Is Timing Analysis a Refinement of Causality Analysis?

Joaquín Aguado
Informatics Theory Group
University of Bamberg

PRETSY-Meeting, Bamberg

April 2013

Why Causality Analysis is Important

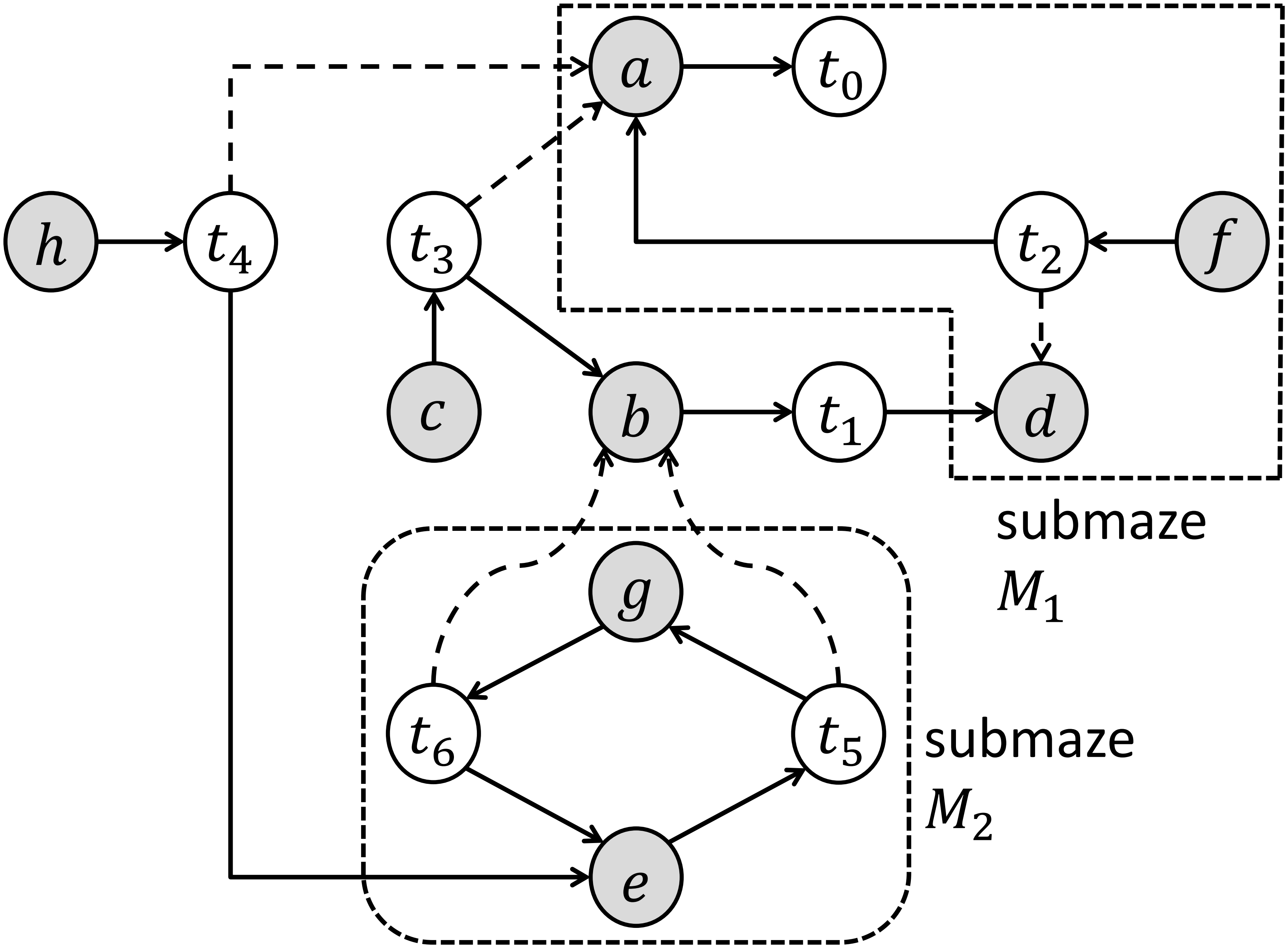
Causality analysis needs to be taken seriously.

“For acyclic circuits the analysis is unnecessary, only a problem for Esterel not for other languages such as Lustre”

The key difference between cyber physical and embedded systems is that the former are subject to strong physical control. The term "cyber" comes from "Kybernetik" and means control loops.

But control loops are (instantaneous) cyclic interactions, and causality analysis (stability, convergence) is an essential ingredient in control theory!

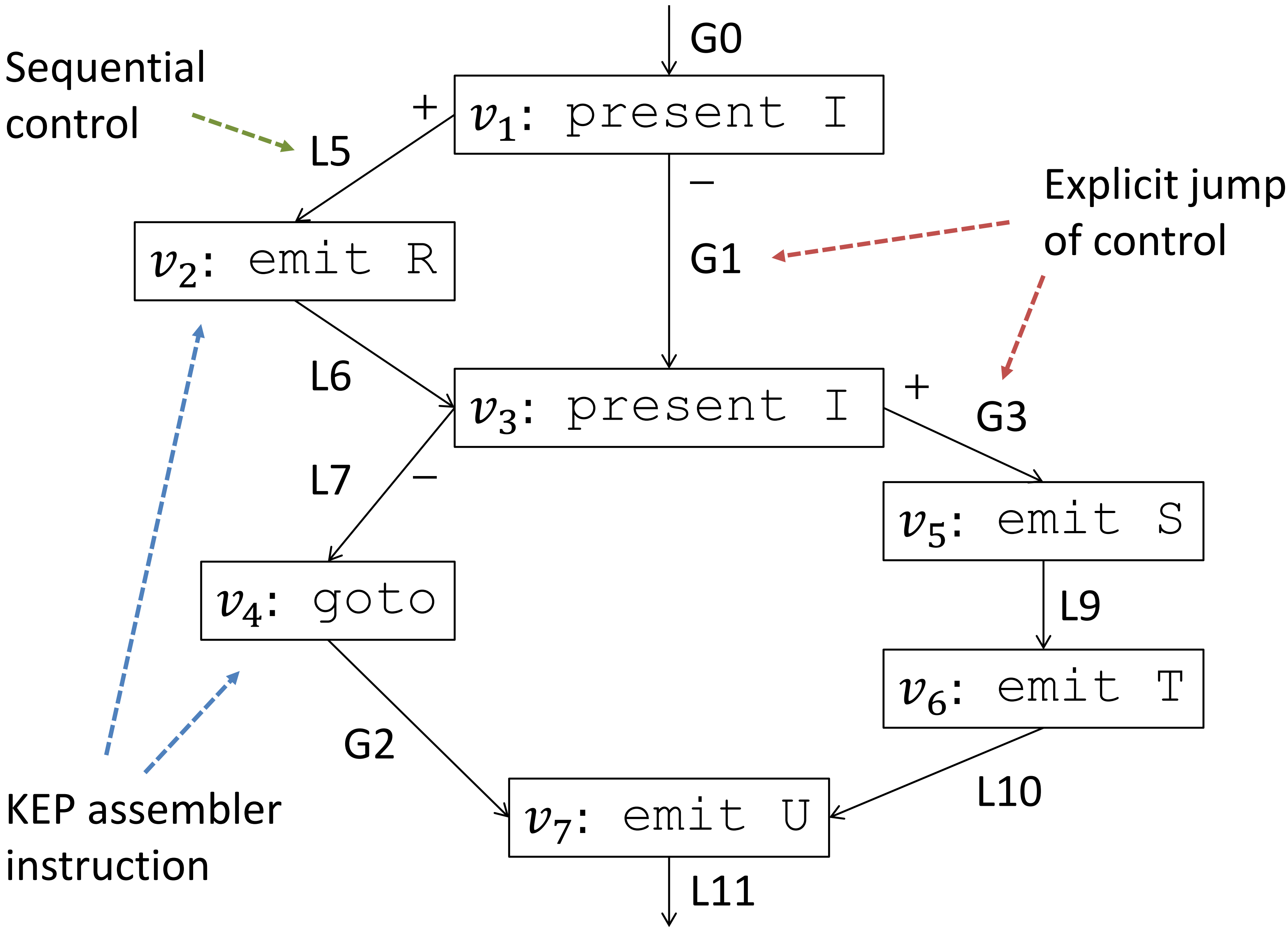
Playing The Maze Game



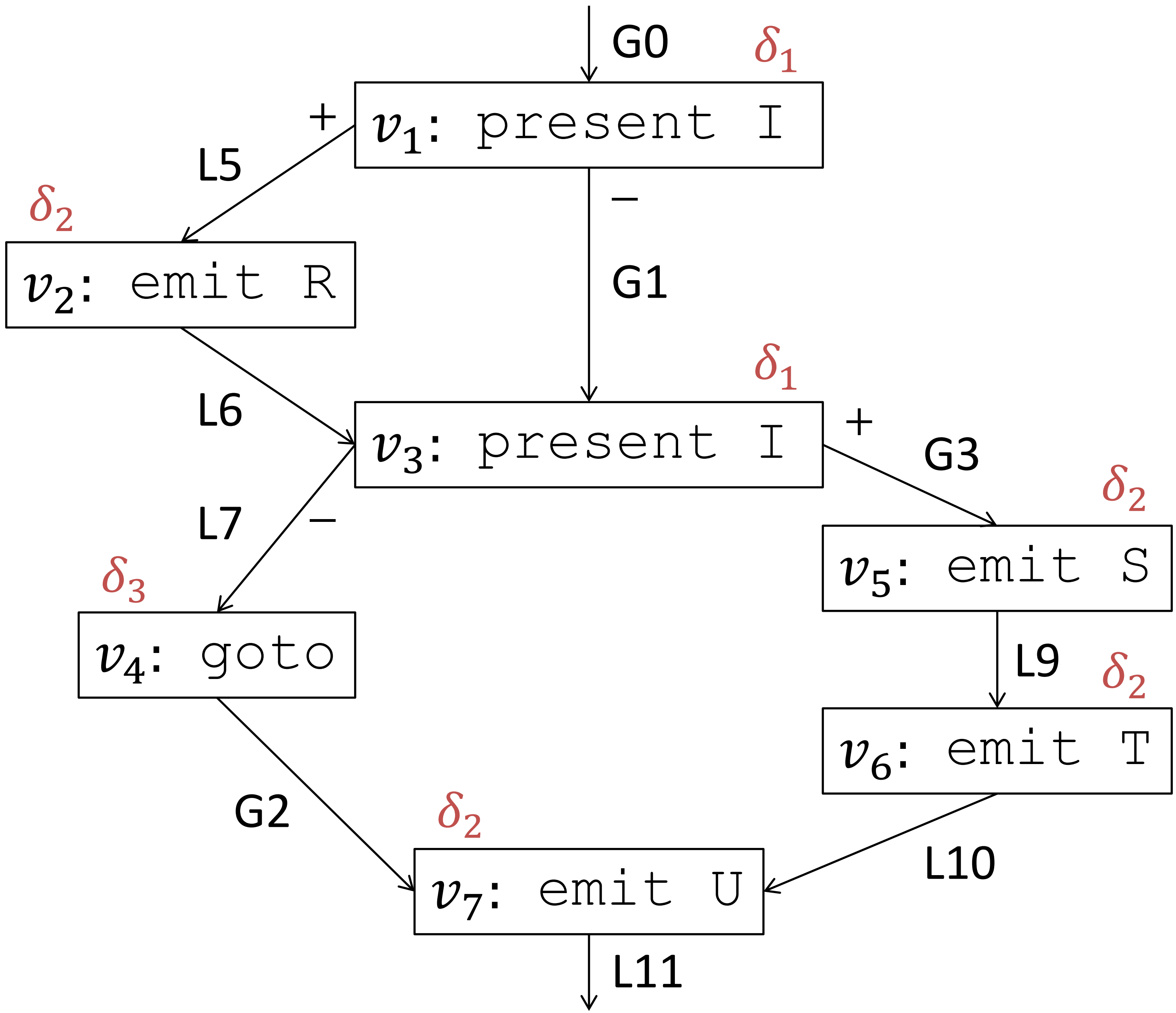
Program of the Maze

```
emit a                                ||
present d then emit b end             ||
present a then
     present d else emit f end
end                                   ||
present a else
     present b then emit c end
end                                   ||
present e then emit h end             ||
present b else
     present g then emit e end
end                                   ||
present e then emit g end
```

