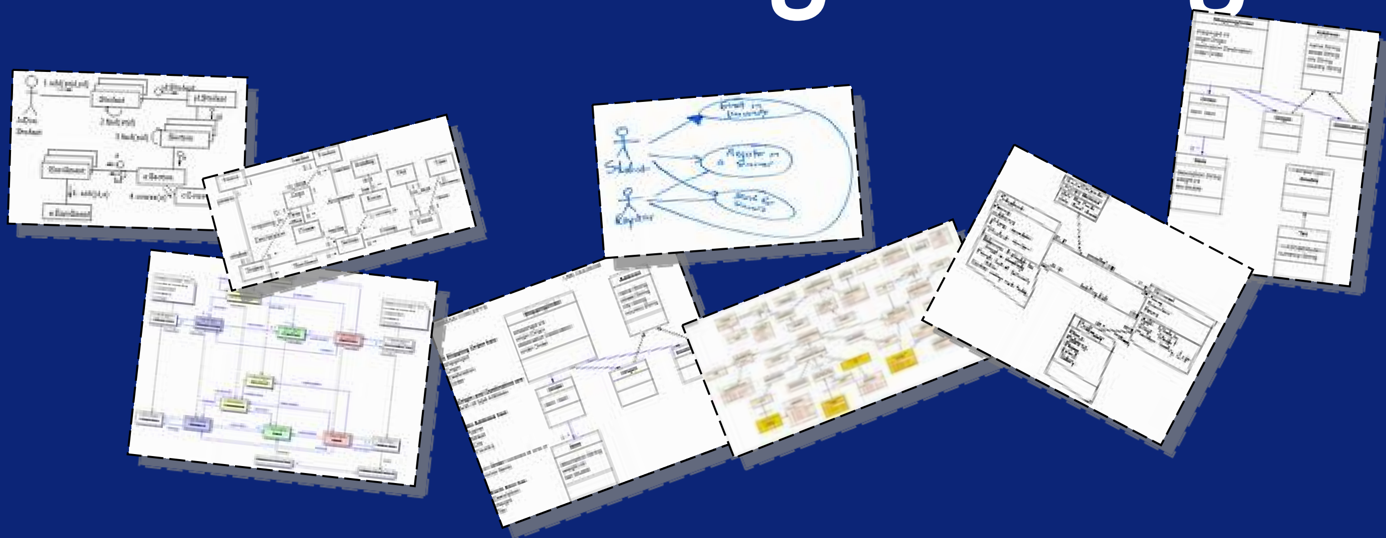
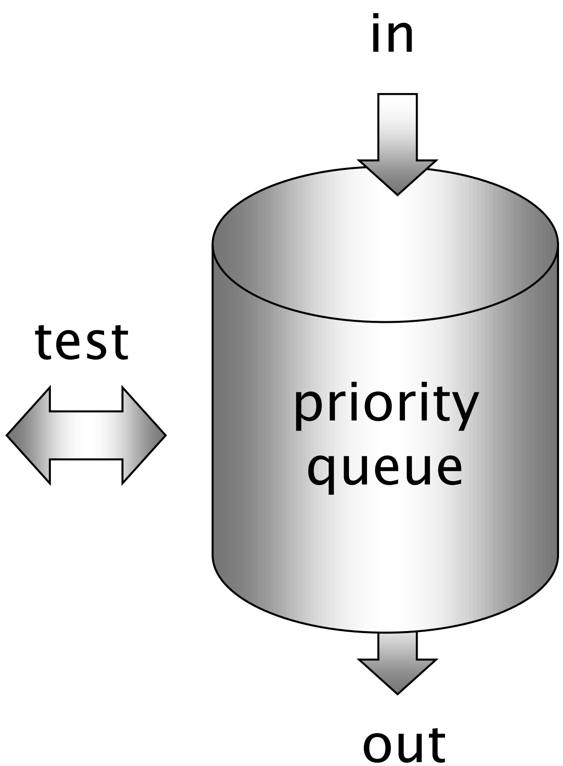
Formal Methods in Software Engineering



Hendrik Jan Hoogeboom, LIACS Algorithms & TCS

Universiteit Leiden

abstract data type



store, sort, transform & retrieve data

implementation

specification

application

interface

systems in general

languages

	Sequential	Concurrent
Algebraic	Larch (Guttag, *et al.*, 1993), OBJ (Futatsugi, *et al.*, 1985)	Lotos (Bolognesi and Brinksma, 1987),
Model-based	Z (Spivey, 1992) VDM (Jones, 1980) B (Wordsworth, 1996)	CSP (Hoare, 1985) Petri Nets (Peterson, 1981)

Figure 27.4 Formal specification languages

goal: reading a specification (in Z)

```
roomsInfo
  status: ROOM ⇸ GUEST

  Init
    status = ∅

  Check-in
    Δ status
    r?: ROOM
    g?: GUEST
    ----------
    r? ∉ dom status
    status′=status ∪ {r? ↦ g?}

  Check-out
    Δ status
    r!: ROOM
    g!: GUEST
    ----------
    r!∈dom status ∧ g!∈ran status
    status′=status \ {r! ↦ g!}
```

```
HtlRooms
  rooms: ℙ ROOM
  occupied: ℙ ROOM
  ----------
  occupied ⊆ rooms

  CkIn
    Δ occupied
    r?: ROOM
    ----------
    r? ∈ rooms ∧ r? ∉ occupied
    occupied' = occupied ∪ {r?}

  CkOut
    Δ occupied
    r?: ROOM
    ----------
    r? ∈ occupied
    occupied'=occupied \ {r?}

  ListVac
    rms!: ℙ ROOM
    ----------
    rms!=rooms \ occupied
```

wikipedia on “Z notation”

Z notation (/ˈzɛd/), … after Zermelo–Fraenkel set theory, is a formal specification language for describing and modelling computing systems. … clear specification of computer programs and computer-based systems

proposed by Jean-Raymond Abrial (1938–) in 1977 with the help of Steve Schuman and Bertrand Meyer. developed further at Oxford University

http://www.usherbrooke.ca/

Abrial answers the question "*Why Z?*" with "*Because it is the ultimate language!*“

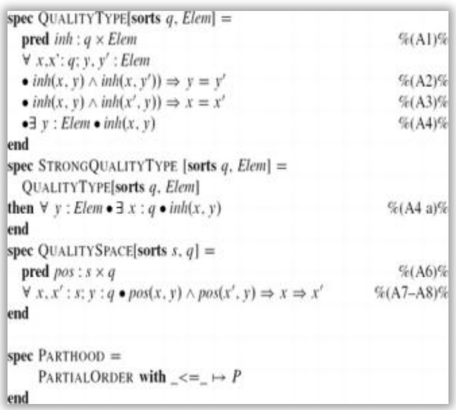
Z is based on the standard mathematical notation used in axiomatic set theory, lambda calculus, and first-order predicate logic.

… Z notation uses many non-ASCII symbols, …

FM perspective on Software

software as a **mathematical object**:

– function or relation or predicate



```
spec QualityType[sorts q, Elem] =
  pred inh : q × Elem                                   %(A1)%
  ∀ x,x': q; y, y' : Elem
  • inh(x, y) ∧ inh(x, y')) ⇒ y = y'                    %(A2)%
  • inh(x, y) ∧ inh(x', y)) ⇒ x = x'                    %(A3)%
  •∃ y : Elem • inh(x, y)                               %(A4)%
end
spec StrongQualityType [sorts q, Elem] =
  QualityType[sorts q, Elem]
then ∀ y : Elem • ∃ x : q • inh(x, y)                   %(A4 a)%
end
spec QualitySpace[sorts s, q] =
  pred pos : s × q                                      %(A6)%
  ∀ x, x' : s; y : q • pos(x, y) ∧ pos(x', y) ⇒ x ⇒ x'  %(A7–A8)%
end

spec Parthood =
    PartialOrder with _<=_ ↦ P
end
```

mathematical techniques for analyzing can be more powerful than human

why formal methods?

