D.M.R. Park. Concurrency and automata on infinite sequences. In P. Deussen, editor, *Proceedings of the 5th GI Conference*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183. Springer-Verlag, 1981.

[Plo76] G.D. Plotkin. A powerdomain construction. *SIAM Journal on Computing*, 5:452–487, 1976.

[Plo79] G.D. Plotkin. Dijkstra's predicate transformer and Smyth's powerdomain. In D. Bjørner, editor, *Proceedings of the Winter School on Abstract Software Specification*, volume 86 of *Lecture Notes in Computer Science*, pages 527–553. Springer-Verlag, 1979.

[Plo81a] G.D. Plotkin. Post-graduate lecture notes in advanced domain theory (incorporating the 'Pisa notes'). Department of Computer Science, University of Edinburgh, 1981.

[Plo81b] G.D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, Computer Science Department, Aarhus University, 1981.

[Pnu77] A. Pnueli. The temporal logic of programs. In *Proceedings of the 18th IEEE Symposioum on the Foundations of Computer Science*, pages 46–57.

[PP58] D. Papert and S. Papert. Sue les treillis des ouverts et les paratopologies. In *Seminaire Ehresmann, topologie et geometrie differentielle*, 1958.

[Pra81] V.R. Pratt. Dynamic logic. In J.W. de Bakker and J. van Leeuwen, editors, *Proceedings of Foundations of Computer Science III, Part 2*, volume 109 of *Mathematical Centre Tracts*, pages 53–84. Mathematical Centre, Amsterdam, 1981.

[Ran52] G. Raney. Completely distributive complete lattices. In *Proceedings of the American Mathematical Society*, volume 3(4), pages 677–680. Menasha, Wis., and Providence, R.I., 1952.

[Rew96] I.M. Rewitzky. *A topological framework for program semantics.* PhD thesis, Department of Mathematics and Applied Mathematics, University of Cape Town, South Africa, 1996.

[Rob88] E. Robinson. Logical aspects of denotational semantics. In *Summer conference on Category Theory and Computer Science*, volume 83 of *Lecture Notes in Computer Science*, pages 238–253. Springer-Verlag, 1988.

[Roe76] W.P. de Roever. Dijkstra's predicate transformer, non-determinism, recursion, and termination. In *Proceedings of the 5th MFCS*, volume 45 of *Lecture Notes in Computer Science*, pages 472–481. Springer-Verlag, 1976.

[RT93] J.J.M.M. Rutten and D. Turi. On the foundations of final semantics: non-standard sets, metric spaces, partial orders. In J.W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *Rex Workshop '92 'Semantics: Foundations and Applications'*, volume 666 of *Lecture Notes in Computer Science*, pages 477–530. Springer-Verlag, 1993.

[Rut95] J.J.M.M. Rutten. Elements of generalized ultrametric domain theory. Technical Report CS-R9507, CWI, Amsterdam, 1995.

[Sc890] E. Schröder. *Vorlesunger über die Algebra der Logik*, volume I. B.G. Teubner, Leipzig, 1890. Republished in 1966 by Chelsea Publishing Co., New York.

[Sch93] A. Schalk. *Algebras for generalized power constructions*. PhD thesis, Technische Hochschule Darmstadt, Germany, 1993.

[Sco70] D.S. Scott. Outline of a mathematical theory of computation. In *Proceedings 4th Annual Princeton Conference on Information Sciences and Systems*, pages 169–176, 1970.

[Sco72] D.S. Scott. Continuous lattices. In *Toposes, Algebraic geometry and Logic*, volume 274 of *Lecture Notes in Mathematics*, pages 97–136. Springer-Verlag, 1972.

[Sco76] D.S. Scott. Data types as lattices. *SIAM Journal on Computing*, 5(2):522–587, 1976.

[Sco82] D.S. Scott. Domains for denotational semantics. In M. Nielsen and E.M. Schmidt, editors, *9th International Colloquium on Automata, Languages and Programming; Aarhus, Denmark*, volume 140 of *Lecture Notes in Computer Science*, pages 577–613. Springer-Verlag, 1982.

[Smy83a] M.B. Smyth. The largest cartesian closed category of domains. *Theoretical Computer Science*, 27:109–119, 1983.

[Smy83b] M.B. Smyth. Power domains and predicate transformers: a topological view. In J. Diaz, editor, *Proceedings 10th International Colloquium on Automata, Languages and Programming; Barcelona, Spain*, volume 154 of *Lecture Notes in Computer Science*, pages 662–675. Springer-Verlag, 1983.