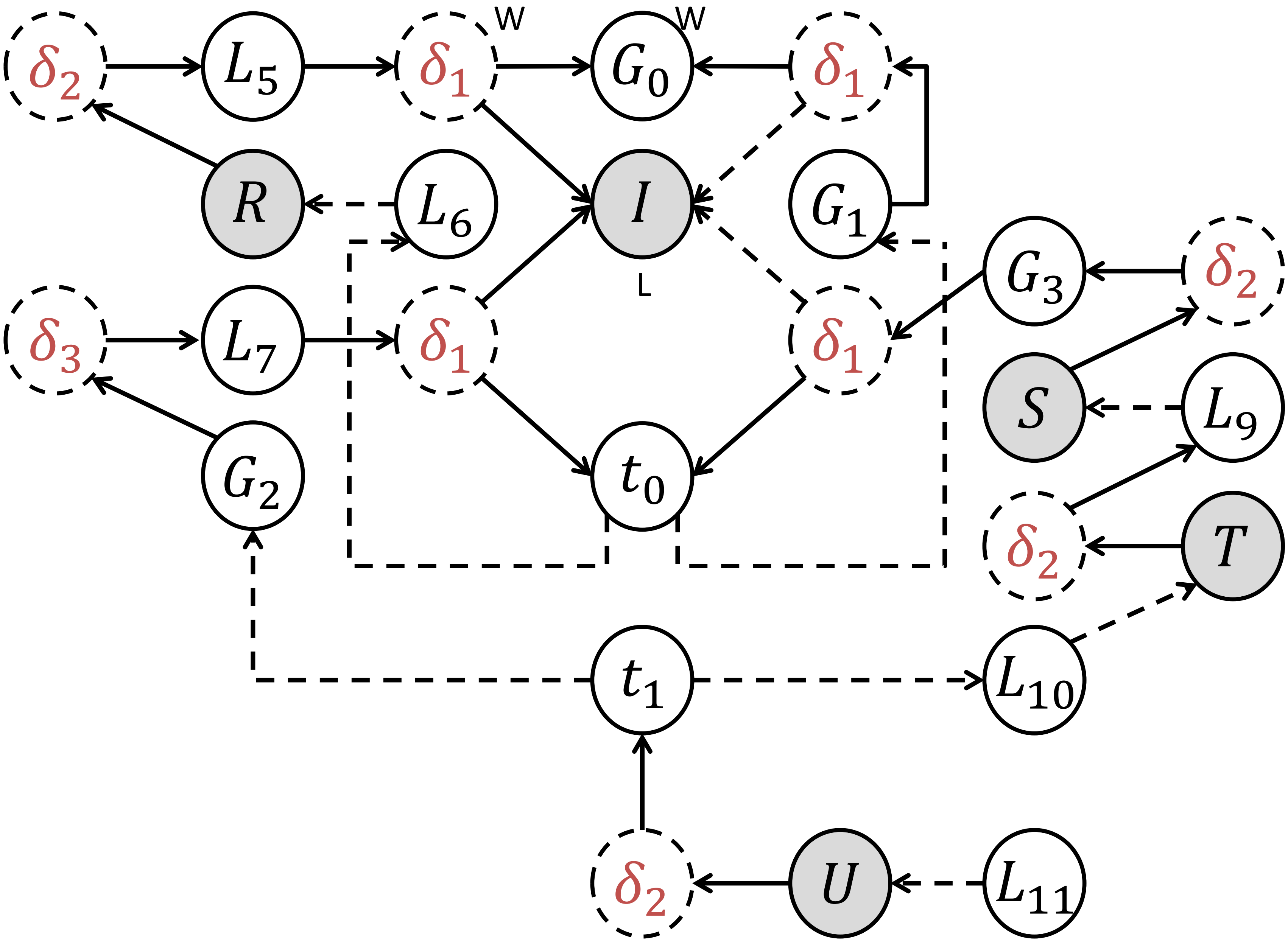
Sequential Block



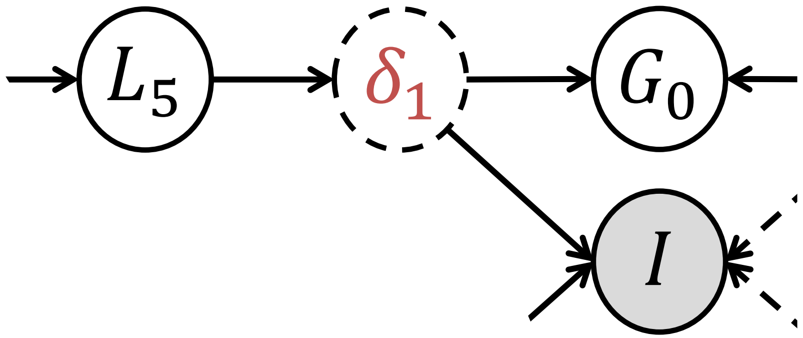
Sequential Block + Timing Information



Two-Player Timed Maze

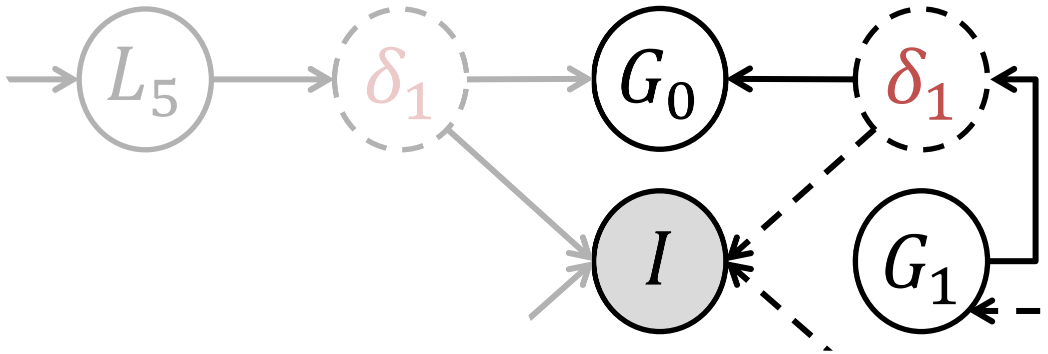


WCRT Interface Algebra



$$[\delta_1]: G0 \wedge I \supset \circ L5$$

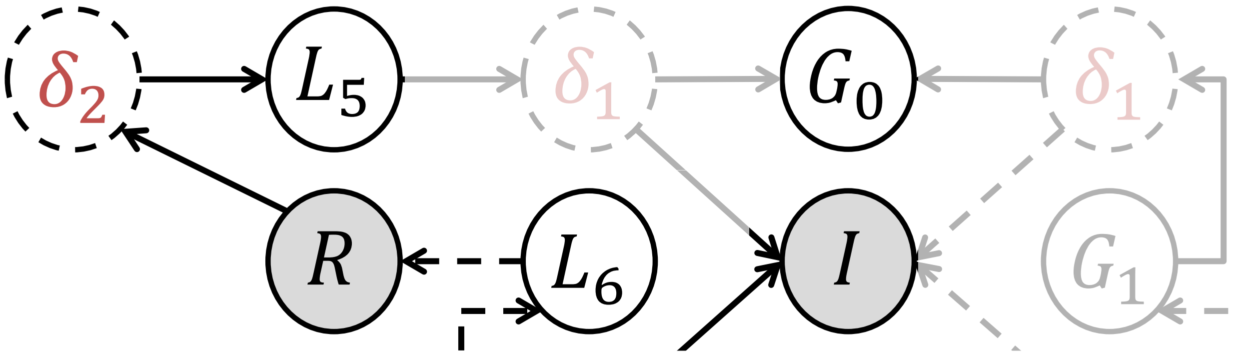
WCRT Interface Algebra



$[\delta_1]: G0 \wedge I \supset \circ L5$

$[\delta_1]: G0 \wedge \neg I \supset \circ G1$

WCRT Interface Algebra

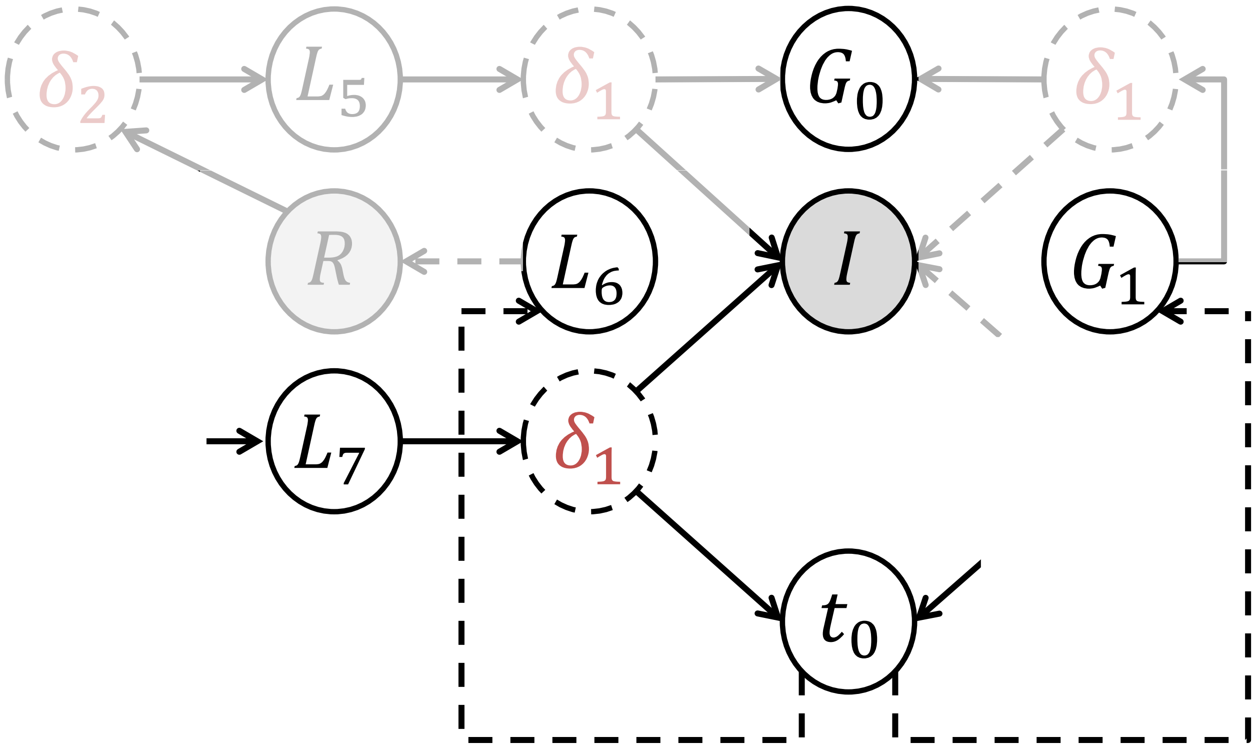


$[\delta_1]: G0 \wedge I \supset \circ L5$

$[\delta_1]: G0 \wedge \neg I \supset \circ G1$

$[\delta_2]: L5 \supset \circ L6$

WCRT Interface Algebra



$[\delta_1]: G0 \wedge I \supset \circ L5$

$[\delta_1]: G0 \wedge \neg I \supset \circ G1$

$[\delta_2]: L5 \supset \circ L6$

$[\delta_1\ \delta_1]: (L6 \vee G1) \wedge I \supset \circ L7$

WCRT Interface Algebra

$$v_1^+ = [\delta_1]\colon G0 \wedge I \supset \circ L5$$
$$v_1^- = [\delta_1]\colon G0 \wedge \neg I \supset \circ G1$$
$$v_3^+ = [\delta_1\ \delta_1]\colon (L6 \vee G1) \wedge I \supset \circ L7$$
$$v_3^- = [\delta_1\ \delta_1]\colon (L6 \vee G1) \wedge \neg I \supset \circ G3$$
$$v_7 = [\delta_2\ \delta_2]\colon (G2 \vee L10) \supset \circ L11$$
$$v_2 = [\delta_2]\colon L5 \supset \circ L6$$
$$v_5 = [\delta_2]\colon G3 \supset \circ L9$$
$$v_4 = [\delta_3]\colon L7 \supset \circ G2$$
$$v_6 = [\delta_2]\colon L9 \supset \circ L10$$

This specification is **open**: It does not preclude that the context adds extra jumps into or emissions in G.