- eliminate ambiguity in specifications
  - avoid reasoning errors
  - prove that an implementation satisfies a specification

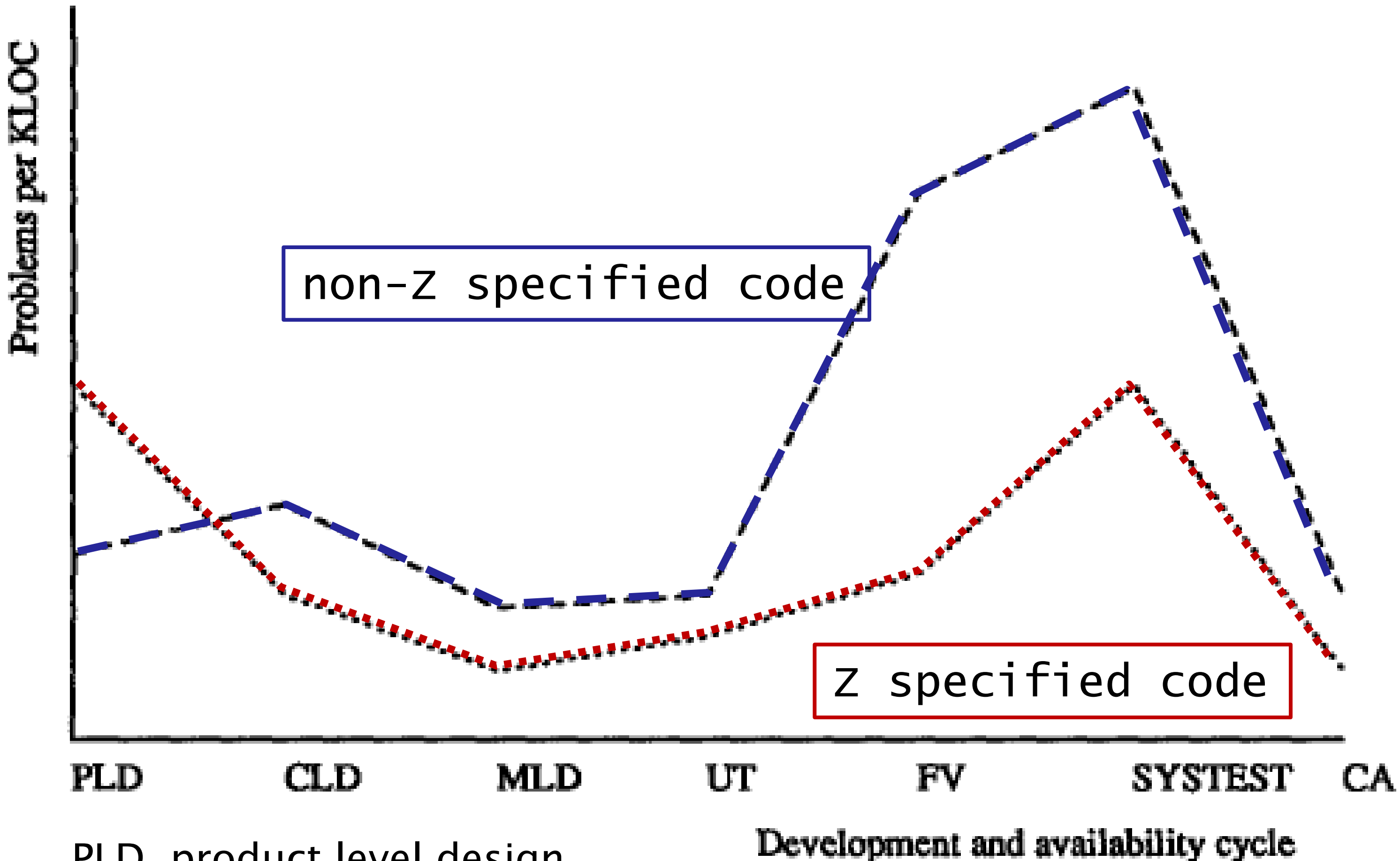
- systematic deduction of programs based on specification

  'correctness by construction'

  start with specification (predicate logic)

- also: basis for many automated QA techniques:
  - model checking (SPIN)
  - automated (model-based) testing

experiences … the use of Z



PLD product level design
CLD component level design
MLD module level design
UT unit test
FV functional verification
STSTEST system test
CA customer availability

CICS project report. Experiences and results from the use of Z in IBM Iain Houston, Steve King. VDM'91 Formal Software Development Methods. LNCS 1991, Volume 551/1991, 588-596, DOI: 10.1007/3-540-54834-3_34

language to learn

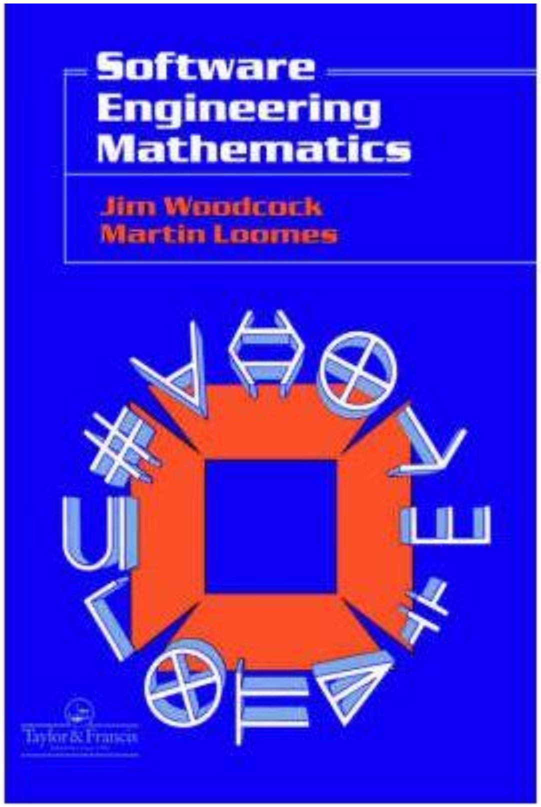
specification & reasoning

- logic: propositional, predicate

data types

- sets, relations, functions

references



Software Engineering Mathematics
Jim Woodcock, Martin Loomes
Pitman / Taylor and Francis, 1988

Software Specification
MohammadReza Mousavi,
Michel Reniers
lecture notes TUE 2009 (ch.4 Z)

propositional logic

- language (syntax)
- meaning (semantics) $\models$
- calculus ('automated' proofs) $\vdash$

natural vs. formal

ambiguities

tomorrow it will rain	time/person dependent
coffee or tea?	(exclusive) or
if it rains I will get wet	expect alternative?
I get wet because it rains	causality & ‘modalities’

all this we want to avoid

proposition

" statement of an alledged fact which must be either true or false "

p $\stackrel{\text{def}}{=}$ power indicator is on
q $\stackrel{\text{def}}{=}$ unit is connected to electricity

compound statements, conjunctions

unit is not connected to electricity
or
if power indicator is on
then unit is connected to electricity

(not q) or (if p then q)

$(\neg q) \vee (p \rightarrow q)$

connectives

developing a language

language: propositional logic

$(\neg q) \vee (p \rightarrow q)$

syntax – form

alphabet – symbols

grammar

semantics – meaning

truth tables

propositional logic

symbols of the language

- *constants* F 0 $\bot$ false
  T 1 $\top$ true
- *propositional symbols*
  $p, q, r, \ldots, p_1, p_2, \ldots$
- *brackets* (,)
- *connectives*

not…	$\neg$	'	$\overline{\cdot}$	negation	$\neg p$ p' $\overline{p}$
…or…	$\vee$	$+$	$\mid$	disjunction	
…and…	$\wedge$	$\cdot$	&	conjunction	
…implies…/if…then…	$\rightarrow$	$\supset$	$\Rightarrow$	implication	
…iff…	$\leftrightarrow$	$=$	$\Leftrightarrow$	biimplication	

propositional logic: syntax

- well formed formula *wff*

$q \qquad p$

$(\neg q) \qquad (p \to q)$

$((\neg q) \vee (p \to q)) \qquad (\neg(q \vee p)) \to q$

brackets!

