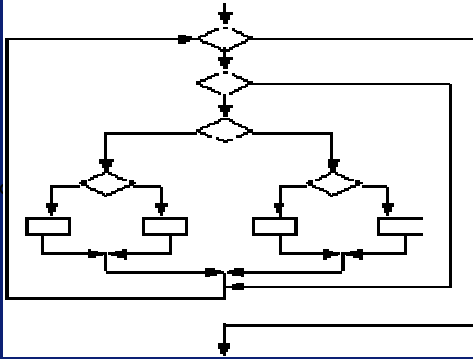


Spring 2009



Program correctness

Transition systems

Marcello Bonsangue



Leiden Institute of Advanced Computer Science
Research & Education

Formal Verification

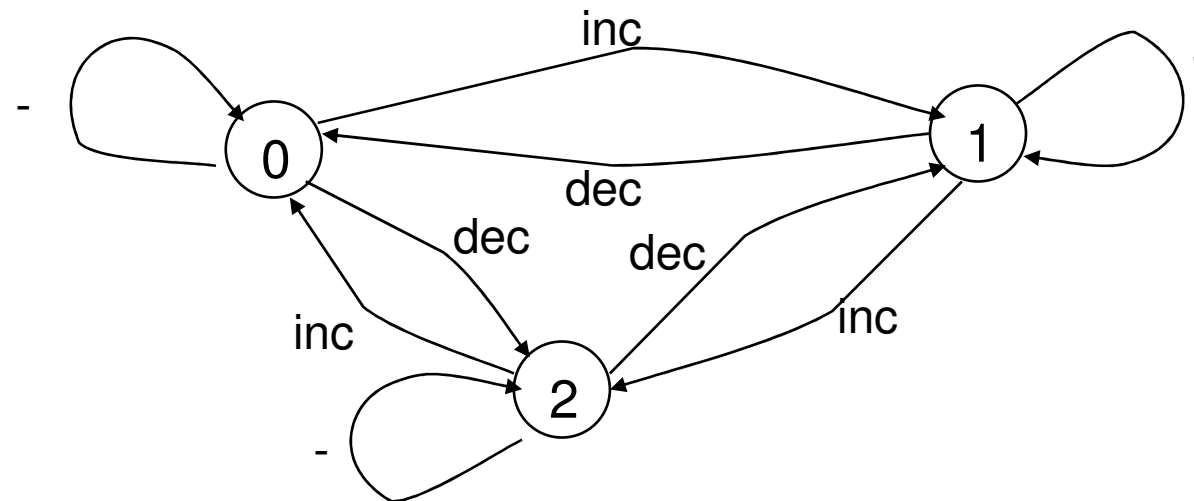
Verification techniques comprise

- a modelling framework M, Γ
to describe a system
- a specification language ϕ
to describe the properties to be verified
- a verification method $M \models \phi, \Gamma \vdash \phi$
to establish whether a model satisfies a property



Transition Systems

- A very general modelling framework
- Intuitively: a system evolves from one **state** to another under the action of a **transition**