WCRT Interface Algebra

$v_1^+ = [\delta_1]: G0 \wedge I \supset \circ L5$

$v_1^- = [\delta_1]: G0 \wedge \neg I \supset \circ G1$

$v_3^+ = [\delta_1\ \delta_1]: (L6 \vee G1) \wedge I \supset \circ L7$

$v_3^- = [\delta_1\ \delta_1]: (L6 \vee G1) \wedge \neg I \supset \circ G3$

$v_7 = [\delta_2\ \delta_2]: (G2 \vee L10) \supset \circ L11$

$v_2 = [\delta_2]: L5 \supset \circ L6$

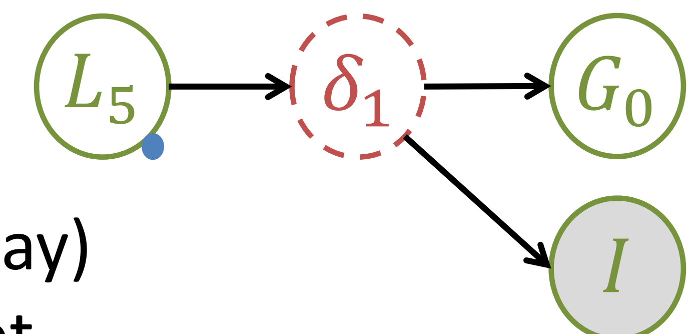
$v_5 = [\delta_2]: G3 \supset \circ L9$

$v_4 = [\delta_3]: L7 \supset \circ G2$

$v_6 = [\delta_2]: L9 \supset \circ L10$

This specification is **open**: It does not preclude that the context adds extra jumps into or emissions in G.

$$G0 \wedge I \supset \circ L5$$



Species that L5 is activated (with delay) whenever control reaches G0 but not what happens if G0 is never activated.

WCRT Interface Algebra

$v_1^+ = [\delta_1]: G0 \wedge I \supset \circ L5$

$v_1^- = [\delta_1]: G0 \wedge \neg I \supset \circ G1$

$v_3^+ = [\delta_1\ \delta_1]: (L6 \vee G1) \wedge I \supset \circ L7$

$v_3^- = [\delta_1\ \delta_1]: (L6 \vee G1) \wedge \neg I \supset \circ G3$

$v_7 = [\delta_2\ \delta_2]: (G2 \vee L10) \supset \circ L11$

$v_2 = [\delta_2]: L5 \supset \circ L6$

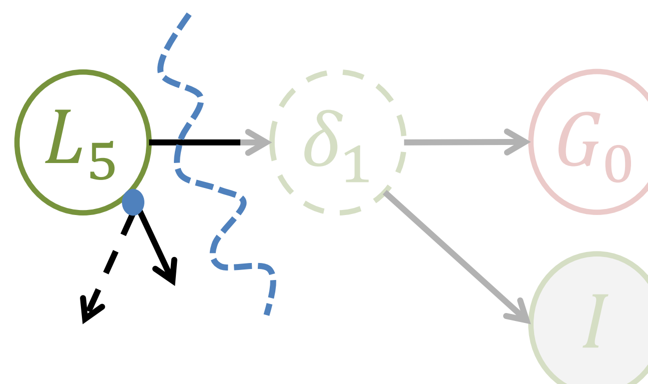
$v_5 = [\delta_2]: G3 \supset \circ L9$

$v_4 = [\delta_3]: L7 \supset \circ G2$

$v_6 = [\delta_2]: L9 \supset \circ L10$

In an **open** system L5 may still be activated by jumps from the program environment of G

$$G0 \wedge I \supset \circ L5$$



WCRT Interface Algebra

$v_1^+ = [\delta_1]: G0 \wedge I \supset \circ L5$

$v_1^- = [\delta_1]: G0 \wedge \neg I \supset \circ G1$

$v_3^+ = [\delta_1\ \delta_1]: (L6 \vee G1) \wedge I \supset \circ L7$

$v_3^- = [\delta_1\ \delta_1]: (L6 \vee G1) \wedge \neg I \supset \circ G3$

$v_7 = [\delta_2\ \delta_2]: (G2 \vee L10) \supset \circ L11$

$v_2 = [\delta_2]: L5 \supset \circ L6$

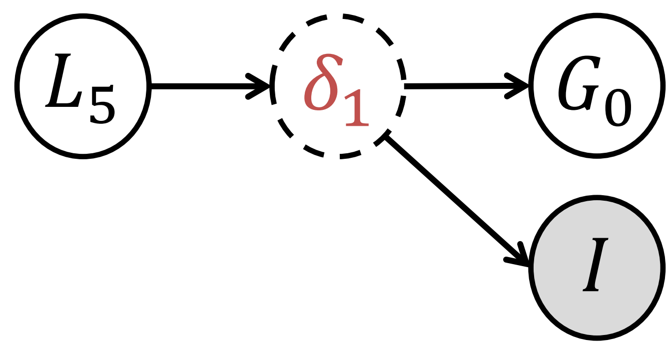
$v_5 = [\delta_2]: G3 \supset \circ L9$

$v_4 = [\delta_3]: L7 \supset \circ G2$

$v_6 = [\delta_2]: L9 \supset \circ L10$

If we want to make L5 inaccessible from outside, we close the specification with an extra clause.

$$G0 \wedge I \supset \circ L5$$
$$\neg G0 \oplus \neg I \supset \circ \neg L5$$



WCRT Interface Algebra

$v_1^+ = [\delta_1]: G0 \wedge I \supset \circ L5$

$v_1^- = [\delta_1]: G0 \wedge \neg I \supset \circ G1$

$v_3^+ = [\delta_1\ \delta_1]: (L6 \vee G1) \wedge I \supset \circ L7$

$v_3^- = [\delta_1\ \delta_1]: (L6 \vee G1) \wedge \neg I \supset \circ G3$

$v_7 = [\delta_2\ \delta_2]: (G2 \vee L10) \supset \circ L11$

$v_2 = [\delta_2]: L5 \supset \circ L6$

$v_5 = [\delta_2]: G3 \supset \circ L9$

$v_4 = [\delta_3]: L7 \supset \circ G2$

$v_6 = [\delta_2]: L9 \supset \circ L10$

The WCRT for G amounts to obtaining the worst-case (tightest) bound δ such that:

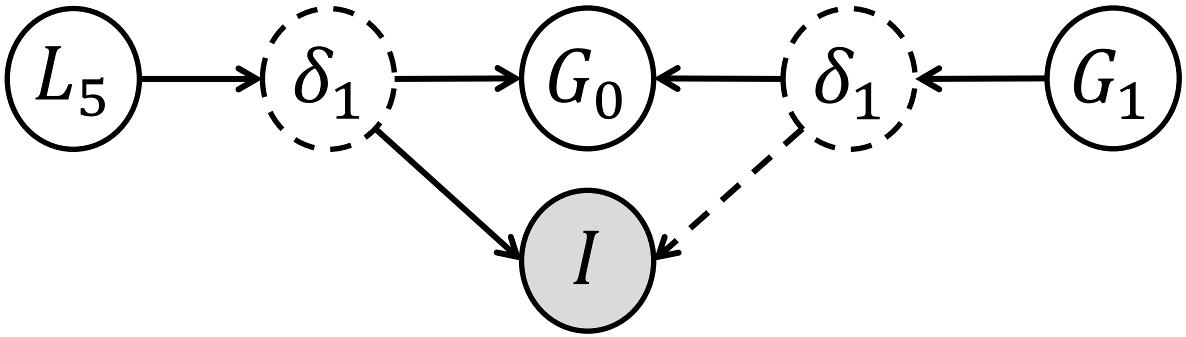
$$\bigwedge_{i=1}^{7} v_i^{\{+,-\}} \preccurlyeq [\delta]:\ G0 \wedge (I \oplus \neg I) \supset \circ L11$$

where $\varphi \preccurlyeq \psi$ (models inclusion): All schedules that satisfy φ also satisfy ψ.

Abstraction:
Over and Under approximation

Ignoring Signal Dependencies

This is a standard abstraction.

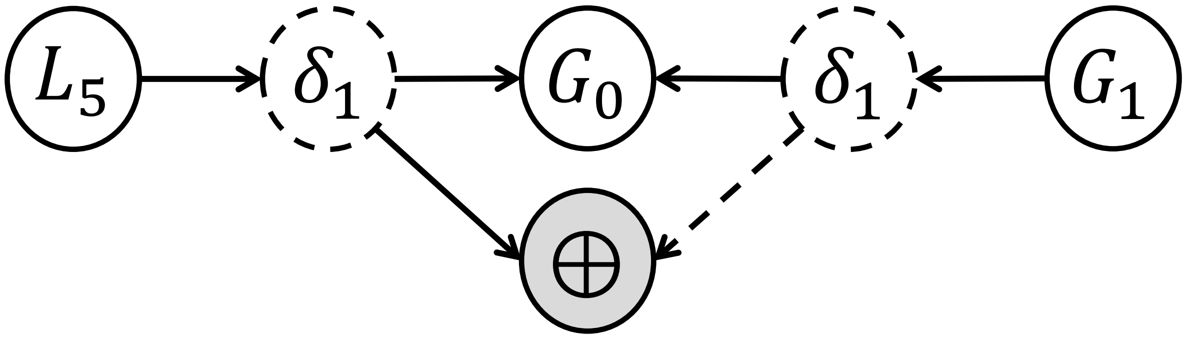


$$[\delta_1\ \delta_1]\colon\ G0 \wedge (I \oplus \neg I) \supset \circ\, G1 \oplus \circ\, L5$$

The associated type specifies that any set of schedules passing through $G0$ when signal I is decided splits non-deterministically into a subset satisfying $\circ\, G1$ and other satisfying $\circ\, L5$.

Ignoring Signal Dependencies

This is a standard abstraction.

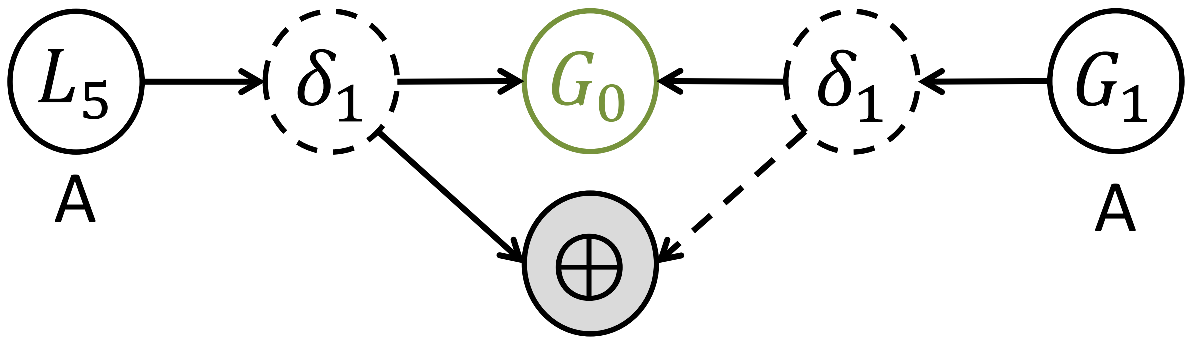


$$[\delta_1\ \delta_1]\colon\ G0 \supset\circ\ G1 \oplus \circ\ L5$$

If signal I is known to be stable $I \oplus \neg I \ \cong true$ then we can drop the condition.