[Smy87] M.B. Smyth. Quasi uniformities: reconciling domains with metric spaces. In M. Main, A. Melton, M. Mislove, and D. Schmidt, editors, *Proceedings of the 3rd Workshop on Mathematical Foundations of Programming Language Semantics*, volume 298 of *Lecture Notes in*

Computer Science, pages 236–253. Springer-Verlag, New Orleans, 1987.

[Smy91] M.B. Smyth. Totally bounded spaces and compact ordered spaces as domains of computation. In G.M. Reed, A.W. Roscoe, and R.F. Wachter, editors, *Topology and Category Theory in Computer Sciences - Proceedings of the Oxford Topology Symposium, June 1990*, pages 207–229. Oxford Science Publications, 1991.

[Smy92] M.B. Smyth. Topology. In S. Abramsky, D.M. Gabbay, and T.S.E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume I - Background: Mathematical Structures, pages 641–761. Clarendon Press, 1992.

[SP82] M.B. Smyth and G.D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM Journal on Computing*, 11:761–783, 1982.

[SS71] D.S. Scott and C. Strachey. Towards a mathematical semantics for computer languages. In *Proceedings of the Symposium on Computers and Automata*, volume 21 of *Microwave Research Institute Symposia series*, 1971.

[Sto36] M.H. Stone. The theory of representation for Boolean algebras. *Transactions of the American Mathematical Society*, 40:37–111, 1936.

[Sto37] M.H. Stone. Topological representation of distributive lattices and Brouwerian logics. *Cas. Math. Fys.*, 67:1–25, 1937.

[Sto77] J. Stoy. *Denotational Semantics: the Scott-Strachey Approach to Programming Language Theory.* The MIT press, 1977.

[Str91] B. Stroustrup. *The C++ Programming Language* Addison-Wesley, 1991.

[SZ92] D. Scholefield and H.S.M. Zedan. Weakest precondition semantics for time and concurrency. *Information Processing Letters*, 43(6):301–308, 1992.

[Tar35] A. Tarski. Zur Grundlegung der Boole'schen Algebra. *Fundamenta Mathematicae*, 24:177–198, 1935.

[Tar55] A. Tarski. A lattice theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–310, 1955.

[Tur49] A.M. Turing. On cheking a large routine. In *Report on the Conference on High-Speed Automatic Calculating Machines*, pages 67–69, University Mathematical Laboratory, Cambridge, 1949.

[US93] A. Udaya Shankar. An introduction to assertional reasoning for concurrent systems. *ACM Computing Surveys*, 25:225–262, 1993.

[Vic89] S.J. Vickers. *Topology via Logic*, volume 5 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

[Wag94] K.R. Wagner. *Solving domain equations with enriched categories*. PhD thesis, Carnegie Mellon University, Pittsburgh, July 1994. Technical report CMU-CS-94-159.

[Wan77] M. Wand. A characterization of weakest preconditions. *Journal of Computer and System Sciences*, 15:209–212, 1977.

[Win93] G. Winskel. *The Formal Semantics of Programming Languages, an introduction*. Foundations of Computing Series. The MIT Press, 1993.

[Wir71] N. Wirth. The design of a Pascal compiler. *Software–Practice and Experience*, 1(4):309–333, 1971.

[Wri90] J. von Wright. *A lattice-theoretical basis for program refinement*. PhD thesis, Åbo Akademi, 1990.

[Wri94] J. von Wright. The lattice of data refinement. *Acta Informatica*, 31(2):105–135, 1994.

[Zha91] G.-Q. Zhang *Logic of Domains*. Progress in Theoretical Computer Science, 1991.

[Zwi87] J. Zwiers. *Compositionality, Concurrency and Partial Correctness*. volume 321 of *Lecture Notes in Computer Science*, pages 1–272. Springer-Verlag, 1987.

Selected notation

$(v \in) IVar$	Set of individual variables.
$(e \in) Exp$	Set of expressions.
$(b \in) BExp$	Set of Boolean expressions.
$(x \in) PVar$	Set of procedure variables.
Val	Set of values.
$(s, t \in) \mathtt{St}$	Set of states.
$\mathcal{E}_\mathcal{V}$	Valuation of expressions.
$\mathcal{B}_\mathcal{V}$	Valuation of Boolean expressions.
$(S \in) Stat$	Collection of statements.
$(G \in) GStat$	Collection of guarded statements.
$(d \in) Decl$	Collection of declarations.
$\mathcal{L} = Decl \times Stat$	Programming language.
$\mathcal{P}(X)$	Collection of all subsets of X.
$\mathcal{P}_{\mathit{fin}}(X)$	Collection of all finite subsets of X.
$\mathcal{P}_{co}(X)$	Compact powerdomain of a metric space X.
$\mathcal{P}_{cl}(X)$	Closed powerdomain of a metric space X.
$\mathcal{P}_l(X)$	Lower powerspace of a topological space X.
$\mathcal{P}_u(X)$	Upper powerspace of a topological space X.
$\mathcal{P}_u^{co}(X)$	Compact upper powerspace of a topological space X.
$\mathcal{P}_c(X)$	Convex powerspace of a topological space X.