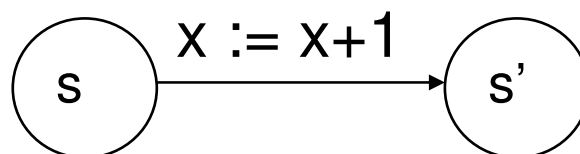


A modulo 3 counter



Example: an assignment

States: $s:\text{Var} \rightarrow \text{Val}$



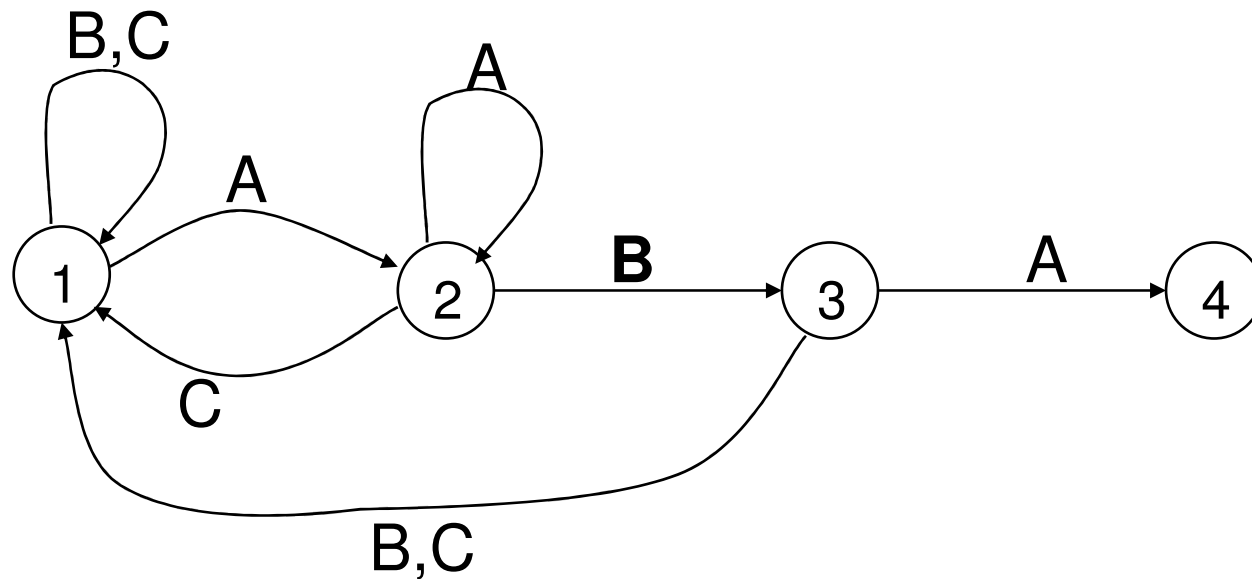
where $s' = s[s(x) + 1/x]$ and

$$f[v/x](y) = \begin{cases} f(y) & \text{if } x \neq y \\ v & \text{if } x = y \end{cases}$$



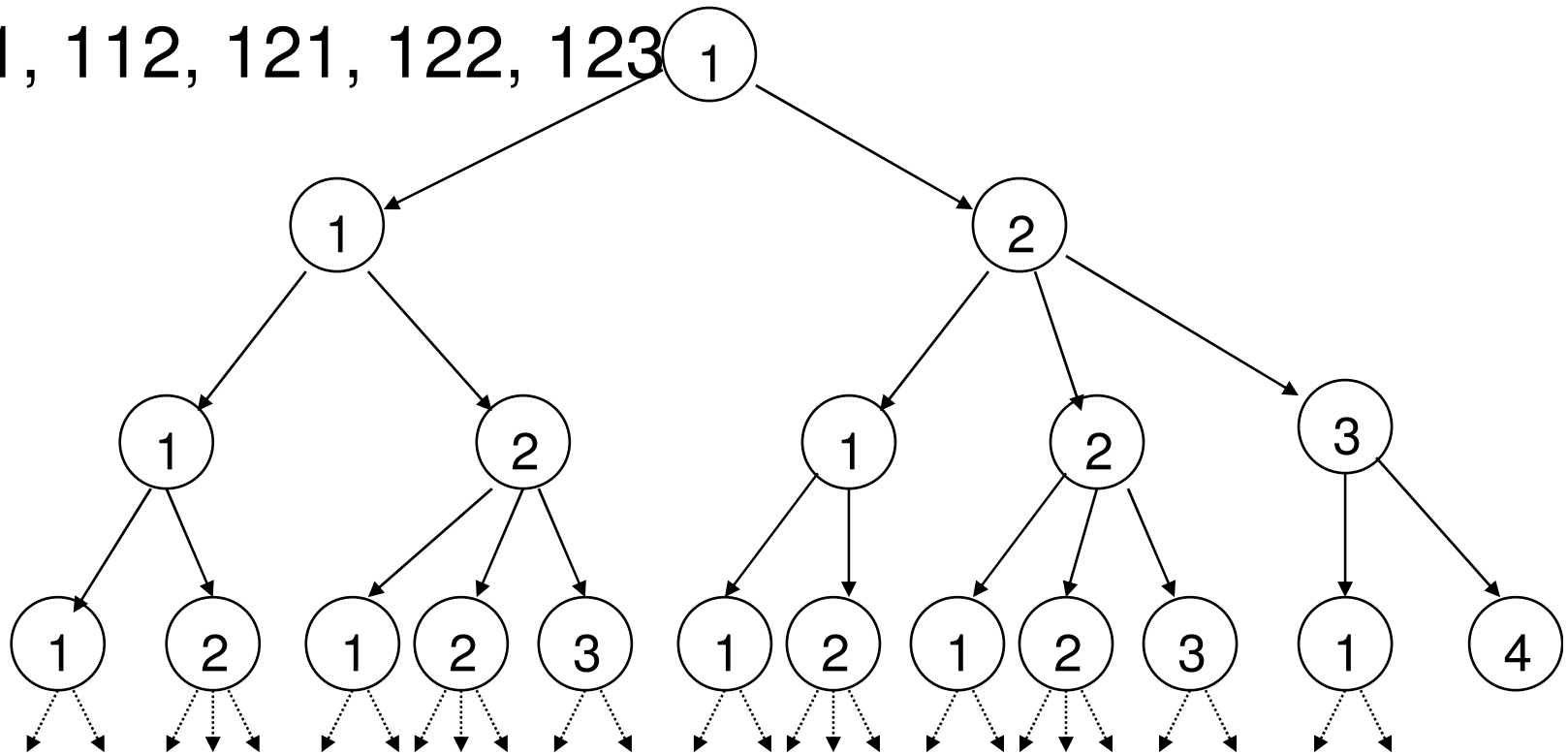
Example: a digicode

- 3 keys: A, B, C
- The door open when ABA is keyed



Digicode's executions

- 1
- 11, 12
- 111, 112, 121, 122, 123
- ...