Ignoring Signal Dependencies

This is a standard abstraction.

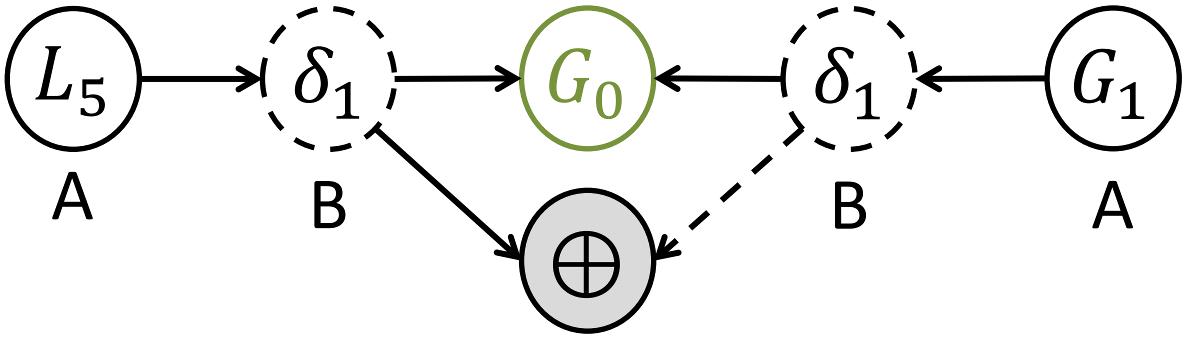


$$[\delta_1\ \delta_1]\text{: } G0 \supset\circ\ G1 \oplus \circ\ L5$$

If signal I is known to be stable $I \oplus \neg I \ \cong true$ and we can drop the condition.

Ignoring Signal Dependencies

This is a standard abstraction.

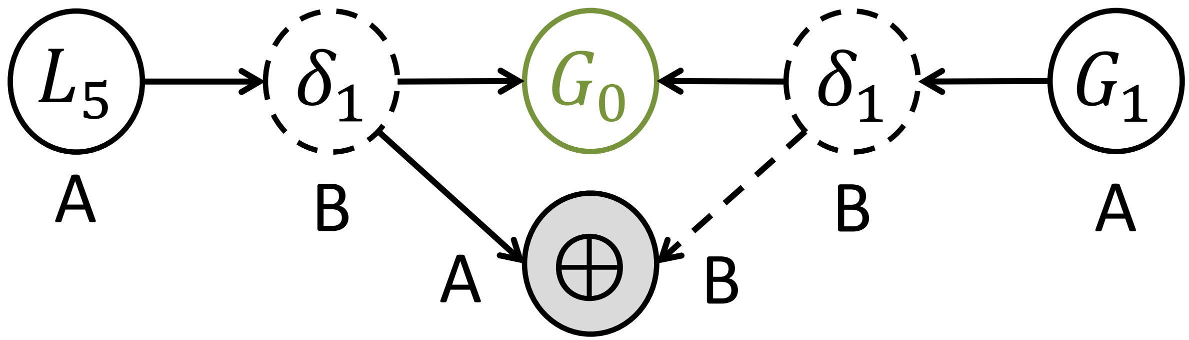


$$[\delta_1\ \delta_1]:\ G0 \supset\circ G1 \oplus \circ L5$$

If signal I is known to be stable $I \oplus \neg I \cong true$ and we can drop the condition.

Ignoring Signal Dependencies

This is a standard abstraction.

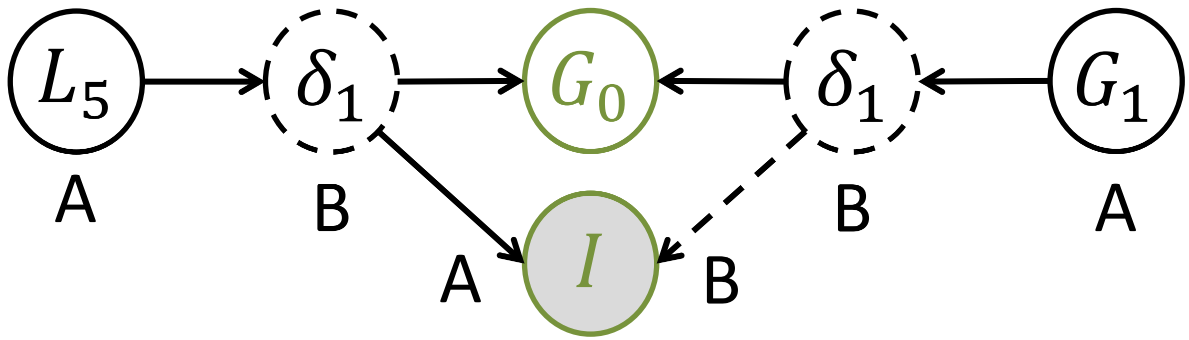


$$[\delta_1\ \delta_1]\text{: } G0 \supset\circ\ G1 \oplus \circ\ L5$$

If signal I is known to be stable $I \oplus \neg I \cong true$ and we can drop the condition.

Ignoring Signal Dependencies

This is a standard abstraction.

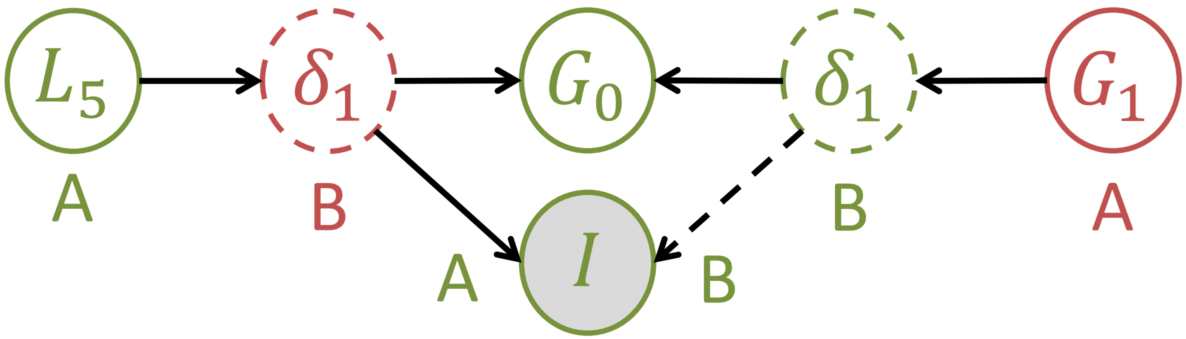


$$[\delta_1\ \delta_1]:\ G0 \supset \circ\, G1 \oplus \circ\, L5$$

If signal I is known to be stable $I \oplus \neg I \cong true$ and we can drop the condition.

Ignoring Signal Dependencies

This is a standard abstraction.

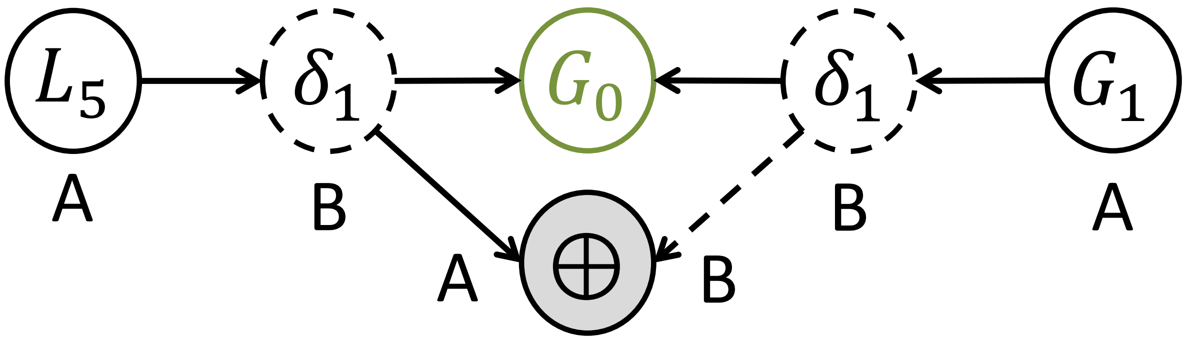


$$[\delta_1\ \delta_1]:\ G0 \supset\circ G1 \oplus \circ L5$$

If signal I is known to be stable $I \oplus \neg I \ \cong true$ and we can drop the condition.

Ignoring Signal Dependencies

This is a standard abstraction.

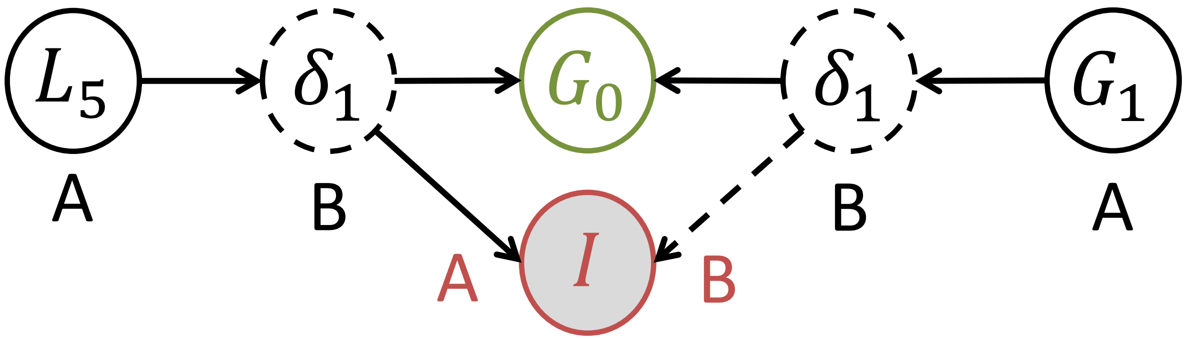


$$[\delta_1\ \delta_1]\text{:}\ G0 \supset\circ\ G1 \oplus \circ\ L5$$

If signal I is known to be stable $I \oplus \neg I \cong true$ and we can drop the condition.

Ignoring Signal Dependencies

This is a standard abstraction.

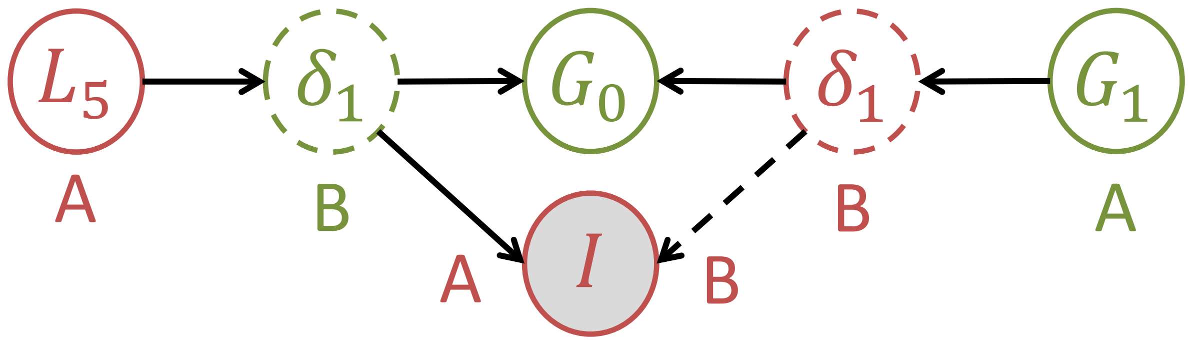


$$[\delta_1\ \delta_1]\colon G0 \supset\circ G1 \oplus \circ L5$$

If signal I is known to be stable $I \oplus \neg I \cong true$ and we can drop the condition.