$\mathcal{P}_c^{co}(X)$	Compact convex powerspace of a topological space X.
$\mathcal{H}(X)$	Hoare powerdomain of an algebraic cpo X.
$\mathcal{S}(X)$	Smyth powerdomain of an algebraic cpo X.
$\mathcal{E}(X)$	Plotkin powerdomain of an algebraic cpo X.
$(o, u \in)\mathcal{O}(X)$	Topology on a set X.
$\mathcal{O}_S(X)$	Scott topology on a dcpo X.
$\mathcal{O}_A(X)$	Alexandrov topology on a poset X.
$\mathcal{KO}(X)$	Compact open subsets of $(X, \mathcal{O}(X))$.
$(q \in)\mathcal{Q}(X)$	Saturated subsets of $(X, \mathcal{O}(X))$.
$\mathcal{KQ}(X)$	Compact saturated subsets of $(X, \mathcal{O}(X))$.
$(c \in)\mathcal{C}(X)$	Closed subsets of $(X, \mathcal{O}(X))$.
L_o	Sub-basic open in the lower topology.
U_o	Basic open in the upper topology.
$Fr(X)$	Free frame over a set X.
$Fr\langle G \mid R\rangle$	Frame presented by a set of generators G and a set of relations R.
$CDL(X)$	Free completely distributive lattice over a set X.
$CDL\langle G \mid R\rangle$	Completely distributive lattice presented by a set of generators G and a set of relations R.
$\overline{F}$	Free completely distributive lattice over a frame F.
$f^{\odot} : Fr(X) \to CDL(Y)$	Free observation frame over an onto function $f{:}X \to Y$.
$Pt(\alpha)$	$\mathcal{T}_0$ space induced by an observation frame α.
$Pt_\omega(F)$	Sober space induced by a frame F.
$\Omega(X)$	Observation frame induced by a space X.
$\mathcal{O}(X)$	Frame induced by a space X.
$(\sigma, \tau \in)ST(X, Y)$	State transformers for specification from X to Y.
$(\sigma, \tau \in)ST_H(X, Y)$	Hoare state transformers from X to Y.

$(\sigma, \tau \in) ST_S(X, Y)$	Smyth state transformers from X to Y.
$(\sigma, \tau \in) ST_S^{fin}(X, Y)$	Finitary Smyth state transformers from X to Y.
$(\sigma, \tau \in) ST_E(X, Y)$	Egli-Milner state transformers from X to Y.
$(\sigma, \tau \in) ST_E^{fin}(X, Y)$	Finitary Egli-Milner state transformers from X to Y.
$(\sigma, \tau \in) \Sigma_2$	Metric state transformers with resumptions.
$(\pi, \rho \in) PT(Y, X)$	Predicate transformers from predicates on Y to predicates on X.
$(\pi, \rho \in) PT_M(Y, X)$	Monotonic predicate transformers from predicates on Y to predicates on X.
$(\pi, \rho \in) PT_T(Y, X)$	Total correctness predicate transformers from predicates on Y to predicates on X.
$(\pi, \rho \in) PT_P(Y, X)$	Partial correctness predicate transformers from predicates on Y to predicates on X.
$(\pi, \rho \in) MPT(Y, X)$	Metric predicate transformers from predicates on Y to predicates on X.
$(\pi, \rho \in) \Pi_2$	Metric predicate transformers with resumptions.
$\lesssim$	Preorder.
$\lesssim_{\mathcal{O}}$	Specialization preorder on topological space.
$\lesssim^B$	Partial bisimulation (for transition systems).
$\lesssim^\omega$	Observable equivalence (for transition systems).
$\lesssim^F$	Finitary preorder (for transition systems).
$P \subseteq_{fin} Q$	P is a finite subset of the set Q.
P^{tc}	Enhancement for total correctness of a predicate P.
P^{pc}	Enhancement for partial correctness of a predicate P.
P^{lin}	Linear enhancement of a predicate P.
$B_\epsilon(x)$	ϵ-ball centered in x.
$(b \in) \mathcal{K}(X)$	Compact (finite) elements of a dcpo X.
$Idl(X)$	Ideal completion of a poset X.