A few definitions

- **Transition system:** $\langle S, L, \rightarrow \rangle$
 - S set of states
 - L set of transition labels
 - $\rightarrow \subseteq S \times L \times S$ transition relation
- **Path:** a sequence π of infinite transitions which follow each other

For example

$$3 \xrightarrow{B} 1 \xrightarrow{A} 2 \xrightarrow{A} 2 \dots$$

is a path of the digicode



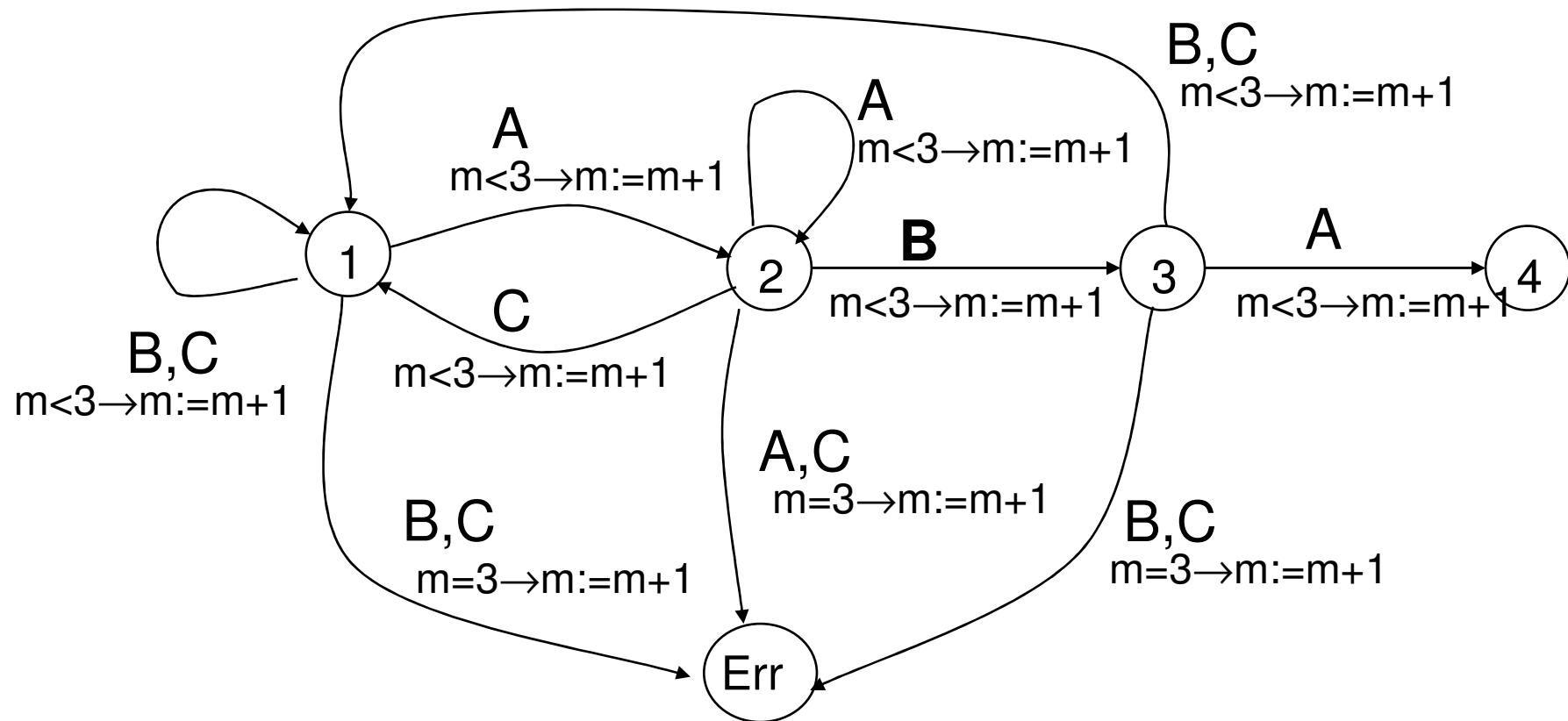
Adding data

- Real-life systems consist of control and data. We can model them by
 - control = states+transitions
 - data = **state variables**
- A transition system interact with state variables in two ways
 - **guards** a transition cannot occur if the condition does not holds
 - **assignment** a transition can modify the value of some state variables



Back to the digicode

- We do not tolerate more than 3 mistakes (recorded by the variable m)