Ignoring Signal Dependencies

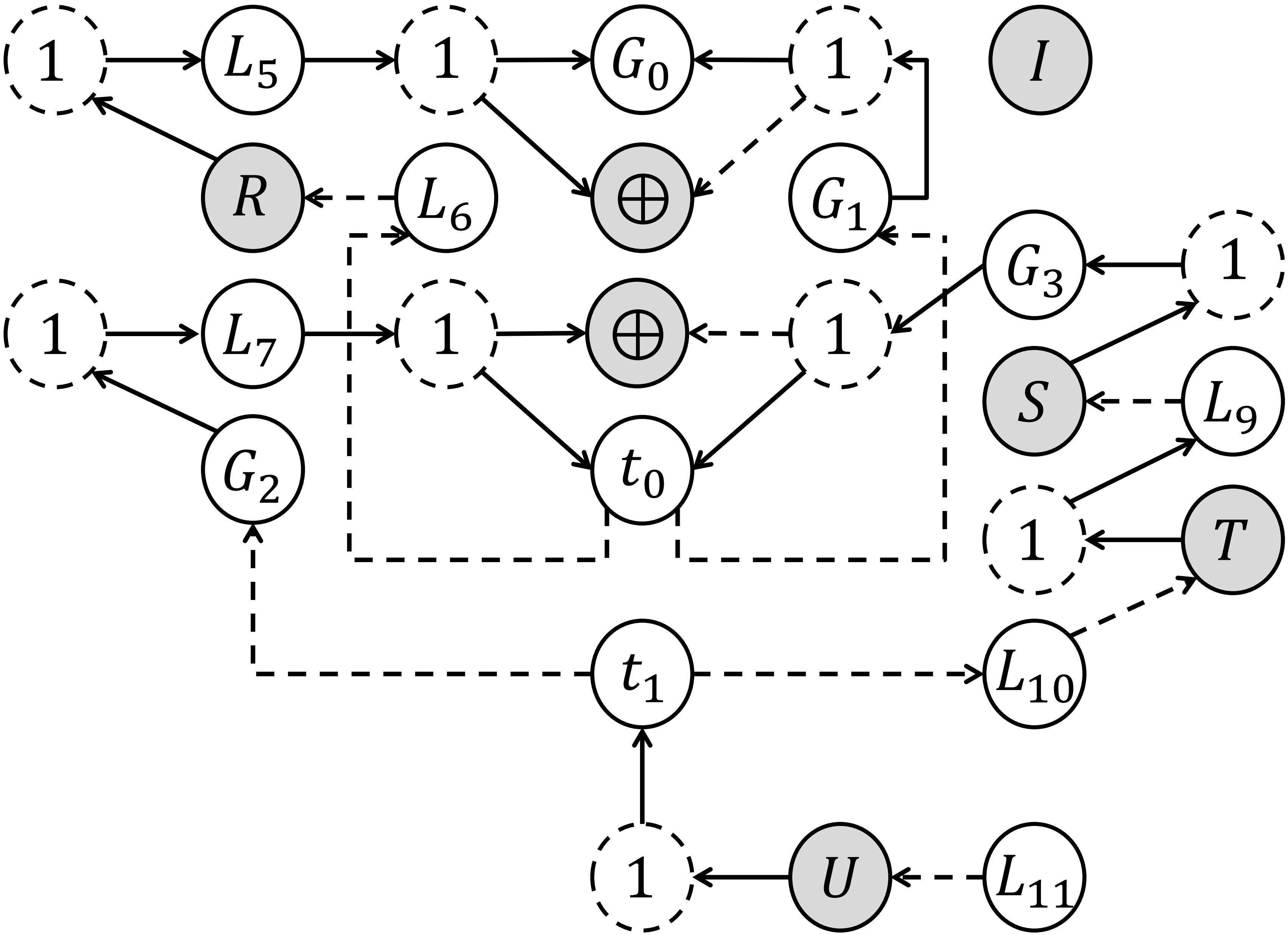
This is a standard abstraction.



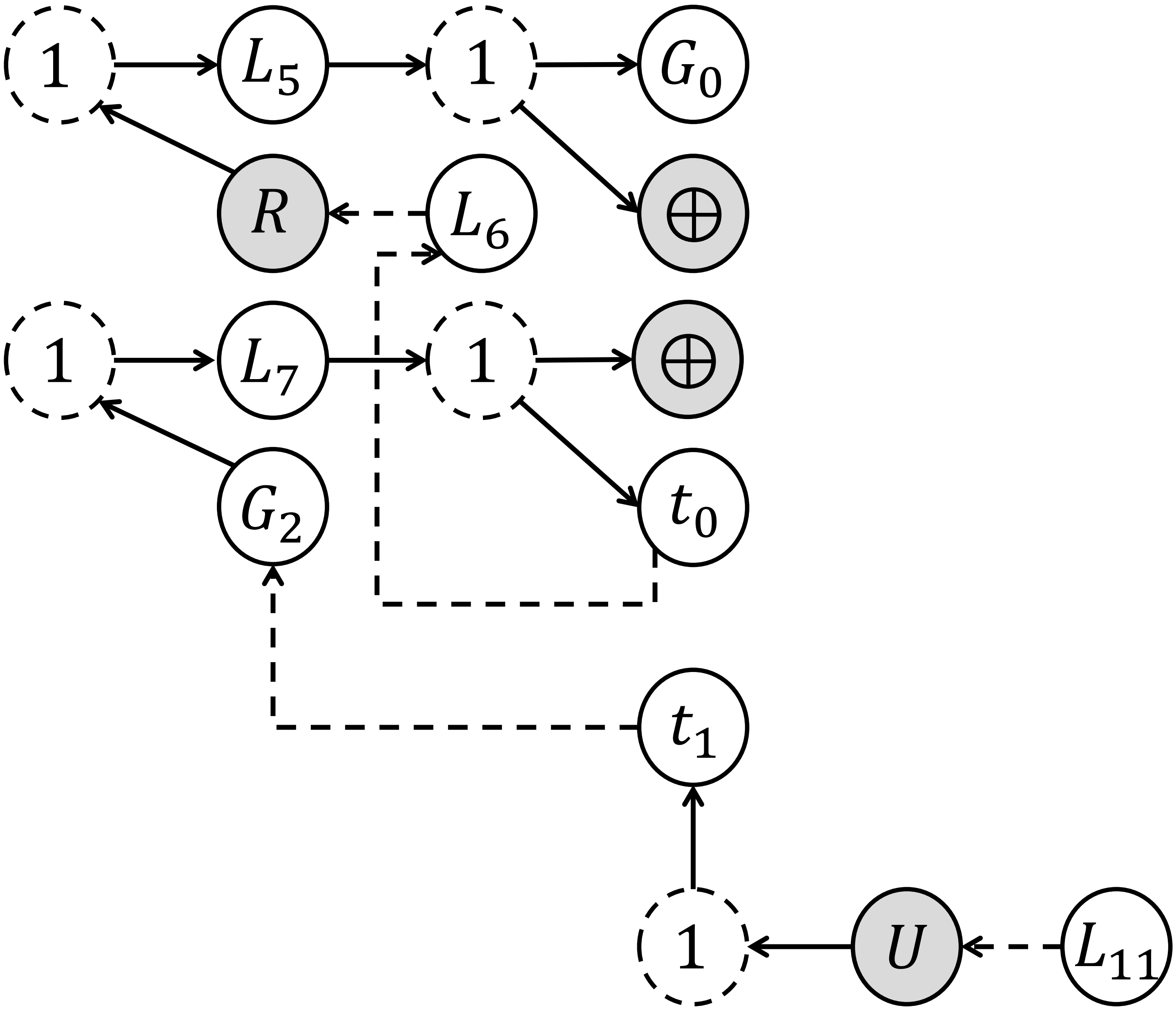
$$[\delta_1\ \delta_1]:\ G0 \supset\circ G1 \oplus \circ L5$$

If signal I is known to be stable $I \oplus \neg I \ \cong true$ and we can drop the condition.