$(n, m \in)\mathbb{N}$	Set of natural numbers.
ω_0	Cardinality of the set of natural number.

Index

Abstract
Topological dualities in semantics

The formal semantics of a programming language consists of assigning to every program of the language an element of a mathematical structure. In this thesis we study the relationship between two different approaches to define the semantics of a program, namely the denotational and the axiomatic one.

The denotational semantics characterizes programs as elements of some mathematical domain in a compositional way: the semantics of a language construct is defined in terms of its components. Due to the possibility of self-application given by some programming languages, the semantic domain must sometimes be defined in a recursive way.

The axiomatic semantics characterizes programs in a logical framework intended for reasoning about programs properties: computations are expressed by relating programs to assertions about their behaviour.

In this thesis we study different transformations which ensure the correctness of one semantics in terms of the other. These transformations form dualities rather than equivalences. This is due to the fact that denotationally programs are identified with functions which transform states on the input space to (sets of) states of the output space, whereas axiomatically programs can be expressed as functions which transform predicates on the output space to predicates on the input space.

The dualities between the denotational and the axiomatic views of a program are topological because they are set in a topological framework: topological spaces are data-types and continuous functions between topological spaces are computations. These interpretations form the basis for a systematic development of a propositional program logic from a denotational semantics.

In the first part of the thesis we consider predicates as subsets of an abstract set of states, and we study several semantic model of sequential languages. We begin by considering the weakest precondition and the weakest liberal precondition semantics. Then we relate them to three denotational models based on state transformation. The relationships between these axiomatic and denotational models generalize the duality of Plotkin between Dijkstra's predicate transformers and the Smyth powerdomain.

Then we extend sequential languages with specification constructs. We use the language of Back's refinement calculus which supports two kinds of unbounded non-determinism. Traditionally, the semantics of the refinement calculus is based on monotonic predicate transformers. Beside it, we give a denotational semantics based on state transformations, and an operational semantics based on a hyper transition system. We relate the three models as follows: the operational semantics coincides with the denotational semantics which, in turn, is dual to the predicate transformer semantics.

In the second part of the thesis, in order to study the semantics of concurrent languages, we refine the notion of predicates by considering affirmative predicates. They are open subsets of an abstract set of states equipped with a topology. This permits us to define dualities between the upper, lower and Vietoris powerspace constructions, and topological predicate transformers.

One of the above dualities is applied to prove the correctness of a new compositional predicate transformer semantics for a concurrent language. The semantics domain is a metric space which is shown to be isometric to the resumption domain of De Bakker and Zucker. Partial and total correctness, and also temporal properties are studied for this metric predicate transformer semantics.

Finally, in the third part of the thesis, we make an abstraction step by regarding predicates as elements of an abstract algebra. We consider a topological space as a function from the abstract set of affirmative predicates (with algebraic operations representing arbitrary unions and finite intersections) to the abstract set of specifications (with algebraic operations representing arbitrary unions and arbitrary intersections). We call this function an observation frame. We first show that topological spaces can be reconstructed from observation frames, and then we prove that observation frames are algebraic structures in a precise categorical sense.

The above theory is applied to extend the finitary domain logic of Abramsky to an infinitary one preserving completeness. As an example we extend Abramsky's finitary domain logic for transition systems to an infinitary logic with arbitrary conjunctions and arbitrary disjunctions. Our extension is conservative in the sense that the domain represented in logical form by the infinitary logic coincides with the domain represented in logical form by Abramsky's finitary logic. As a consequence we obtain soundness and completeness of the infinitary logic for the class of all finitary transition systems.

Samenvatting
Topologische dualiteit in semantiek

De formele semantiek van een programmeertaal komt tot stand door aan ieder programma van die taal een element van een wiskundige structuur toe te kennen. In dit proefschrift bestuderen we de relatie tussen twee verschillende manieren waarop we een dergelijke semantiek kunnen definiëren: de denotationele en de axiomatische.

De denotationele semantiek wordt gekenmerkt door compositionaliteit: de semantiek van een programma wordt gedefinieerd in termen van zijn samenstellende delen. Aangezien sommige programmeertalen de mogelijkheid bieden tot zelf-applicatie, moeten we het semantische domein soms op een recursieve manier definiëren.

Een axiomatische semantiek karakteriseert programma's binnen een logisch formalisme dat bedoeld is om over de eigenschappen van die programma's te redeneren, waarbij berekeningen beschreven worden als relaties tussen enerzijds programma's en anderzijds asserties over hun gedrag.