Unfolding

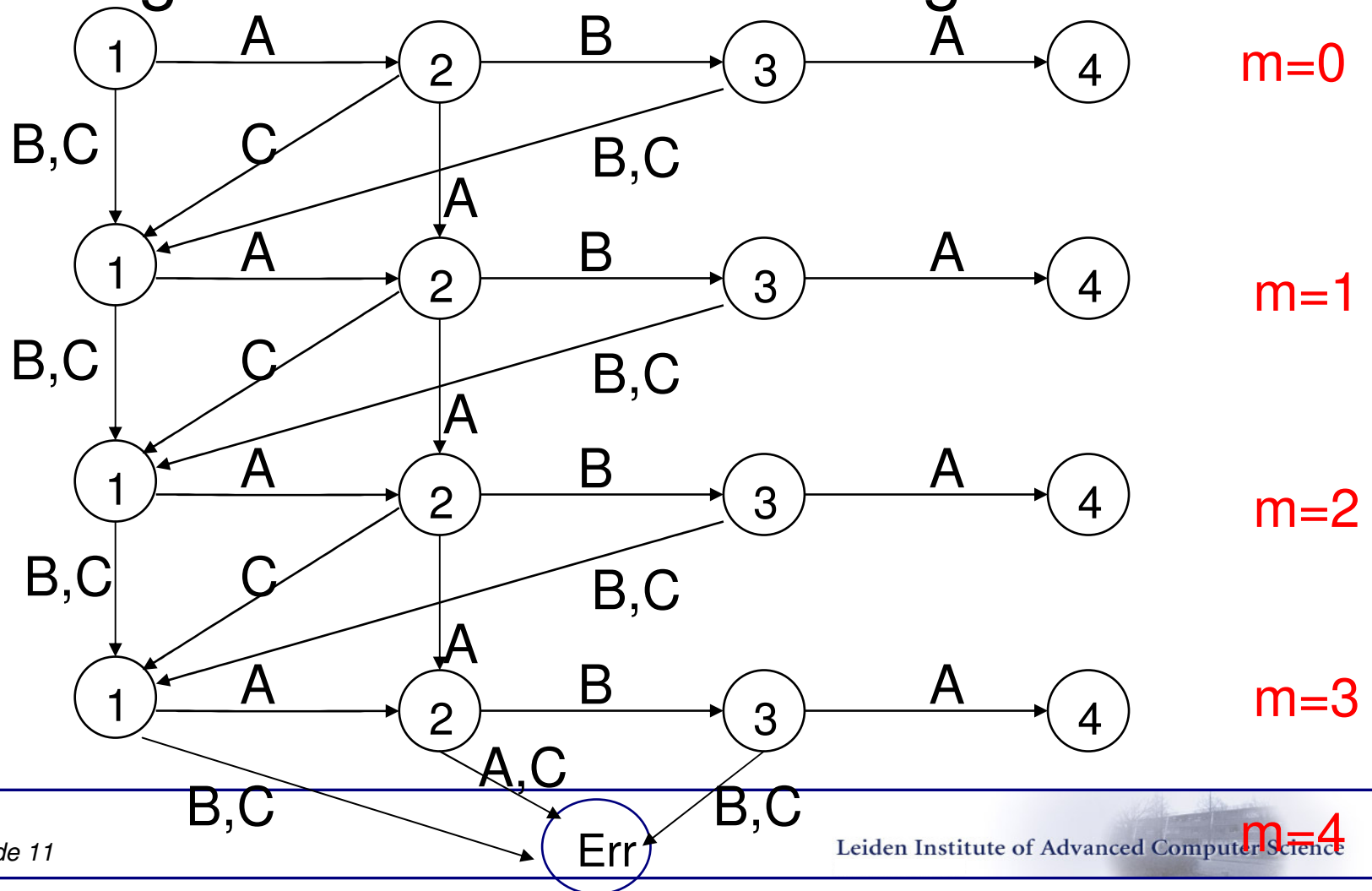
From a theoretical point of view, transition systems with state variables are not strictly necessary, as we can **unfold** them into ordinary transition systems.

- The new states correspond to the old ones + a component for each variable giving its value
- no more guards and assignment on the new transitions



Unfolding: example

- The digicode with error counting



Composing systems

- Systems often consists of cooperating subsystems. Next we describe how to obtain a global transition system from its subsystem by having them cooperate
- There are many ways to cooperate:
 - **product** (no interaction)
 - **synchronous product**
 - by message passing
 - by asynchronous channels
 - by shared variables