- grammar

$X ::= T \mid F \mid P \mid$
$\quad (\neg X) \mid (X \vee X) \mid (X \wedge X) \mid (X \to X) \mid (X \leftrightarrow X)$

$P ::= p \mid q \mid \ldots$

I am not consistent in my bracketing

propositional logic

- language (syntax)
- meaning (semantics) $\models$
- calculus ('automated' proofs) $\vdash$

semantics

- truth tables

not

p	¬p
T	F
F	T

and

p	q	p∧q
T	T	T
T	F	F
F	T	F
F	F	F

or

p	q	p∨q
T	T	T
T	F	T
F	T	T
F	F	F

implies

p	q	p→q
T	T	T
T	F	F
F	T	T
F	F	T

iff

p	q	p↔q
T	T	T
T	F	F
F	T	F
F	F	T

equivalence ⇔

truth tables

inside out

or		∨
T	T	T
T	F	T
F	T	T
F	F	F

and		∧
T	T	T
T	F	F
F	T	F
F	F	F

iff		↔
T	T	T
T	F	F
F	T	F
F	F	T

p	q	r	p∧	(q∨r)
T	T	T	T	T
T	T	F	T	T
T	F	T	T	T
T	F	F	F	F
F	T	T	F	T
F	T	F	F	T
F	F	T	F	T
F	F	F	F	F
			2	1

truth tables

or		∨
T	T	T
T	F	T
F	T	T
F	F	F

and		∧
T	T	T
T	F	F
F	T	F
F	F	F

iff		↔
T	T	T
T	F	F
F	T	F
F	F	T

p	q	r	(p∧	(q∨r))	↔	((p∧q)	∨	(p∧r))
T	T	T	T	T	T	T	T	T
T	T	F	T	T	T	T	T	F
T	F	T	T	T	T	F	T	T
T	F	F	F	F	T	F	F	F
F	T	T	F	T	T	F	F	F
F	T	F	F	T	T	F	F	F
F	F	T	F	T	T	F	F	F
F	F	F	F	F	T	F	F	F
			2	1		1	3	2

equivalent

$$p\wedge(q\vee r) \Leftrightarrow (p\wedge q) \vee (p\wedge r)$$

tautology

$$\models (p\wedge(q\vee r))\leftrightarrow((p\wedge q)\vee(p\wedge r))$$

algebraic equivalences

in elementary arithmetic

$17+29+33+11 = (17+33)+(29+11)$

$x+y = y+x$

$(x+y)+z = x+(y+z)$

$17*98 = 17*(100-2) = 17*100-17*2$

$x*(y+z) = x*y+x*z$

$x*0 = 0 \quad x+0 = x$

$x*1 = x$

algebraic equivalences

for any wff X,Y,Z

commutative

$$X \vee Y \Leftrightarrow Y \vee X$$

associative

$$(X \vee Y) \vee Z \Leftrightarrow X \vee (Y \vee Z)$$

distributive

$$X \vee (Y \wedge Z) \Leftrightarrow (X \vee Y) \wedge (X \vee Z)$$

idempotence

$$X \Leftrightarrow X \vee X$$

DeMorgan's Theorem

$$\neg(X \wedge Y) \Leftrightarrow \neg X \vee \neg X$$

zero/false

$$X \vee F \Leftrightarrow X$$

unit/true

$$X \vee T \Leftrightarrow T$$

involution

$$\neg(\neg X) \Leftrightarrow X$$

complement

$$X \vee \neg X \Leftrightarrow T$$

algebraic equivalences

for any wff X,Y,Z

commutative

$$X \wedge Y \Leftrightarrow X \wedge Y \qquad X \vee Y \Leftrightarrow Y \vee X$$

associative

$$(X \wedge Y) \wedge Z \Leftrightarrow X \wedge (Y \wedge Z) \qquad (X \vee Y) \vee Z \Leftrightarrow X \vee (Y \vee Z)$$

distributive

$$X \wedge (Y \vee Z) \Leftrightarrow (X \wedge Y) \vee (X \wedge Z) \qquad X \vee (Y \wedge Z) \Leftrightarrow (X \vee Y) \wedge (X \vee Z)$$

idempotence

$$X \wedge X \Leftrightarrow X \Leftrightarrow X \vee X$$

DeMorgan's Theorem

$$\neg(X \vee Y) \Leftrightarrow \neg X \wedge \neg Y \qquad \neg(X \wedge Y) \Leftrightarrow \neg X \vee \neg X$$

zero/false

$$X \wedge F \Leftrightarrow F \qquad X \vee F \Leftrightarrow X$$

unit/true

$$X \wedge T \Leftrightarrow X \qquad X \vee T \Leftrightarrow T$$

involution

$$\neg(\neg X) \Leftrightarrow X$$

complement

$$X \wedge \neg X \Leftrightarrow F \qquad X \vee \neg X \Leftrightarrow T$$

propositional logic

- language (syntax)
- meaning (semantics) $\models$
- calculus ('automated' proofs) $\vdash$

see also: predicate logic
tools: *theorem provers*
correctness wrt. behaviour
specification vs. implementation

proof system

formal reasoning:

automated proofs

natural deduction (Gentzen)

inference rules

∧-introduction

if we have

$$\frac{X, \quad Y}{X \wedge Y}$$

we conclude

1.
2. Y
3.
4.
5. X
6.
7. X ∧ Y   5,2 ∧-introduction

Gerhard Gentzen
(November 24, 1909, Greifswald, Germany – August 4, 1945, Prague, Czechoslovakia)

inference rules (1)

∧-introduction

$$\frac{X,\ Y}{X \wedge Y}$$

∧-elimination

$$\frac{X \wedge Y}{X} \qquad \frac{X \wedge Y}{Y}$$

∨-introduction

$$\frac{X}{X \vee Y} \qquad \frac{X}{Y \vee X}$$

¬-elimination

$$\frac{\neg\neg X}{X}$$

→-elimination

$$\frac{X,\ X \rightarrow Y}{Y}$$

¬not ∧and ∨or →implies

propositional calculus

proof-within-proof

use assumptions,
valid only within scope!
clearly indicated
scope is 'closed' using proper rule

¬-introduction

$$\frac{X \vdash Y, \quad X \vdash \neg Y}{\neg X}$$

" if X holds, we get a contradiction, thus ¬X must hold "

1.		
2.	X	assumption
3.		
4.		
5.	Y	
6.		
7.	¬Y	
8.	¬X	2-7 ¬-introduction

$X \vdash \ldots$

proof-within-proof

…-within-proof

nested

$\neg$-introduction

$$\frac{X \vdash Y,\ \ X \vdash \neg Y}{\neg X}$$

$\vdash p \vee \neg p$ theorem

$\neg(p \vee \neg p) \vdash \ldots$

assumptions: scope

1	$\neg(p \vee \neg p)$	assumption
2	p	assumption
3	$p \vee \neg p$	$\vee$-introduction
4	$\neg(p \vee \neg p)$	1 copied
5	$\neg p$	2–4 $\neg$-introduction
6	$p \vee \neg p$	$\vee$-introduction
7	$\neg\ \neg\ (p \vee \neg p)$	1–6 $\neg$-introduction
8	$p \vee \neg p$	$\neg$-elimination

$\neg(p \vee \neg p),\ p \vdash \ldots$

inference rules (2: using assumptions)

∨-elimination

$$\frac{X \vdash Z,\quad Y \vdash Z,\quad X \lor Y}{Z}$$

¬-introduction

$$\frac{X \vdash Y,\quad X \vdash \neg Y}{\neg X}$$

→-introduction

$$\frac{X \vdash Y}{X \to Y}$$

¬not ∧and ∨or →implies

activities for the weekend

$p \stackrel{\text{def}}{=}$ sun shines
$q \stackrel{\text{def}}{=}$ we go for a walk
$r \stackrel{\text{def}}{=}$ we go to a museum

$$((p \land q) \lor (\neg p \land r)) \to (q \lor r)$$

whatever the weather … we have an activity

¬not ∧and ∨or →implies

whatever the weather … we have an activity

and we can prove it …

1	(p∧q)∨(¬p∧r)	assumption
2	p∧q	assumption
3	q	2 ∧-elimination
4	q∨r	3 ∨-introduction
5	¬p∧r	assumption
6	r	5 ∧-elimination
7	q∨r	6 ∨-introduction
8	q∨r	2-7 ∨-elimination
9	((p∧q)∨(¬p∧r))→(q∨r)	1-9 →-introduction