In dit proefschrift bestuderen we verschillende transformaties van de denotationele semantiek in de axiomatische en vice versa. Deze transformaties vormen geen equivalenties, maar dualiteiten. Dit is het gevolg van het feit dat de denotationele semantiek van een programma begintoestanden overvoert in (verzamelingen van) eindtoestanden, terwijl in de axiomatische benadering een programma wordt beschreven als een functie die predikaten op eindtoestanden overvoert in predikaten op begintoestanden.

De dualiteit tussen denotationele en axiomatische semantiek formuleren we binnen een topologische context, waarbij topologische ruimten worden gezien als data-typen en de continue functies tussen topologische ruimten als berekeningen. Deze interpretaties vormen de basis voor een systematische afleiding van een propositionele programma-logica uit denotationele semantiek.

In het eerste deel van dit proefschrift beschouwen we predikaten als deelverzamelingen van een abstracte verzameling van toestanden en bestuderen we verschillende semantische modellen van sequentiële talen. Allereerst bekijken we de zwakste en de zwakste liberale preconditie-semantiek. Daarna vergelijken we deze axiomatische modellen met drie denotationele modellen die gebaseerd zijn op toestandstranformaties. De relaties tussen deze axiomatische en denotationele modellen generaliseren Plotkins dualiteit tussen Dijkstra's predikaattransformaties en het 'Smyth powerdomain'.

Vervolgens breiden we eerdergenoemde sequentiële talen uit met constructies voor het specificeren van programma's. We gebruiken de taal van Backs verfijningscalculus, die twee soorten van onbegrensd non-determinisme kent. Oorspronkelijk is de semantiek van deze calculus gebaseerd op monotonische predikaattransformaties. Daarnaast geven we zowel een denotationele se-

mantiek, die gebaseerd is op toestandstranformaties, als een operationele semantiek, die gebaseerd is op een hypertransitiesysteem. Voorts beschrijven we het verband tussen deze drie semantische modellen voor de verfijningscalculus: er is een equivalentie tussen het operationele model en het denotationele model, en er is een dualiteit tussen het denotationele model en het model dat gebaseerd is op predikaattransformaties.

In het tweede deel van dit proefschrift komen we tot een verfijning van het begrip predikaat door affirmatieve predikaten te beschouwen, zodat we ook de semantiek van parallelle talen kunnen bestuderen. Deze affirmatieve predikaten zijn de open deelverzamelingen van een abstracte verzameling van toestanden die voorzien is van een topologie. Dit stelt ons in staat dualiteiten te definiëren tussen topologische hyperruimten enerzijds en topologische predikaattransformaties anderzijds.

Eén van deze dualiteiten gebruiken we om de correctheid te bewijzen van een nieuwe, compositionele predikaattransformatie-semantiek voor een parallelle taal. Het domein van deze semantiek is een metrische ruimte, waarvan we aantonen dat zij isometrisch is met het resumptiedomein van De Bakker en Zucker. Verder bestuderen we partiële en totale correctheid, en ook temporele eigenschappen.

In het derde deel van dit proefschrift nemen we een nog abstracter standpunt in door predikaten op te vatten als elementen van een abstracte algebra. We beschouwen een topologische ruimte als een functie van de abstracte verzameling van affirmatieve predikaten (met algebraïsche operaties die willekeurige verenigingen en eindige doorsneden representeren) naar de abstracte verzameling van specificaties (met willekeurige verenigingen en willekeurige doorsneden). Deze functie noemen we een 'observation frame'. Allereerst laten we zien dat topologische ruimten gereconstrueerd kunnen worden uit zulke 'observation frames'. Vervolgens bewijzen we dat 'observation frames' algebraïsche structuren zijn in categorische zin.

De hieruit voortvloeiende theorie kan worden gebruikt om de eindige domeinlogica van Abramsky uit te breiden tot een oneindige logica, waarbij de volledigheid wordt behouden. Bij wijze van voorbeeld breiden we Abramsky's eindige domeinlogica voor transitiesystemen uit tot een oneindige logica met willekeurige conjuncties en disjuncties. Deze uitbreiding is in die zin conservatief, dat het domein dat in zijn logische vorm wordt gerepresenteerd door de oneindige logica, samenvalt met het domein dat in zijn logische vorm wordt gerepresenteerd door de eindige logica van Abramsky. Hieruit volgt dat de oneindige logica voor de klasse van alle eindige transitiesystemen zowel gezond als volledig is.