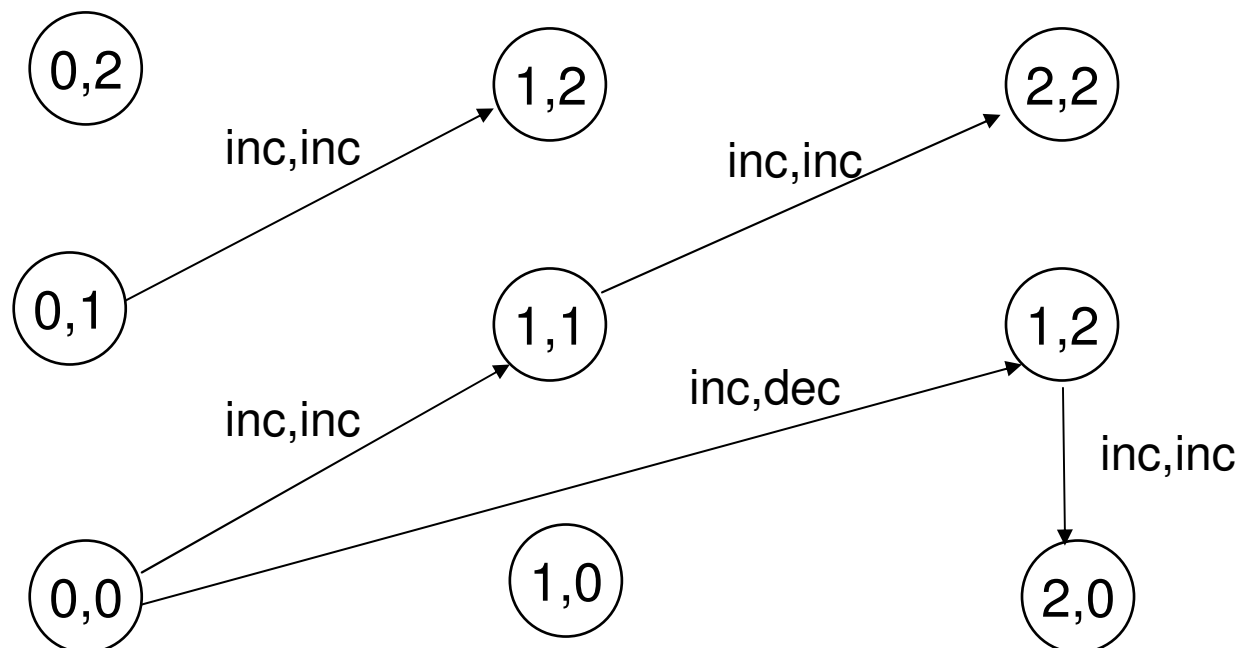
Product

- Subsystems do not interact with each other
- The resulting transition system $\langle S, L, \rightarrow \rangle$ is the **cartesian product** of the transition systems $\langle S_1, L_1, \rightarrow \rangle, \dots, \langle S_n, L_n, \rightarrow \rangle$ representing the subsystems
 - $S = S_1 \times \dots \times S_n$
 - $L = L_1 \times \dots \times L_n$
 - $\langle s_1, \dots, s_n \rangle \xrightarrow{\langle e_1, \dots, e_n \rangle} \langle t_1, \dots, t_n \rangle$ if for all i , $s_i \xrightarrow{e_i} t_i$



Example

- Few transitions of the product of two modulo 3 counters



Synchronized Product

- Subsystems interact by doing some step together (synchronization).
- To synchronize subsystems we restrict the transitions allowed in their cartesian product.
- A **synchronization set**

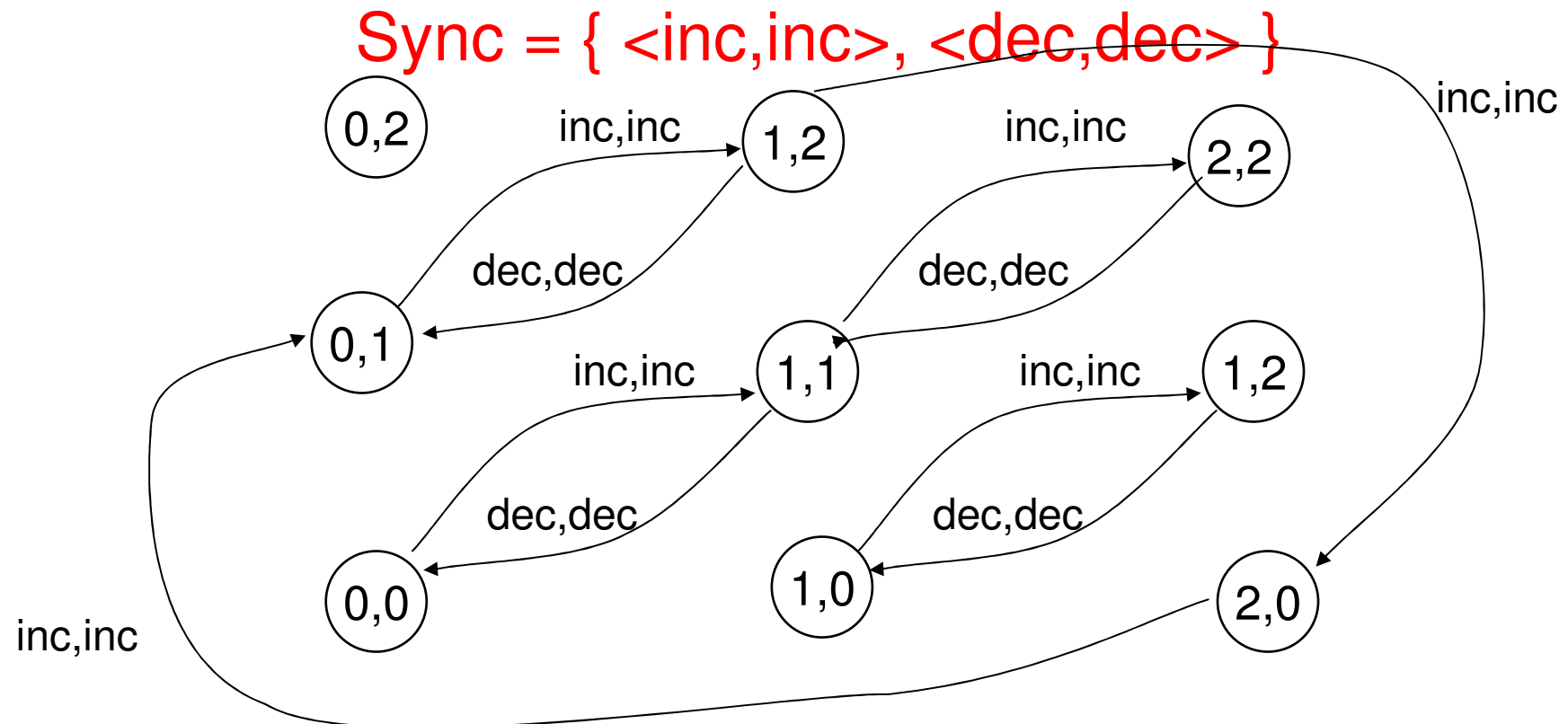
$$\text{Sync} \subseteq L_1 \times \dots \times L_n$$