∧-elimination

$$\frac{X \wedge Y}{X} \qquad \frac{X \wedge Y}{Y}$$

∨-introduction

$$\frac{X}{X \vee Y} \qquad \frac{X}{Y \vee X}$$

∨-elimination

$$\frac{X \vdash Z,\ Y \vdash Z,\ X \vee Y}{Z}$$

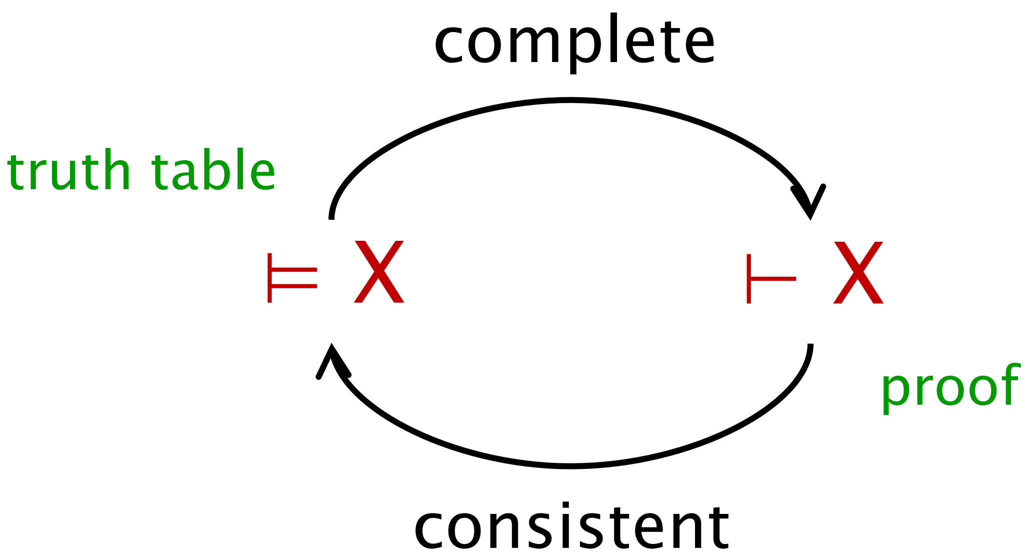
→-introduction

$$\frac{X \vdash Y}{X \to Y}$$

¬not ∧and ∨or →implies

central result

for propositional logic (& predicate logic)



Kurt Gödel

(April 28, 1906 Brno, Austria-Hungary – January 14, 1978 Princeton, NJ, United States)

1930, "Die Vollständigkeit der Axiome des logischen Funktionenkalküls." *Monatshefte für Mathematik und Physik* **37**: 349–60


BEGRIFFSSCHRIFT,

EINE DER ARITHMETISCHEN NACHGEBILDETE

FORMELSPRACHE

DES REINEN DENKENS.

VON

DR. GOTTLOB FREGE,

PRIVATDOCENTEN DER MATHEMATIK AN DER UNIVERSITÄT JENA.

HALLE A/S.

VERLAG VON LOUIS NEBERT.

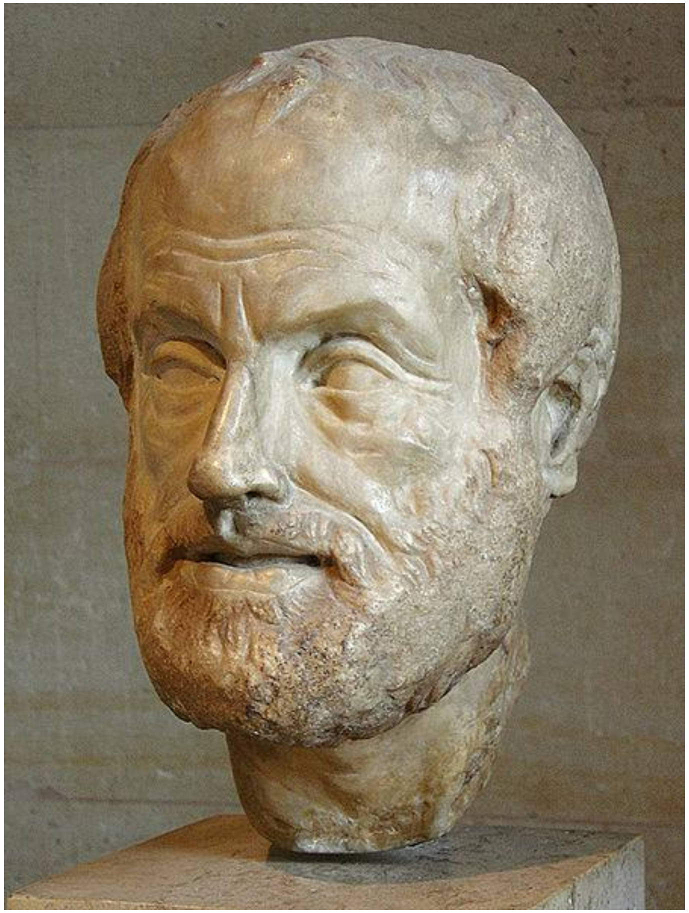
1879.

predicate logic

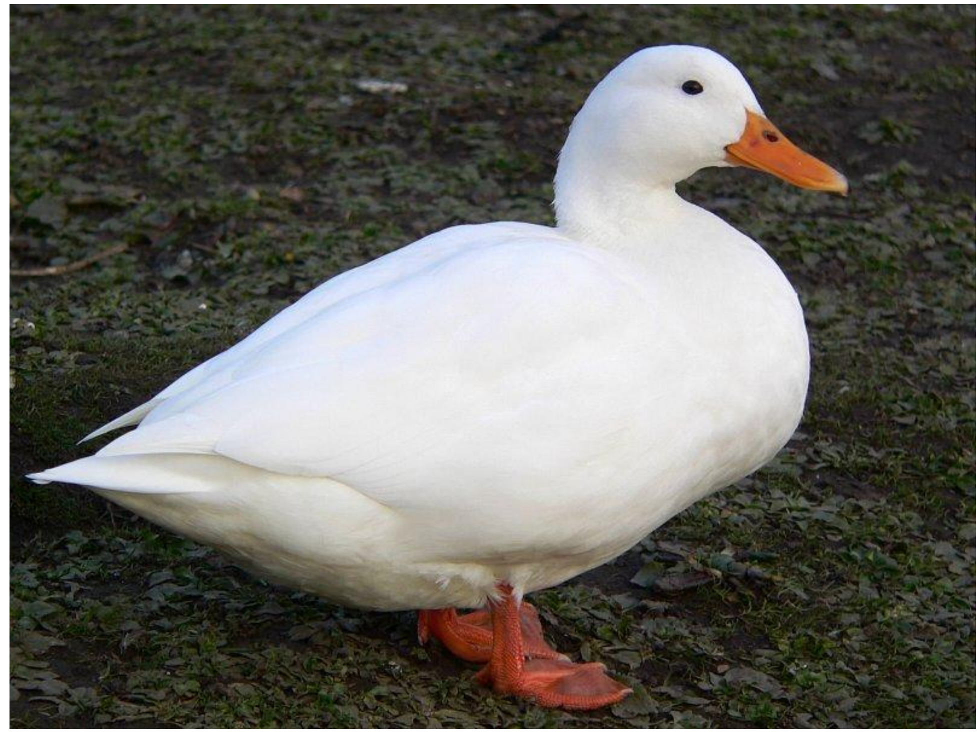
- objects and properties
- for all, exists $\forall \exists$
- calculus (automated reasoning)

no easy truth tables

syllogism



Aristotle (Ἀριστοτέλης, Aristotélēs)
(384 BC Stageira, Chalcidice – 322 BC Euboea)



Esmeralda

$p \stackrel{\text{def}}{\equiv}$ Esmeralda is a duck
$q \stackrel{\text{def}}{\equiv}$ all ducks like water
$r \stackrel{\text{def}}{\equiv}$ Esmeralda likes water

$p, \; q \models r$

duck(Esmeralda)
(forall x)(duck(x) → likes(x,water))
likes(Esmeralda,water)

D(e)
$(\forall x)(D(x) \rightarrow L(x,w))$
L(e,w)

predicate logic: syntax

two “sorts”

ducks	*integers*
D(x)	x=0
L(x,y)	x=y
mother(x)	−x, x+y
Esmeralda	0, 1

domain

Esmeralda, x, mother(x)

x, x+y, 1

truth values

$(\forall x)(\ D(x) \rightarrow L(x, mother(Esmeralda))\ )$

$(\forall x)(\exists y)\ (\ (x+y=0) \wedge (x \leq 1)\ )$

$(\forall x)(\ P(f(x)) \rightarrow \neg(\forall y)[\ P(x) \rightarrow Q(f(y),x,z)\ ]\ )$

predicate logic: syntax

quantifiers + variables

for all… $\forall$ universal

exists… $\exists$ existential

predicates = relations (statements about domain)

functions (on domain)

names = constants

ducks	integers
D(x)	$x \leq y$
L(x,y)	$-x$
mother(x)	$x+y$
Esmeralda	0, 1

elt domain V := c | x | f(V,…,V)

truth value P := T | F | A(V,…,V) | ($\forall$x) P | ($\exists$x) P |

($\neg$P) | (P$\vee$P) | (P$\wedge$P) | (P$\rightarrow$P) | (P$\leftrightarrow$P)

quiz

$\mathbb{Z} = \{ \ldots, -2, -1, 0, 1, 2, \ldots \}$ integers

are these statements true (in $\mathbb{Z}$)?