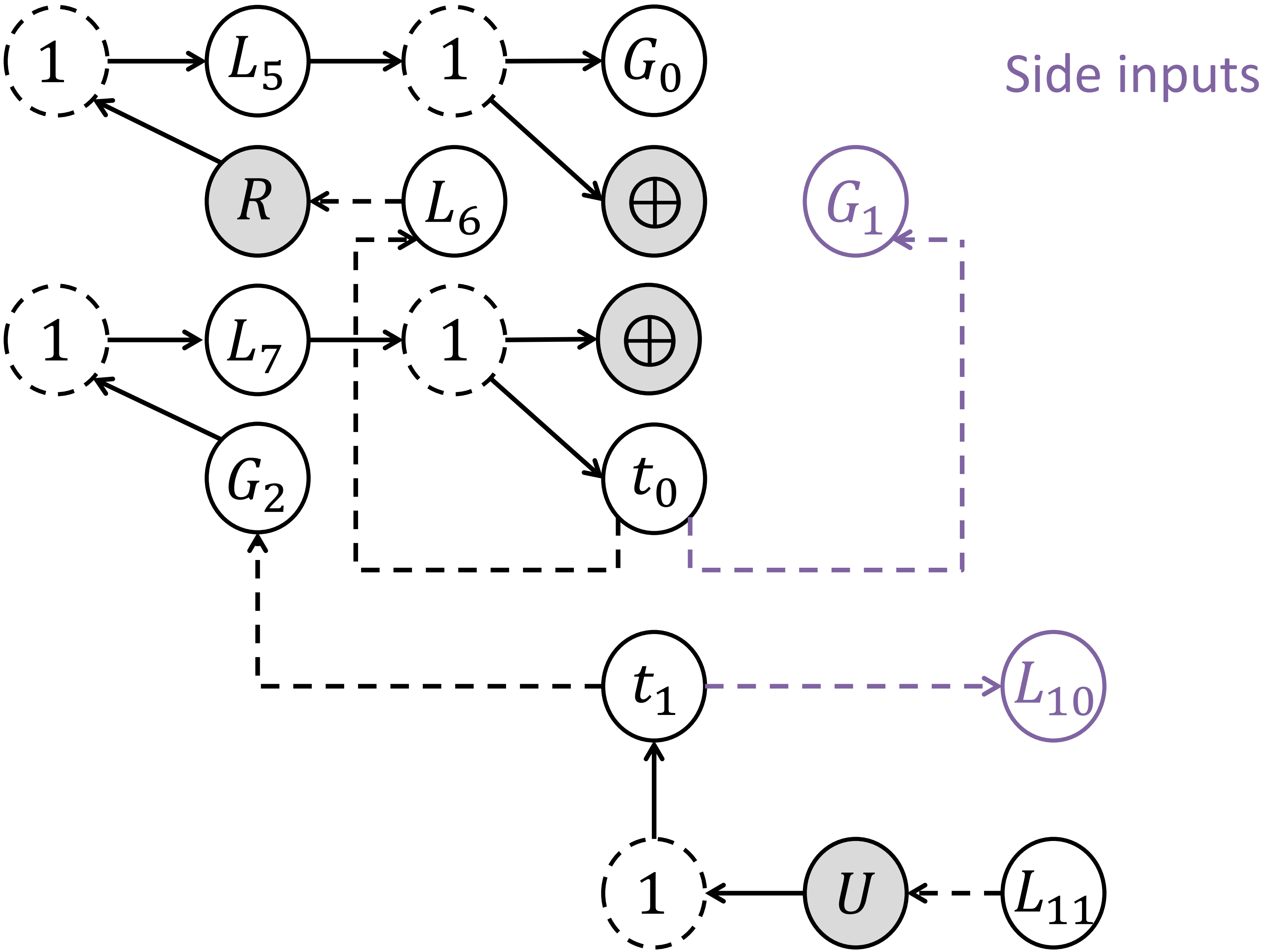
Weaving Paths



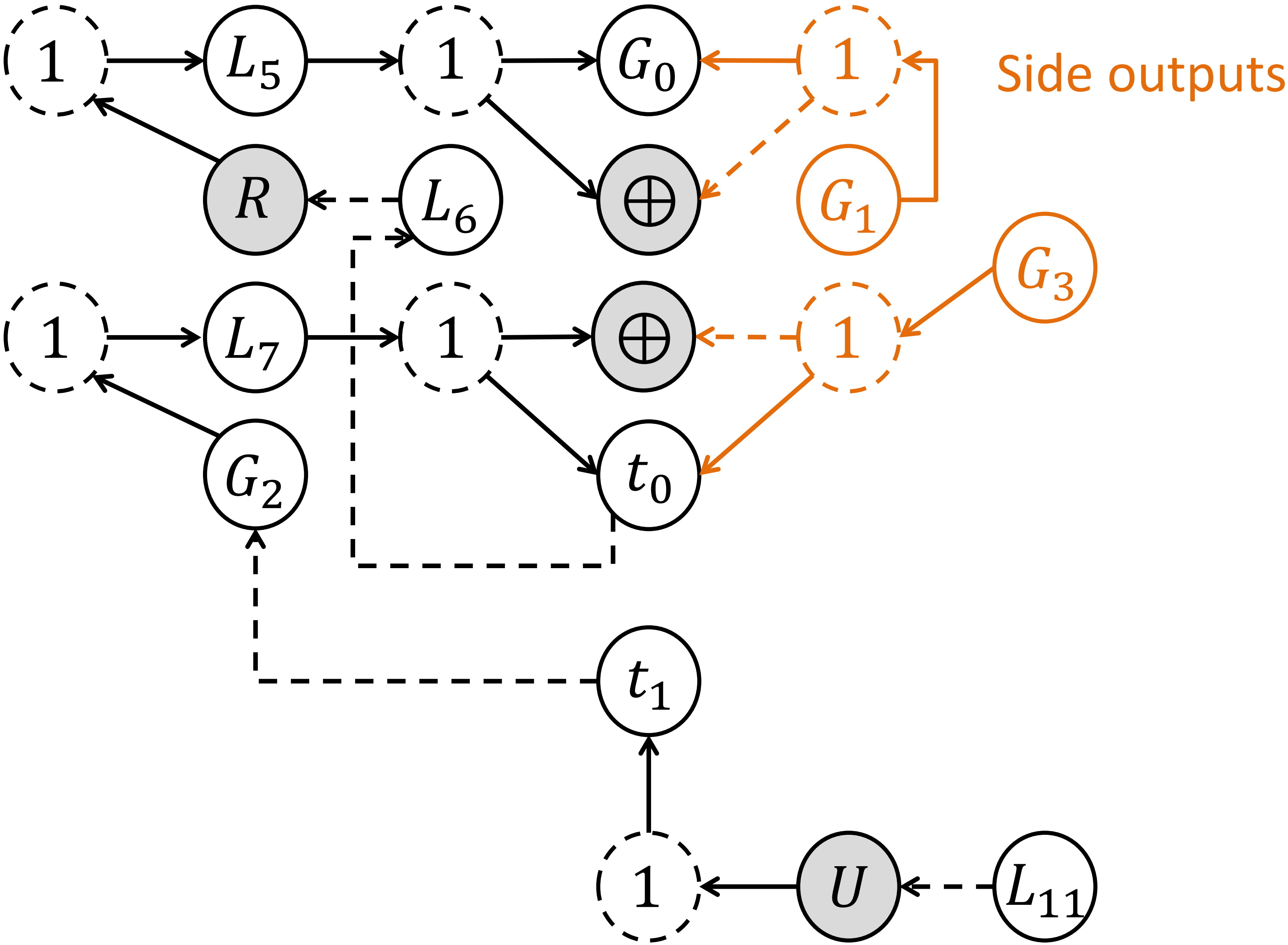
Weaving Paths



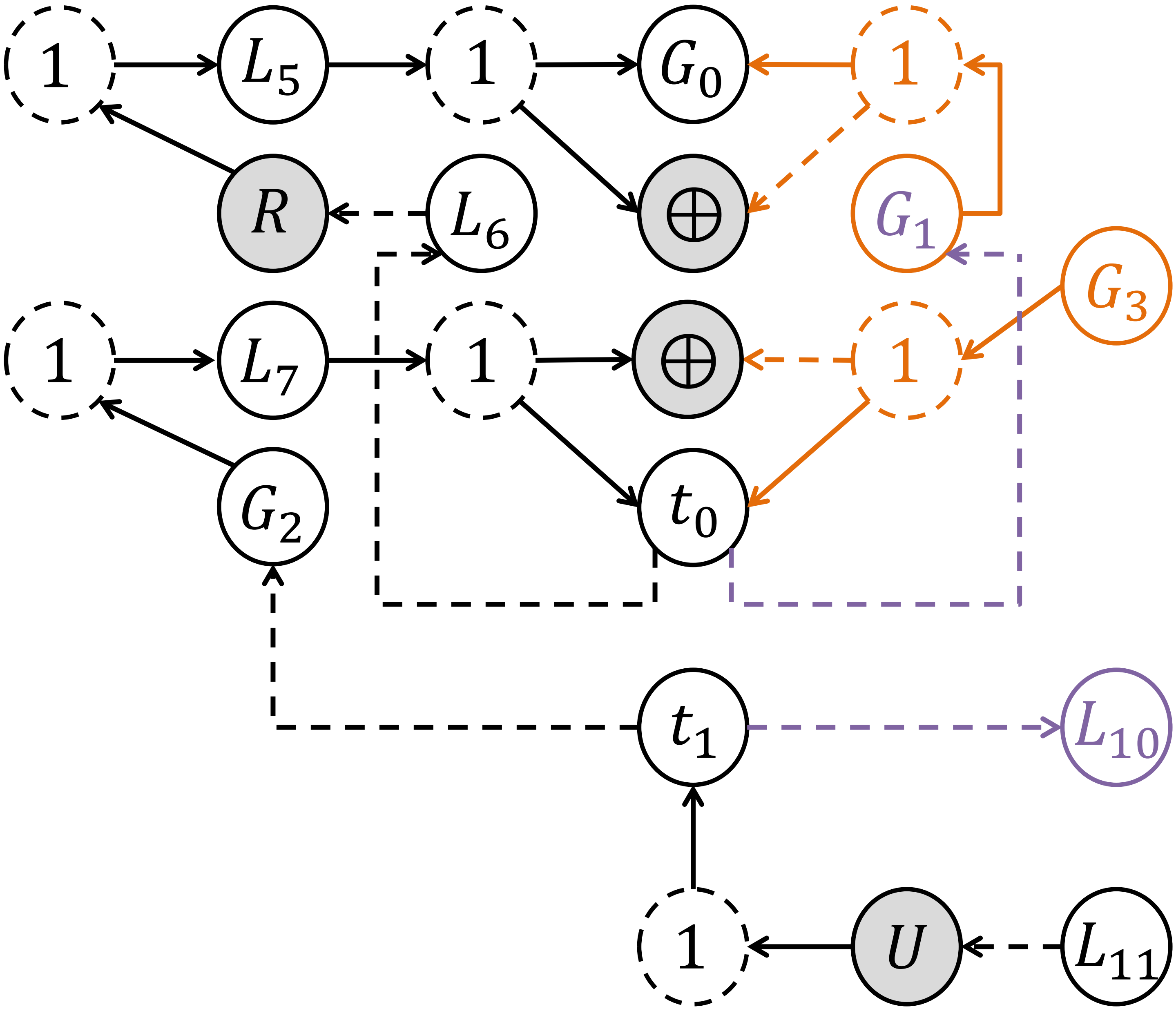
Weaving Paths



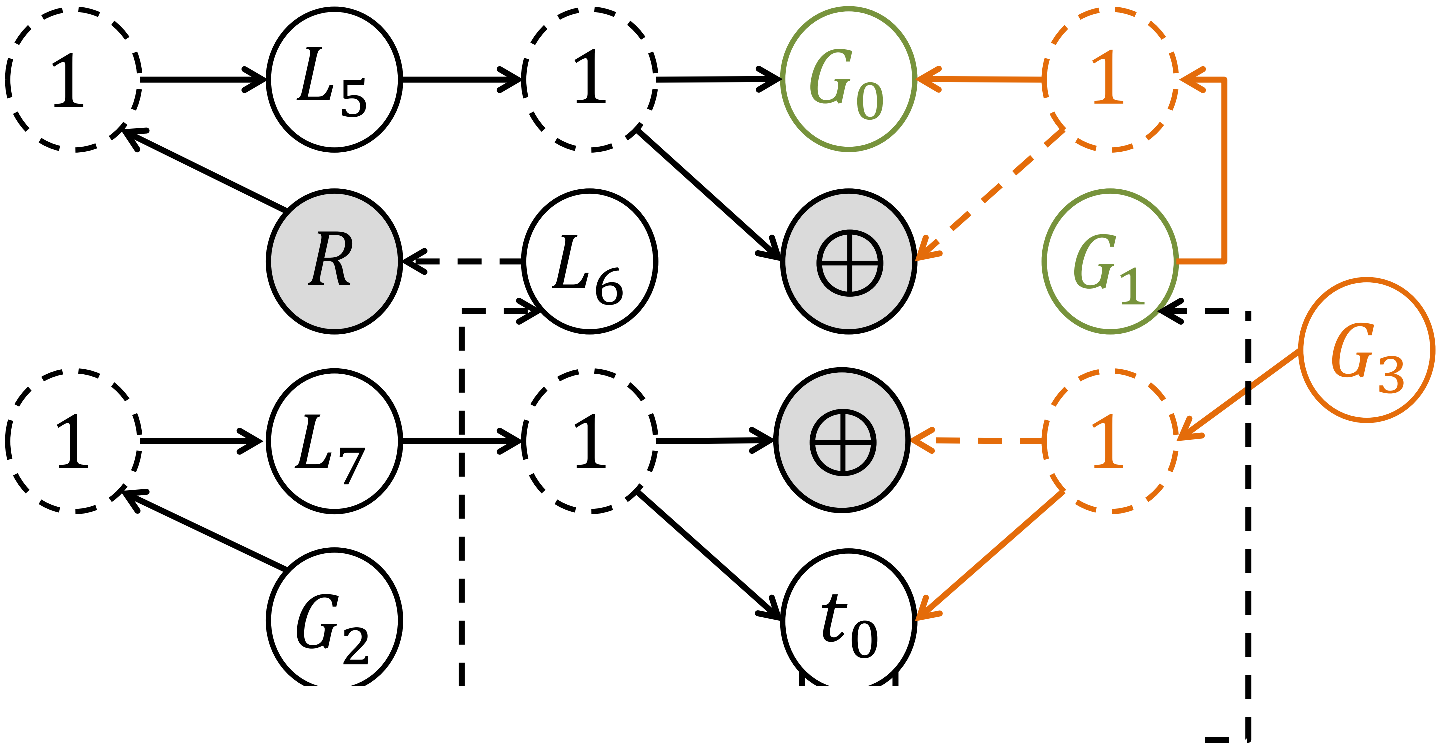
Weaving Paths



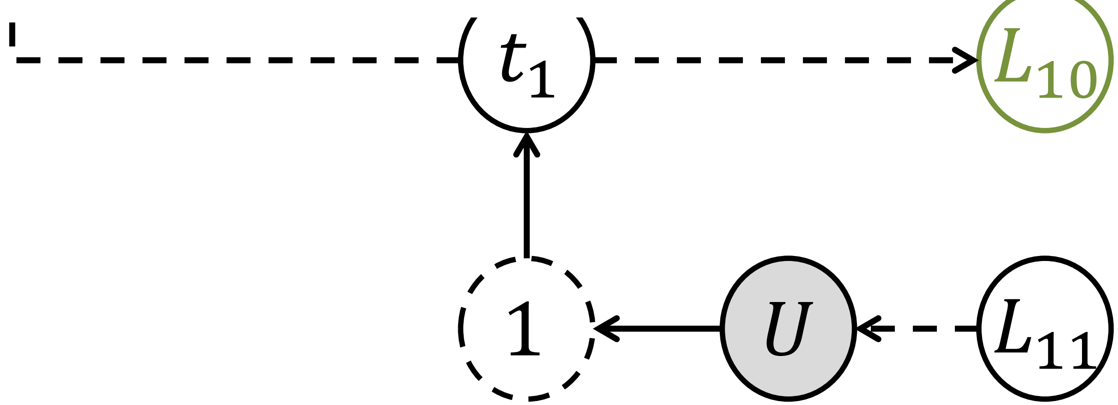
Weaving Paths



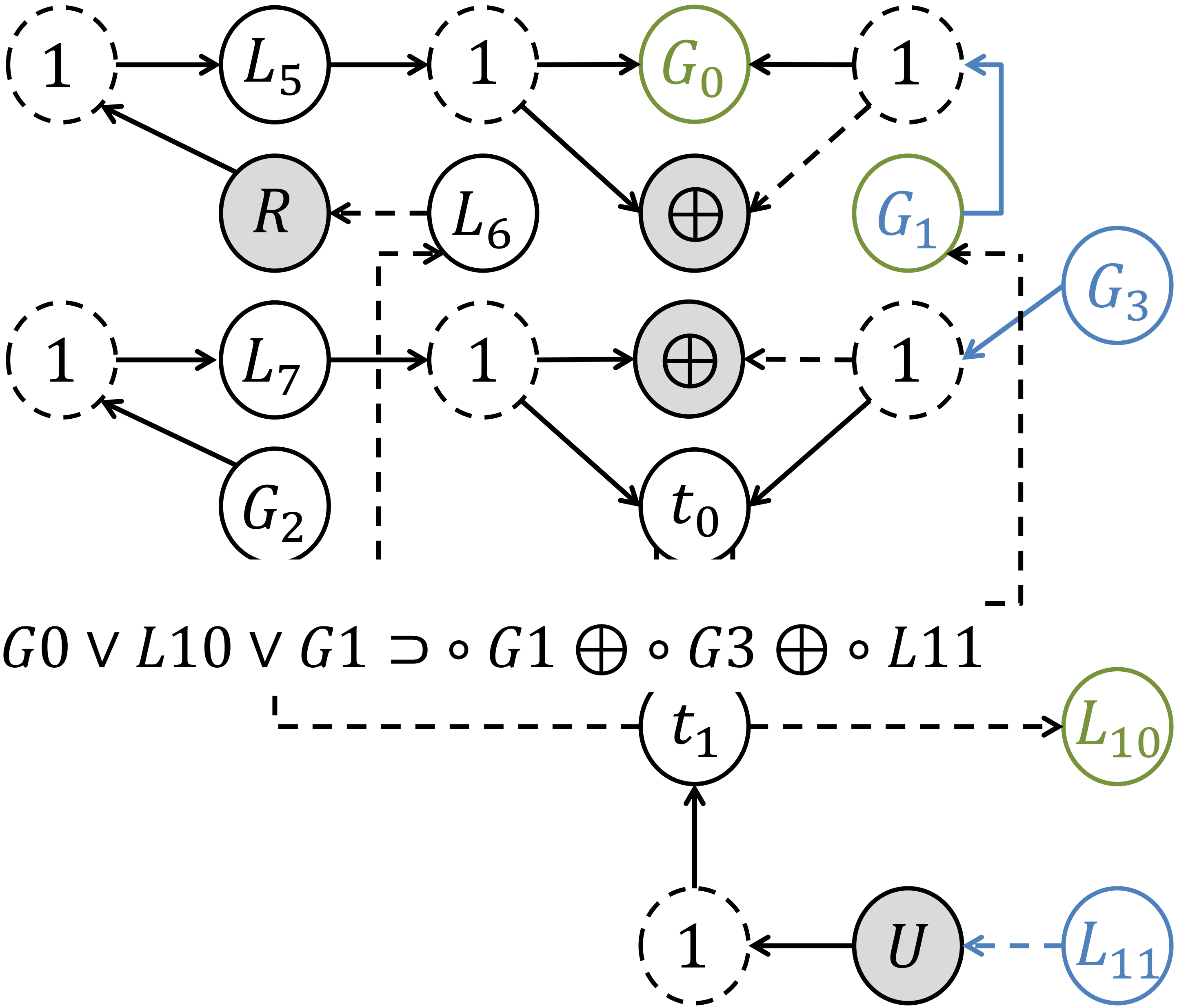
Weaving Paths



$G0 \vee L10 \vee G1 \supset ?$



Weaving Paths



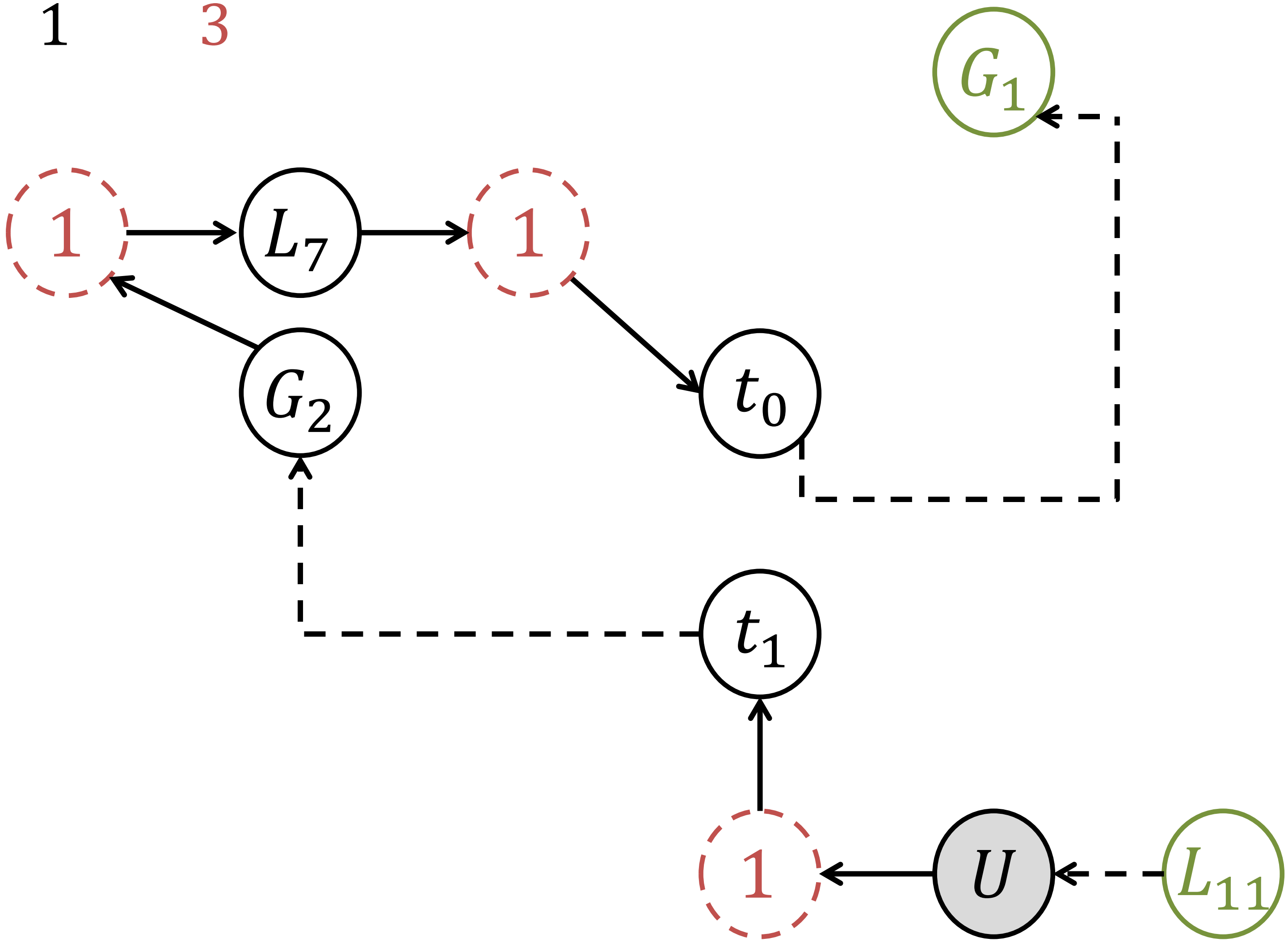
Weaving Paths

$$\begin{matrix} 1 & -\infty & -\infty \\ 3 & -\infty & 1 \\ 5 & 1 & 3 \end{matrix} : G0 \vee L10 \vee G1 \supset \circ G1 \bigoplus \circ G3 \bigoplus \circ L11$$