define the labels of those transitions corresponding to a synchronization. Transitions with other labels are forbidden.



Example

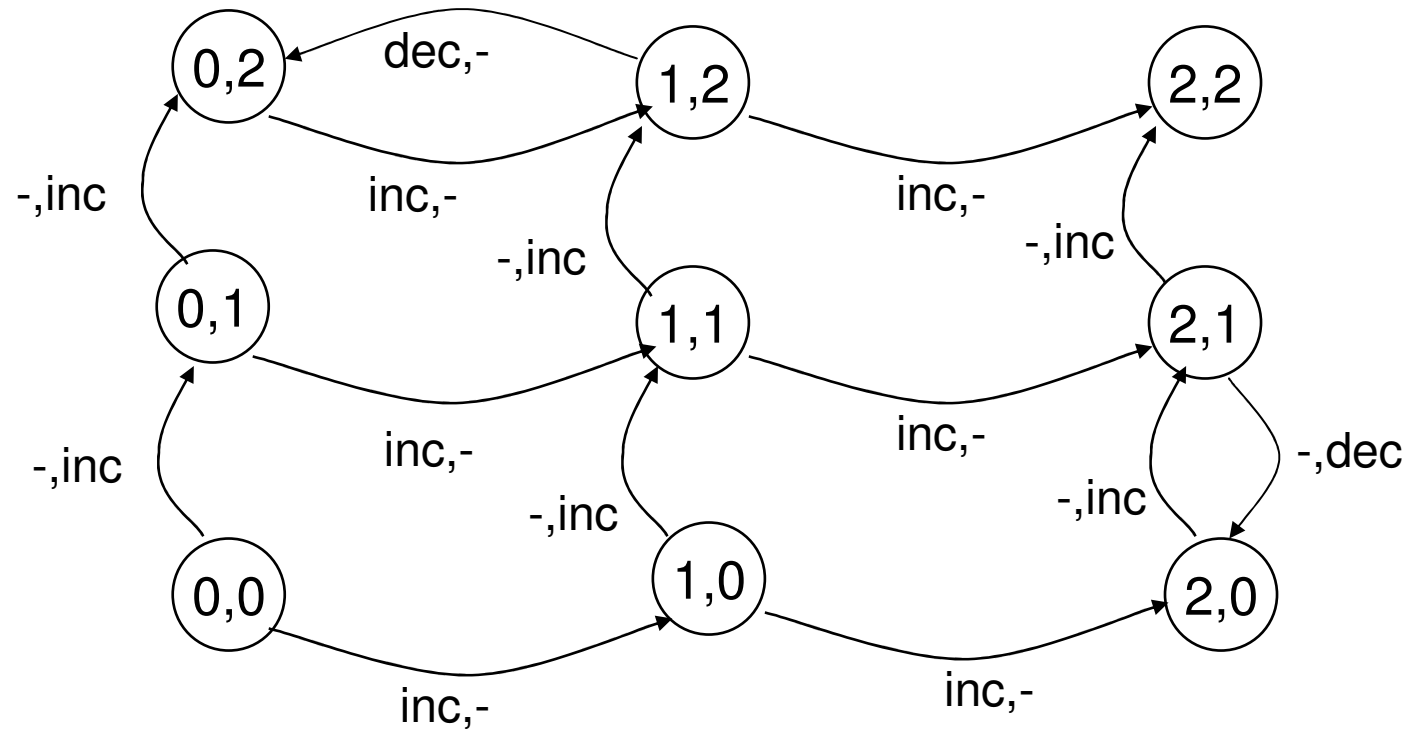
- Few transitions of two counters counting at the same time