$+$ addition $\cdot$ multiplication $=$ equality

1. $(\forall x)(\exists y) \quad x+y=0$
2. $(\exists x)(\forall y) \quad x+y=0$
3. $(\forall x)(\exists y) \quad x \cdot y=0$
4. $(\exists x)(\forall y) \quad x \cdot y=0$

additional inference rules

∀-elimination

$$\frac{(\forall x)P(x)}{P(t)}$$

replace x
by any term t

∀-introduction

$$\frac{P(a)}{(\forall x)P(x)}$$

a ‘fresh’

∃-elimination

$$\frac{(\exists x)P(x),\ (\forall x)(\ P(x)\rightarrow Q\ )}{Q}$$

∃-introduction

$$\frac{P(t)}{(\exists x)P(x)}$$

any term t

theory: model of the 'world'
to reason within a certain reality we use axioms

Ax.1 father(John) = Harry
Ax.2 father(Bill) = John
Ax.3 $(\forall x)$ father(father(x)) = grandfather(x)

Bill = 1, John = 2, Harry = 4
father(x) = 2x
grandfather(x) = 4x

1.	father(John) = Harry	axiom
2.	father(Bill) = John	axiom
3.	father(father(Bill)) = Harry	1,2 substitution
4.	$(\forall x)$ father(father(x)) = grandfather(x)	axiom
5.	father(father(Bill)) = grandfather(Bill)	4 $\forall$-elimination
6.	Harry = grandfather(Bill)	3,5 substitution

sets, relations, functions

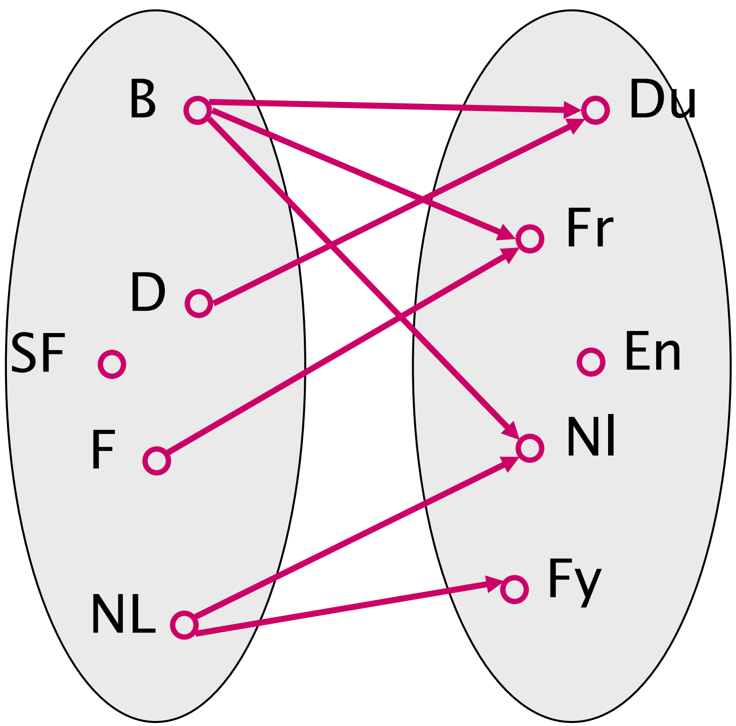
specification data types

- sets
- relations & functions

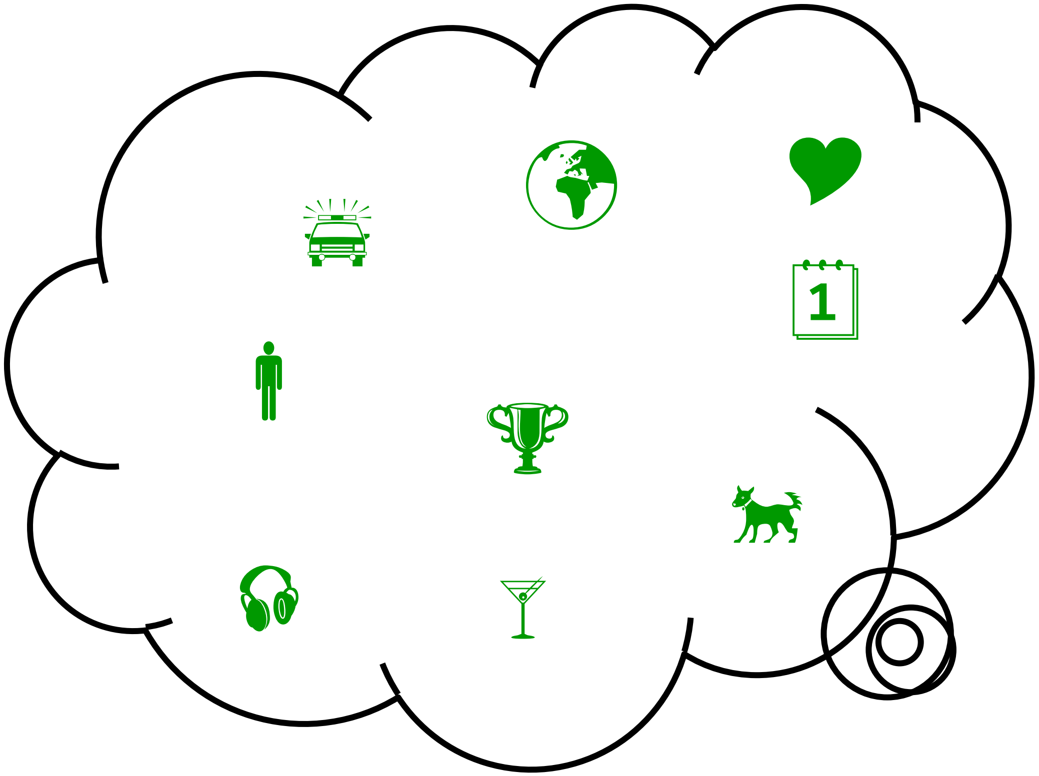
languages spoken

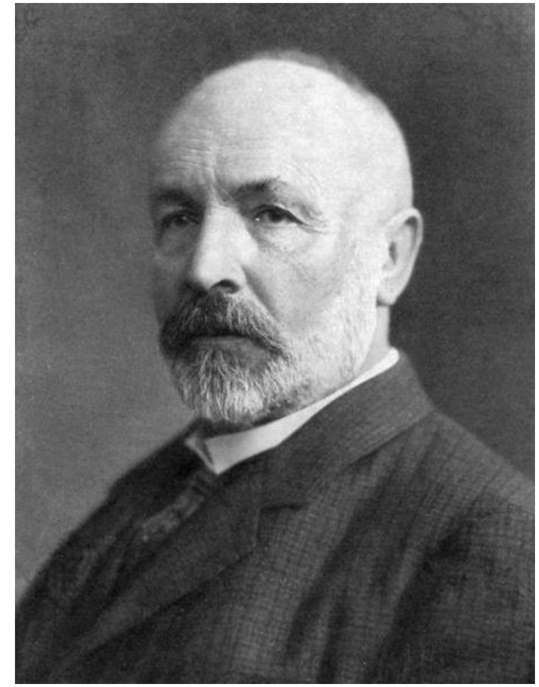
countries B, D, F, NL, SF
languages Du, En, Fr, Fy, Nl

B↦Du, B↦Fr, B↦Nl, D ↦Du, F↦Fr, NL↦Nl, NL↦Fy



set





Georg Cantor

(March 3, 1845 Saint Petersburg, Russian Empire – January 6, Halle, German Empire)

definition (?)