Weaving Paths

$$\begin{array}{ccc} 1 & -\infty & -\infty \\ 3 & -\infty & 1 \\ 5 & 1 & 3 \end{array}$$

$: G0 \vee L10 \vee G1 \supset \circ G1 \oplus \circ G3 \oplus \circ L11$



Weaving Paths

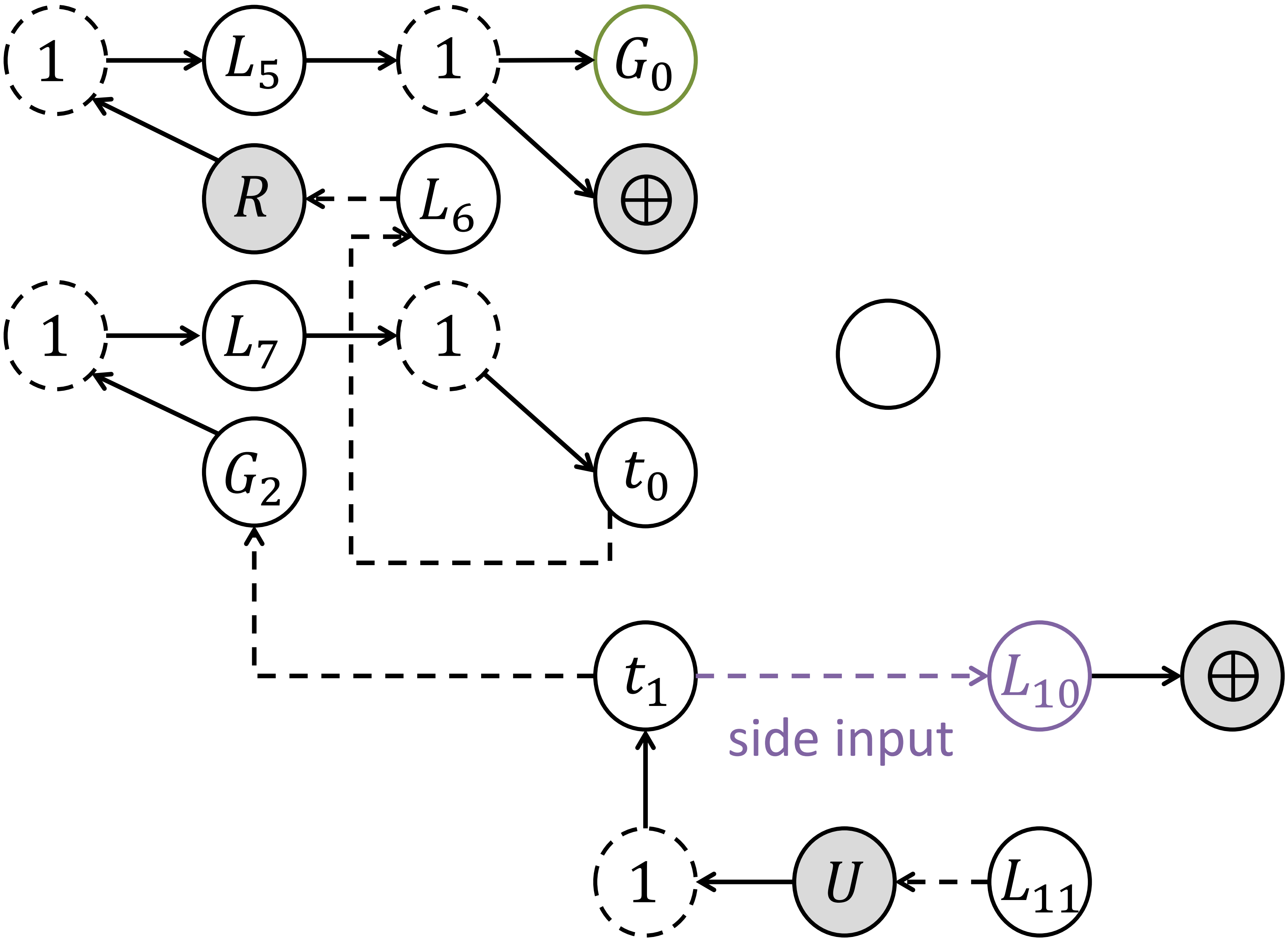
$$\begin{matrix} 1 & -\infty & -\infty \\ 3 & -\infty & 1 \\ 5 & 1 & 3 \end{matrix} : G0 \vee L10 \vee G1 \supset \circ G1 \oplus \circ G3 \oplus \circ L11$$

Assume that for the WCRT we are not interested in the exact timing of the side inputs.