Example

- Few transitions of two counters counting one at the time

Sync = { <inc,->, <dec,->, <-,inc>, <-,dec> }



Message Passing

- A special case of synchronized product
- Two special sets of labels
 - !m emission of message m
 - ?m reception of message m

- In message passing, only transitions in which a given emission is executed simultaneously with the corresponding reception will be permitted



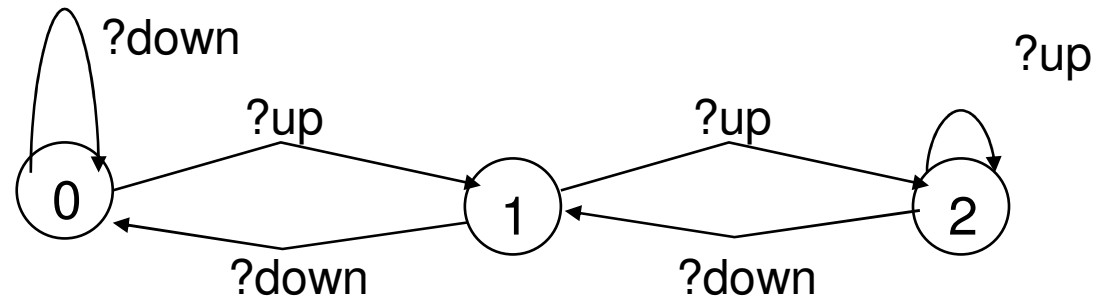
Example: An elevator

- An elevator in a three floors building consists of
 - a cabin which goes up and down
 - three doors which open and close
 - a controller which commands the three doors and the cabin
- Elevator requests from people at one of the three floors are not modeled, as they are the environment outside the system