"a set is any well-defined collection of objects"

{ 0,2,4,6 } — extension

{ 0,2,4, … }

{ x | x is even } — comprehension

{ x | P(x) } property P

no order, no duplicates

{ 1,2,3 } = { 3,1,2 } = { 1,2,2,3,3,3 }

element of $x \in A$ $x \notin A$ 'in'

equality $A = B$ $(\forall x)(x \in A \leftrightarrow x \in B)$

subset

universe U
empty set $\varnothing$

subset $A \subseteq B$ inclusion, contained in

$\{ 3,5,7,11,13 \} \subseteq \{ 1,3,5,7,9,11,13,15 \}$
$\{ 2,3,5,7 \} \not\subseteq \{ 1,3,5,7,9,11,13,15 \}$

$A = B$ iff $A \subseteq B$ and $B \subseteq A$

sets and predicates

$\{ x \in U \mid P(x) \}$

$\mathbb{N} = \{ x \in \mathbb{N} \mid \text{true} \}$

$\varnothing = \{ x \in \mathbb{N} \mid \text{false} \}$

$\text{Even} \stackrel{\text{def}}{=} \{ x \in \mathbb{N} \mid (\exists z \in \mathbb{N})(2 \cdot z = x) \}$

$= \{ 2 \cdot x \mid x \in \mathbb{N} \}$ comprehension by form

$\text{Primes} \stackrel{\text{def}}{=}$

$\{ x \in \mathbb{N} \mid (\forall y \in \mathbb{N})[(\exists z \in \mathbb{N})(z \cdot y = x) \rightarrow (y = 1 \lor y = x)]$

y divides x exactly

reasoning with sets

$P = \{ x \mid P(x) \}, \quad Q = \{ x \mid Q(x) \}$

$(\forall x)(P(x) \rightarrow Q(x)) \quad \text{iff} \quad P \subseteq Q$

set-axioms
comprehension

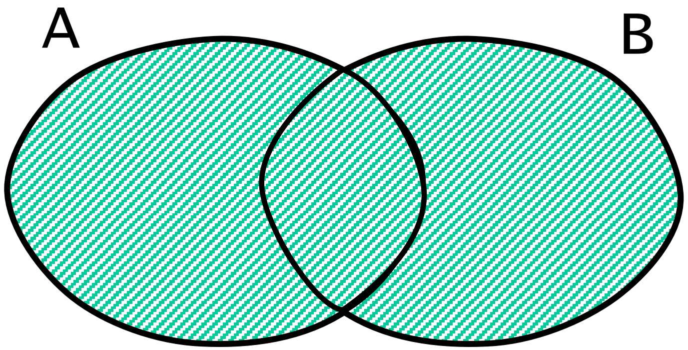
$P(a) \text{ iff } a \in P$

comprehension

$$\frac{a \in \{x \mid P(x)\}}{P(a)}$$

1.	$(\forall x)(P(x) \rightarrow Q(x))$	assumption
2.	$P(a) \rightarrow Q(a)$	1 $\forall$-elimination
3.	$a \in \{x \mid P(x)\}$	assumption
4.	$P(a)$	3 comprehension
5.	$Q(a)$	2,4 $\rightarrow$-elimination
6.	$a \in \{x \mid Q(x)\}$	5 comprehension
7.	$a \in \{x \mid P(x)\} \rightarrow a \in \{x \mid Q(x)\}$	3–6 $\rightarrow$-introduction
8.	$(\forall x)(x \in \{x \mid P(x)\} \rightarrow x \in \{x \mid Q(x)\})$	8 $\forall$-introduction
9.	$(\forall x)(P(x) \rightarrow Q(x)) \rightarrow$ "$P \subseteq Q$"	1–8 $\rightarrow$-introduction

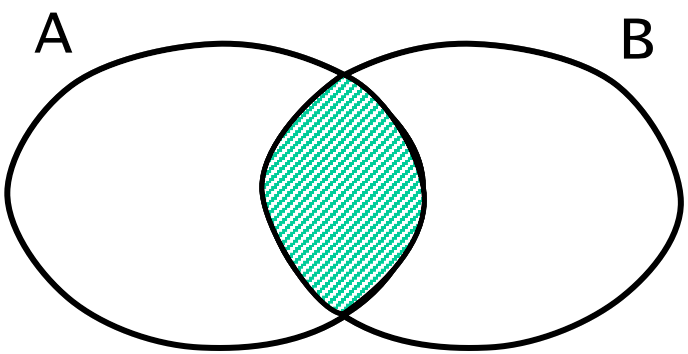
boolean set operations



universe U 'type'

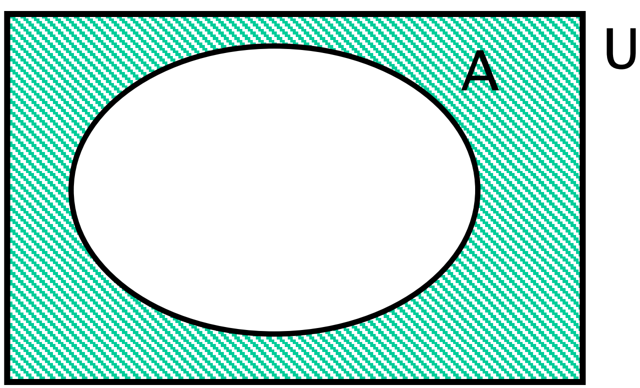
union

$A \cup B = \{ x \in U \mid x \in A \lor x \in B \}$



intersection

$A \cap B = \{ x \in U \mid x \in A \land x \in B \}$



complement

$A^c = \{ x \in U \mid \neg(x \in A) \}$

powerset

$\mathbb{P}A = \{ x \mid x \subseteq A \}$

powerset of A   all subsets $\wp(A)$ 2^A

'collection'   *higher level set*

$\mathbb{P}\{a,b,c\} =$

$\{ \emptyset, \{a\}, \{b\}, \{a,b\}, \{c\}, \{a,c\}, \{b,c\}, \{a,b,c\} \}$

$\mathbb{P}\emptyset = \{ \emptyset \}$

$X \subseteq A$ iff $X \in \mathbb{P}A$ $\{a,b\} \subseteq A$ iff $\{a,b\} \in \mathbb{P}A$

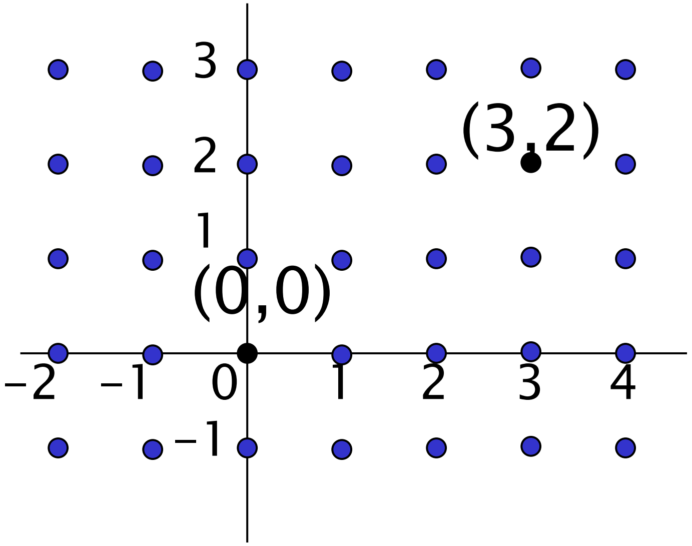
$\#(\mathbb{P}A) = 2^{\#(A)}$ finite set A

sets, relations, functions

specification data types
- sets
- relations & functions

coordinates

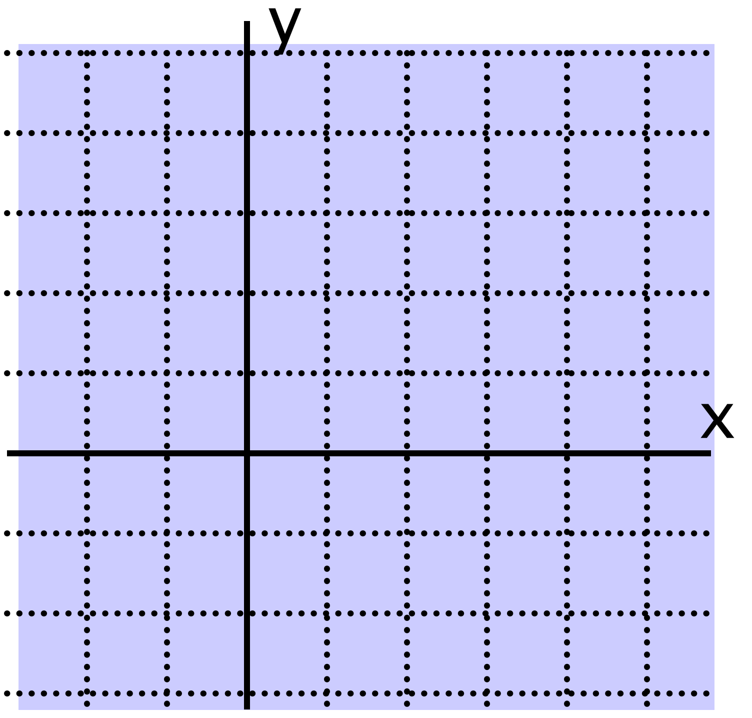
grid



$\mathbb{Z}\times\mathbb{Z} = \mathbb{Z}^2 =$

$\{ (x,y) \mid x,y \in \mathbb{Z} \}$

plane



$\mathbb{R}\times\mathbb{R} = \mathbb{R}^2 =$

$\{ (x,y) \mid x,y \in \mathbb{R} \}$

$\mathbb{R}^3$ space …

stud	*cour*	*mark*
8303	M250	7
8303	T350	8
4722	B140	7
4722	S570	10
4722	T480	9
0347	M250	6
4948	B140	9
4948	M250	9
1576	C250	7
9594	T250	6
9352	U161	9
2592	A470	8
2592	M350	9
2592	V400	6

stud S = { 0000, …, 9999 }
cour C = { A000, …, Z999 }
mark M = { 0, 1, …, 9, 10 }

V = { (8303,M250,7),
(8303,T350,8),
…
(2592,V400,6) }

$V \subseteq S \times C \times M$

$V \in \mathbb{P}(S \times C \times M)$ ‘type’ product $A \times B$ (ordered) pairs

$\{ (a,b) \mid a \in A \text{ and } b \in B \}$

$A = \{ \text{me, you} \}$ $B = \{ 1,2,3,4 \}$

$A \times B = \{ (\text{me},1), (\text{me},2), (\text{me},3), (\text{me},4)$
$(\text{you},1), (\text{you},2), (\text{you},3), (\text{you},4) \}$

$A^2 = A \times A$

cartesian product $A_1 \times A_2 \times \ldots \times A_n$

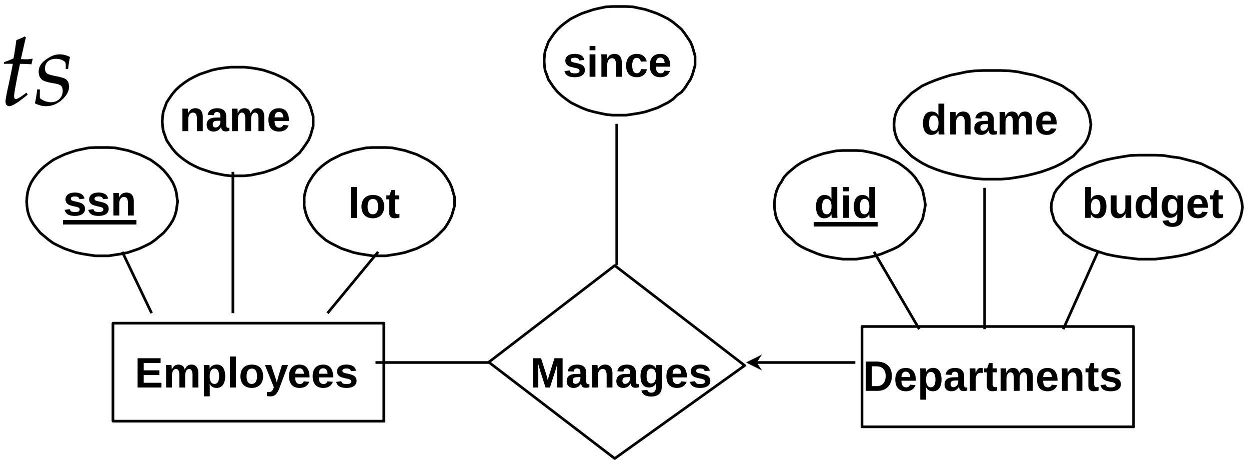
$R \subseteq A \times B$ (binary) relation subset full product
$R \in \mathbb{P}(A \times B)$

$A = \{ \text{me}, \text{you} \}$ $B = \{ 1,2,3,4 \}$
$\{ (\text{me},1), (\text{me},3), (\text{you},2) \} \in \mathbb{P}(A \times B)$

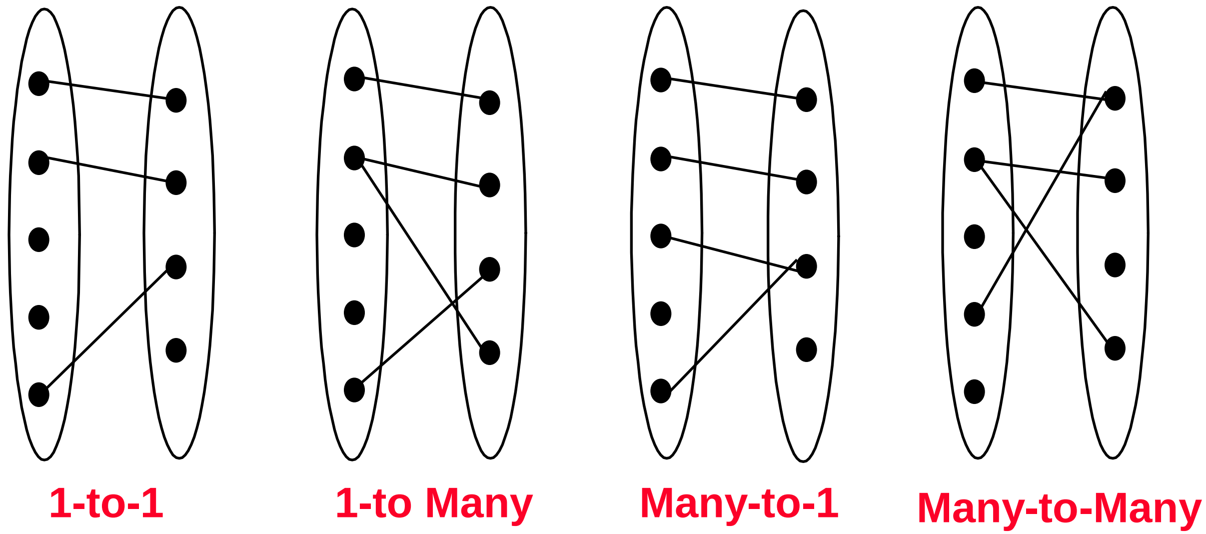
$\text{Succ} \stackrel{\text{def}}{=} \{ (n,n+1) \mid n \in \mathbb{N} \} \in \mathbb{P}(\mathbb{Z} \times \mathbb{Z})$
$\text{Div} \stackrel{\text{def}}{=} \{ (d,n) \in \mathbb{N}^2 \mid (\exists q)(d \cdot q = n) \} \in \mathbb{P}(\mathbb{Z} \times \mathbb{Z})$

object	'type'	example
number	$\mathbb{Z}$	3
pair	$\mathbb{Z} \times \mathbb{Z}$	(4,2)
set	$\mathbb{P}\mathbb{Z}$	{ 1,2,3,4 }
relation	$\mathbb{P}(\mathbb{Z} \times \mathbb{Z})$	{ (1,2), (3,1), (3,7), … }
subset $\subseteq$	$\mathbb{P}(\mathbb{P}\mathbb{Z} \times \mathbb{P}\mathbb{Z})$	{ …, ({1,2},{1,2,3}), … }