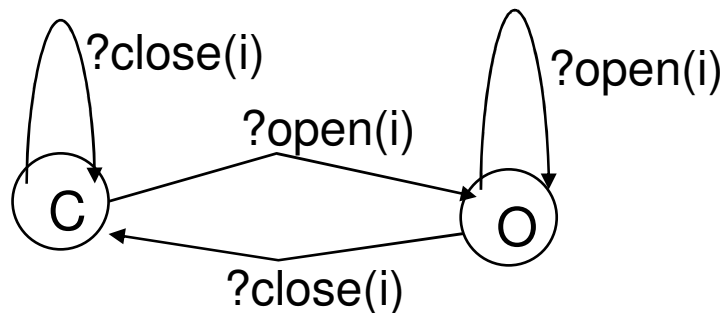


Example: An elevator

- The cabin

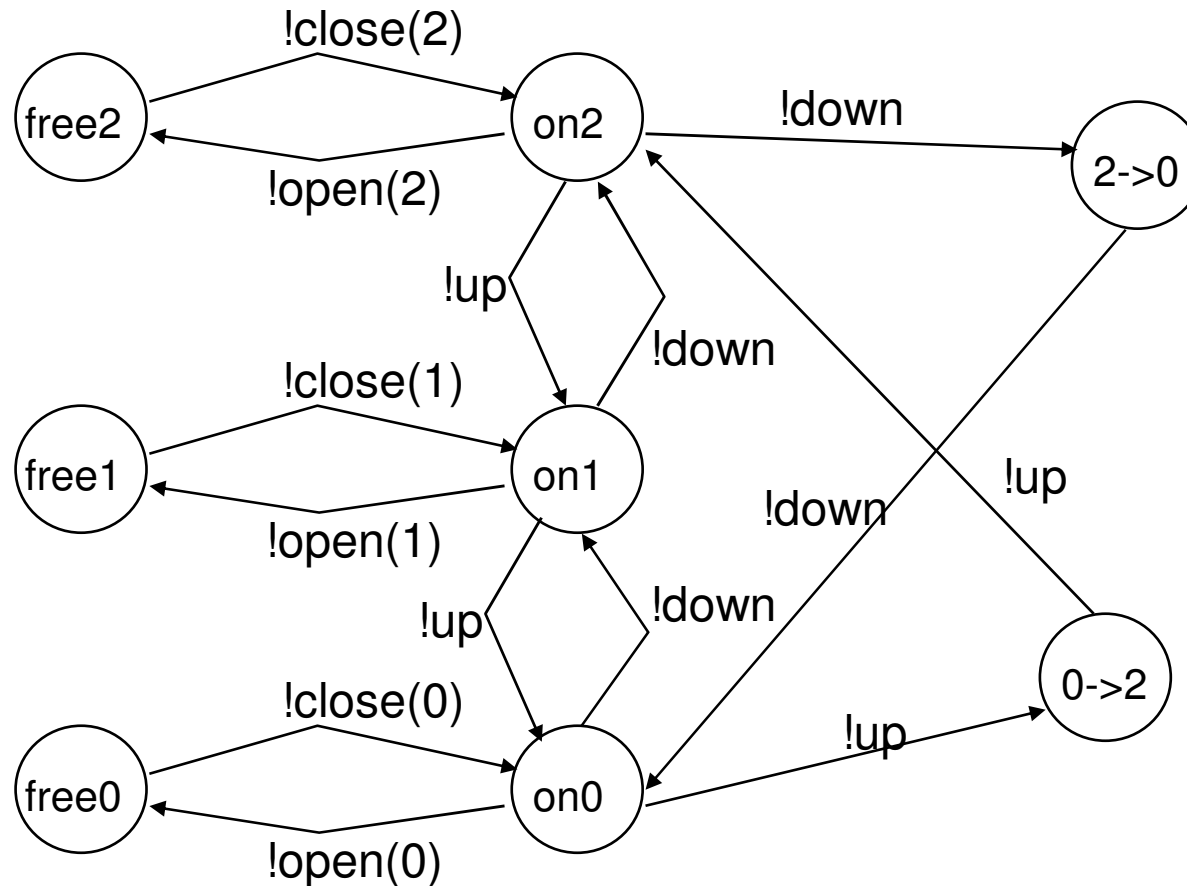


- The i-th door



Example: An elevator

■ The controller



Example: An elevator

- The synchronization

- Sync =

- {<?open(0),-,-,-,!open(0)>,<?close(0),-,-,-,!close(0)>,
<-,?open(1),-,-,-,!open(1)>,<-,?close(1),-,-,-,!close(1)>,
<-,-,?open(2),-,-,!open(2)>,<-,-,?close(2),-,-,!close(2)>,
<-,-,-,?down,!down>,<-,-,-,?up,!up>}



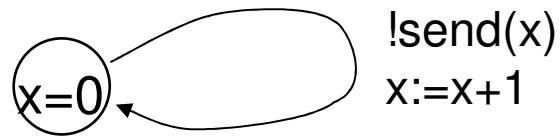
Asynchronous Messages

- Like message passing, but messages are not received instantly.
- Emitted messages but not yet received remain in a **communication channel**, usually a FIFO buffer
- A communication channel can be modeled by a transition system with a variable (for the buffer content)

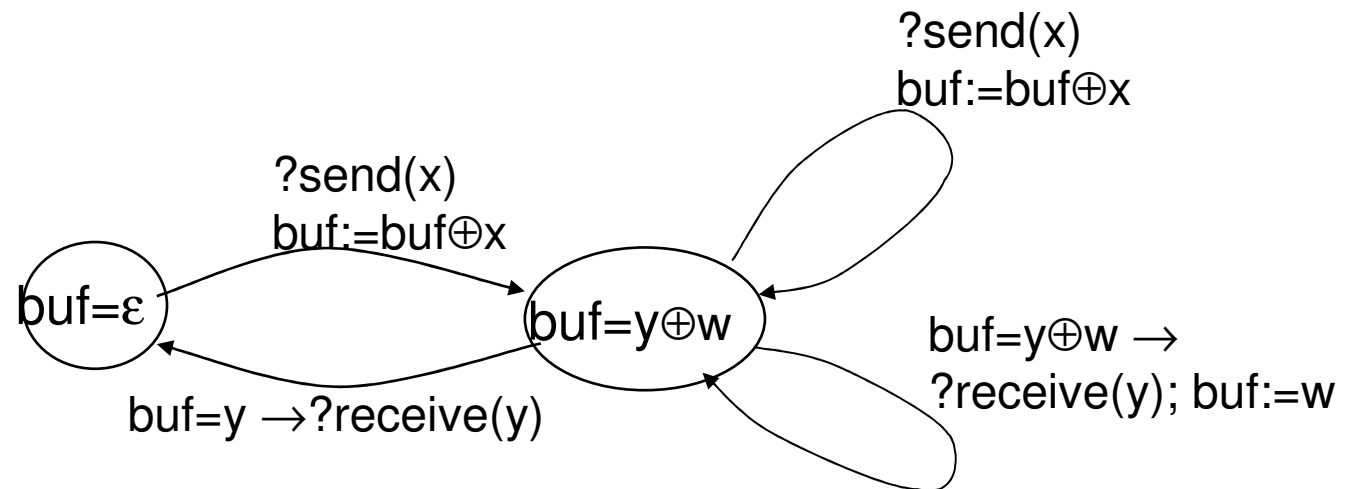


Example:

■ Producer



■ Buffer



■ Consumer