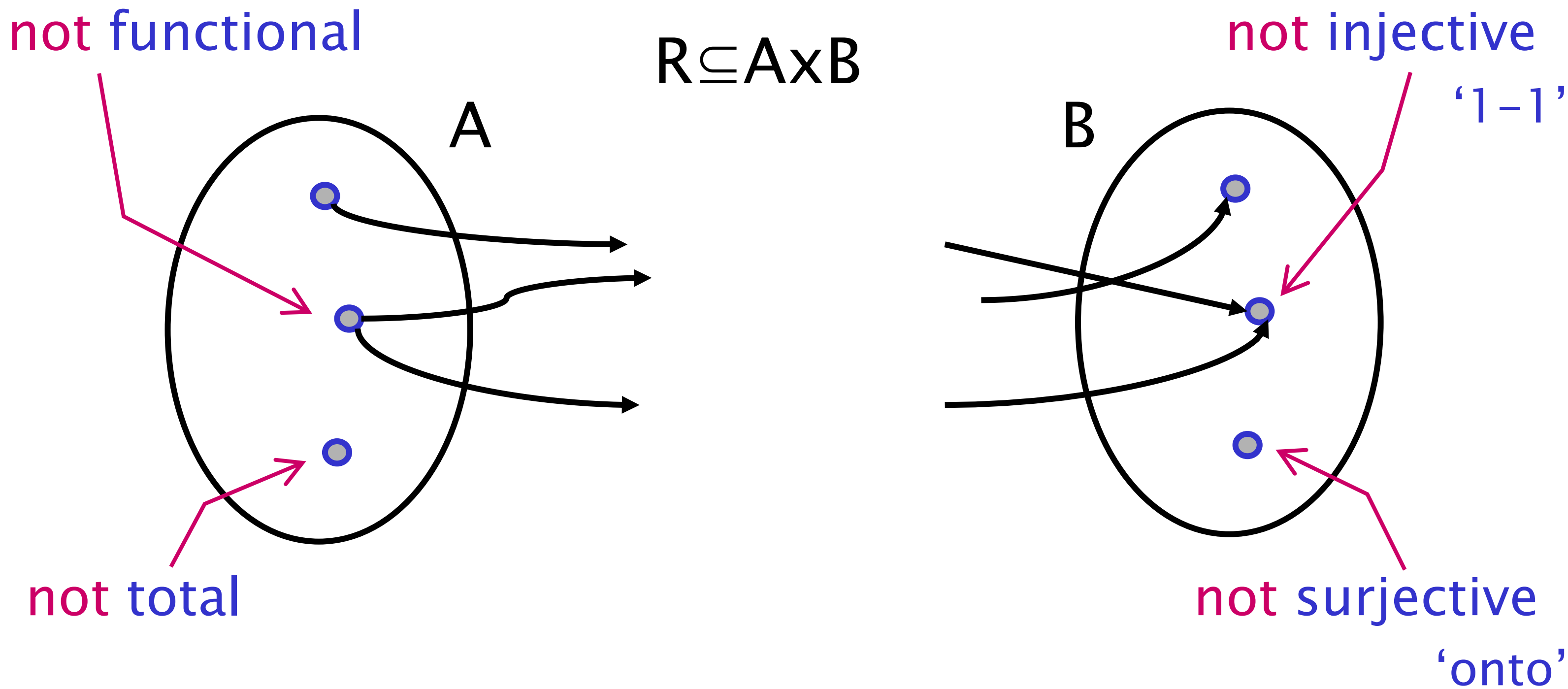
Key Constraints



- Consider Works_In: An employee can work in many departments; a dept can have many employees.
- In contrast, each dept has at most one manager, according to the *key constraint* on Manages.



relation types



domain $dom(R) = \{ x \in A \mid xRy \text{ for some } y \in B \}$

range $ran(R) = \{ y \in B \mid xRy \text{ for some } x \in A \}$

functional if xRy_1 and xRy_2 then $y_1 = y_2$

total $dom(R) = A$

injective if x_1Ry and x_2Ry then $x_1 = x_2$

surjective $ran(R) = B$

4.3 Functions

Name

$\pfun$	–	Partial functions
$\fun$	–	Total functions
$\pinj$	–	Partial injections
$\inj$	–	Total injections
$\psurj$	–	Partial surjections
$\surj$	–	Total surjections
$\bij$	–	Bijections

Definition

$$X \pfun Y == \{ f : X \leftrightarrow Y \mid (\forall x : X; y_1, y_2 : Y \bullet (x \mapsto y_1) \in f \land (x \mapsto y_2) \in f \Rightarrow y_1 = y_2) \}$$

$$X \fun Y == \{ f : X \pfun Y \mid \mathrm{dom}\, f = X \}$$

$$X \pinj Y == \{ f : X \pfun Y \mid (\forall x_1, x_2 : \mathrm{dom}\, f \bullet f(x_1) = f(x_2) \Rightarrow x_1 = x_2) \}$$

$$X \inj Y == (X \pinj Y) \cap (X \fun Y)$$

$$X \psurj Y == \{ f : X \pfun Y \mid \mathrm{ran}\, f = Y \}$$

$$X \surj Y == (X \psurj Y) \cap (X \fun Y)$$

$$X \bij Y == (X \surj Y) \cap (X \inj Y)$$

The Z Notation: A Reference Manual – J. M. Spivey

example: Z specification
Birthday Book

http://michellelast.typepad.co.uk/my_weblog/2008/08/new-stampin-up.html

The Z Notation: A Reference Manual – J. M. Spivey

specification

BirthdayBook

known : ℙ NAME
birthday : NAME ⇸ DATE

known = dom birthday

known = { John, Mike, Henry }
birthday = { John ↦ 25–Mar,
Mike ↦ 01–Aug,
Henry ↦ 01–Aug }

AddBirthday

ΔBirthdayBook
name? : NAME
date? : DATE

name? ∉ known
birthday′ = birthday ∪ { name? ↦ date? }

known′ = known ∪ { name? }

FindBirthday

ΞBirthdayBook
name? : NAME
date! : DATE

name? ∈ known
date! = birthday(name?)

known′ = known
birthday′ = birthday

InitBirthdayBook

BirthdayBook

$known = \varnothing$

Remind

$\Xi BirthdayBook$

$today? : DATE$

$cards! : \mathbb{P}\ NAME$

$cards! = \{ n : known \mid birthday(n) = today? \}$

$$m \in \{ n : known \mid birthday(n) = today? \} \Leftrightarrow m \in known \land birthday(m) = today?$$

strengthening the specification

$name? \notin known$

what must we do otherwise?

AddBirthday

$\Delta BirthdayBook$
$name? : NAME$
$date? : DATE$

$name? \notin known$
$birthday' = birthday \cup \{ name? \mapsto date? \}$

FindBirthday

$\Xi BirthdayBook$
$name? : NAME$
$date! : DATE$

$name? \in known$
$date! = birthday(name?)$

strengthening the specification

$name? \notin known$

$REPORT ::= ok \mid already\_known \mid not\_known$

Success

$result! : REPORT$

$result! = ok$

AlreadyKnown

$\Xi BirthdayBook$

$name? : NAME$

$result! : REPORT$

$name? \in known$

$result! = already\_known$

NotKnown

$\Xi BirthdayBook$

$name? : NAME$

$result! : REPORT$

$name? \notin known$

$result! = not\_known$

revised

$RAddBirthday \triangleq (AddBirthday \wedge Success) \vee AlreadyKnown$

RAddBirthday

$\Delta BirthdayBook$
$name? : NAME$
$date? : DATE$
$result! : REPORT$

$(name? \notin known \wedge$
$\quad birthday' = birthday \cup \{ name? \mapsto date? \} \wedge$
$\quad result! = ok) \vee$
$(name? \in known \wedge$
$\quad birthday' = birthday \wedge$
$\quad result! = already\_known)$

$RFindBirthday \triangleq (FindBirthday \wedge Success) \vee NotKnown$

$RRemind \triangleq Remind \wedge Success$

final words

1980's vision

1980s: by the 21st century, a large proportion of software developed using formal methods.
clearly, this prediction has not come true.

- successful software engineering.

structured methods, configuration management and information hiding

- market changes.

critical issue is not quality but time-to-market

- limited scope of formal methods.

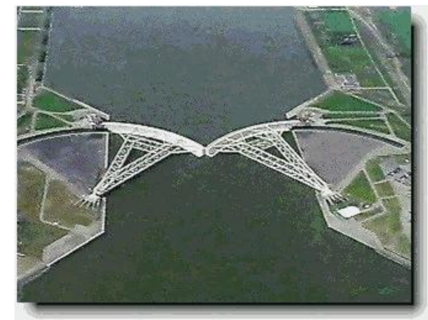
not well suited to specifying user interfaces and user interaction.

- limited scalability of formal methods.

mostly concerned with relatively small, critical kernel systems

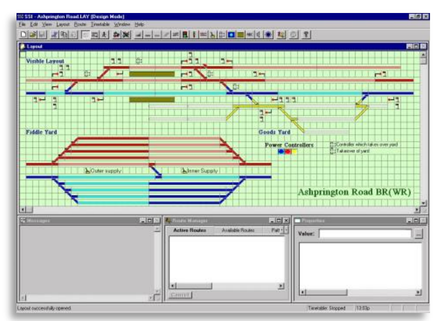
Formal Methods in Practice

Formal Methods are not ‘silver bullet’ / panacea, but *one* technique that can be used to improve quality of software.



typical use is to apply formal methods only to critical parts of system;

industrial use of formal methods:
- traffic control (air, rail, …)
- medical devices
- space software (satellites, robots, …)



{end}