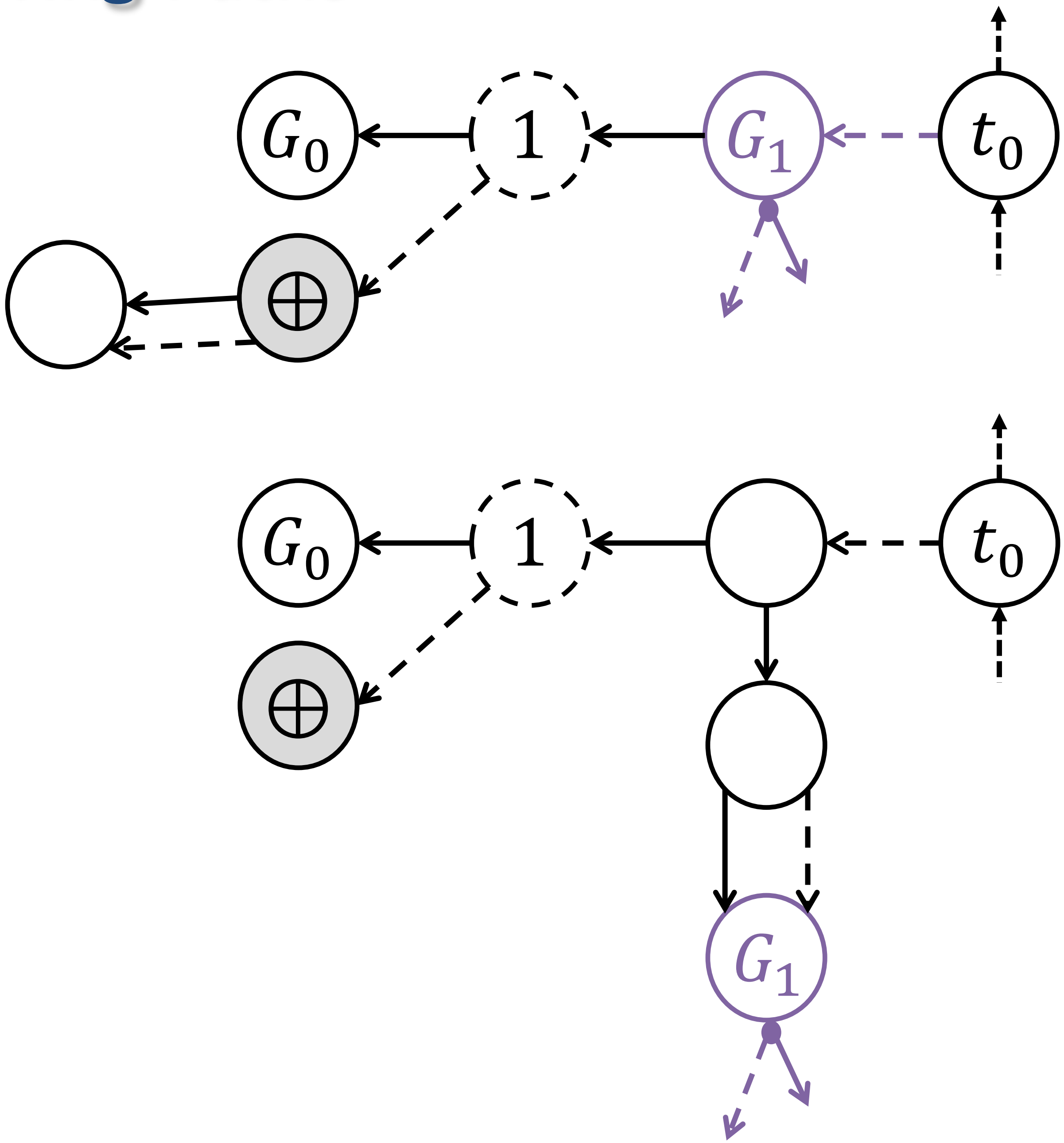
$$\begin{matrix} 0 \\ -\infty \\ -\infty \end{matrix} : G0 \supset \circ G0 \oplus \circ L10 \oplus \circ G1$$

$$\begin{matrix} 1 & -\infty & -\infty \\ 3 & -\infty & 1 \\ 5 & 1 & 3 \end{matrix} \cdot \begin{matrix} 0 \\ -\infty \\ -\infty \end{matrix} = \begin{matrix} 1 \\ 3 \\ 5 \end{matrix} : G0 \supset \circ G1 \oplus \circ G3 \oplus \circ L11$$

Weaving Paths



Weaving Paths



Weaving Paths

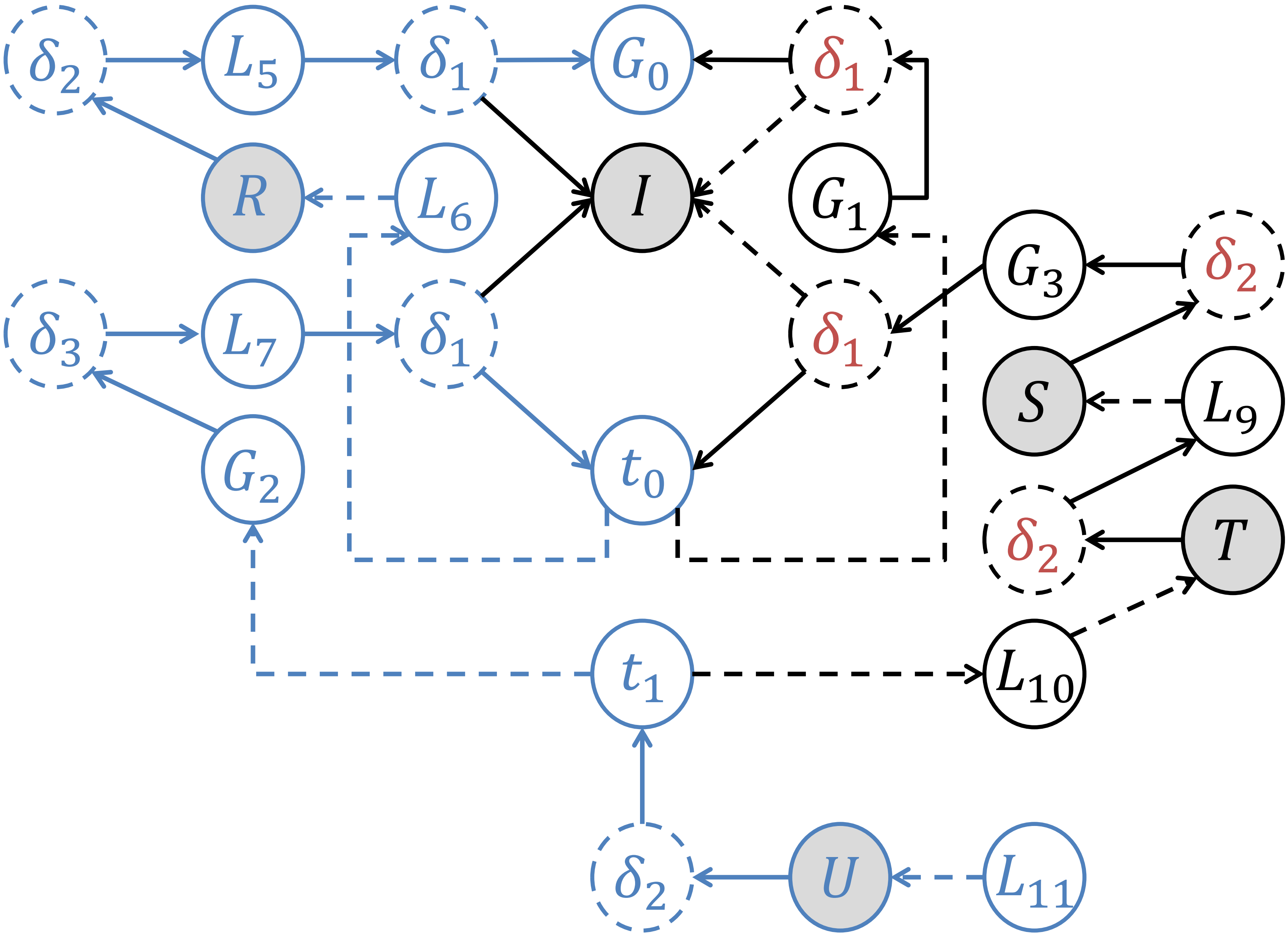
$$\begin{matrix}1\\3\\5\end{matrix} : G0 \supset \circ G1 \oplus \circ G3 \oplus \circ L11$$

Can we also suppress the references to side outputs on the right of the scheduling type?

$$G0 \wedge \neg G1 \wedge \neg G3 \supset \circ f \oplus \circ f \oplus \circ L11$$

$$\begin{matrix}1\\3\\5\end{matrix} \quad \begin{matrix}-\infty & -\infty & 0\\ -\infty & 1 & -\infty\\ 1 & 3 & -\infty\end{matrix} \cdot = \begin{matrix}1\\3\\5\end{matrix} : G0 \supset \circ G1 \oplus \circ G3 \oplus \circ L11$$

Weaving Paths



Playing The Maze Game

P ::=		0	`nothing`
	|	$!s$	`emit` s
	|	$s^{+}?(P)$	`present` s `then` P `end`
	|	$s^{-}?(P)$	`present` s `else` P `end`
	|	$P \mid P$	$P \parallel P$
	|	$P\ ;\ P$	$P\ ;\ P$
	|	$P \backslash s$	`signal` s `in` P `end`

Esterel more general choice statement:

`present` s `then` P_1 `else` P_2 `end`

can be recovered by the construct:

$s^{+}?(P_1) \mid s^{-}?(P_2).$

Formalising Mazes

Mazes are finite graphs with two types of directed edges, namely *visible* and *secret*.

These graphs are represented as *systems of unfolding rules*

$$M \ \coloneqq (x \Leftarrow m_x)_{x \in \mathrm{V}}$$

in a language of mazes, for some finite set of variables V representing *rooms* and *maze terms* m_x.

We write $m\{m'/x\}$ for the syntactic substitution that replaces all free occurrences of x by term m' in m.

Formalising Mazes

Maze terms are defined in a process algebraic fashion:

$$m \;\coloneqq 0 \mid x \mid \iota.m \mid \tau.m \mid \sum_{i \in I} m_i \;\mid \mu x.m$$

Intuitively, 0 is a *dungeon*, $\iota.m$ ($\tau.m$) represents a room with a *visible* (*secret*) corridor to room m.

$\sum_{i \in I} m_i$ corresponds to a room that *merges* all rooms m_i with $i \in I$ and we write $m_1 + m_2$ for $\sum_{i \in \{1,2\}} m_i$.

If $x \Leftarrow m_x$ is the *unfolding rule* defining room x then x corresponds to the *term* m_x and $\mu x.m_x$ is the *least fixed-point* solution for x.

Formalising Mazes

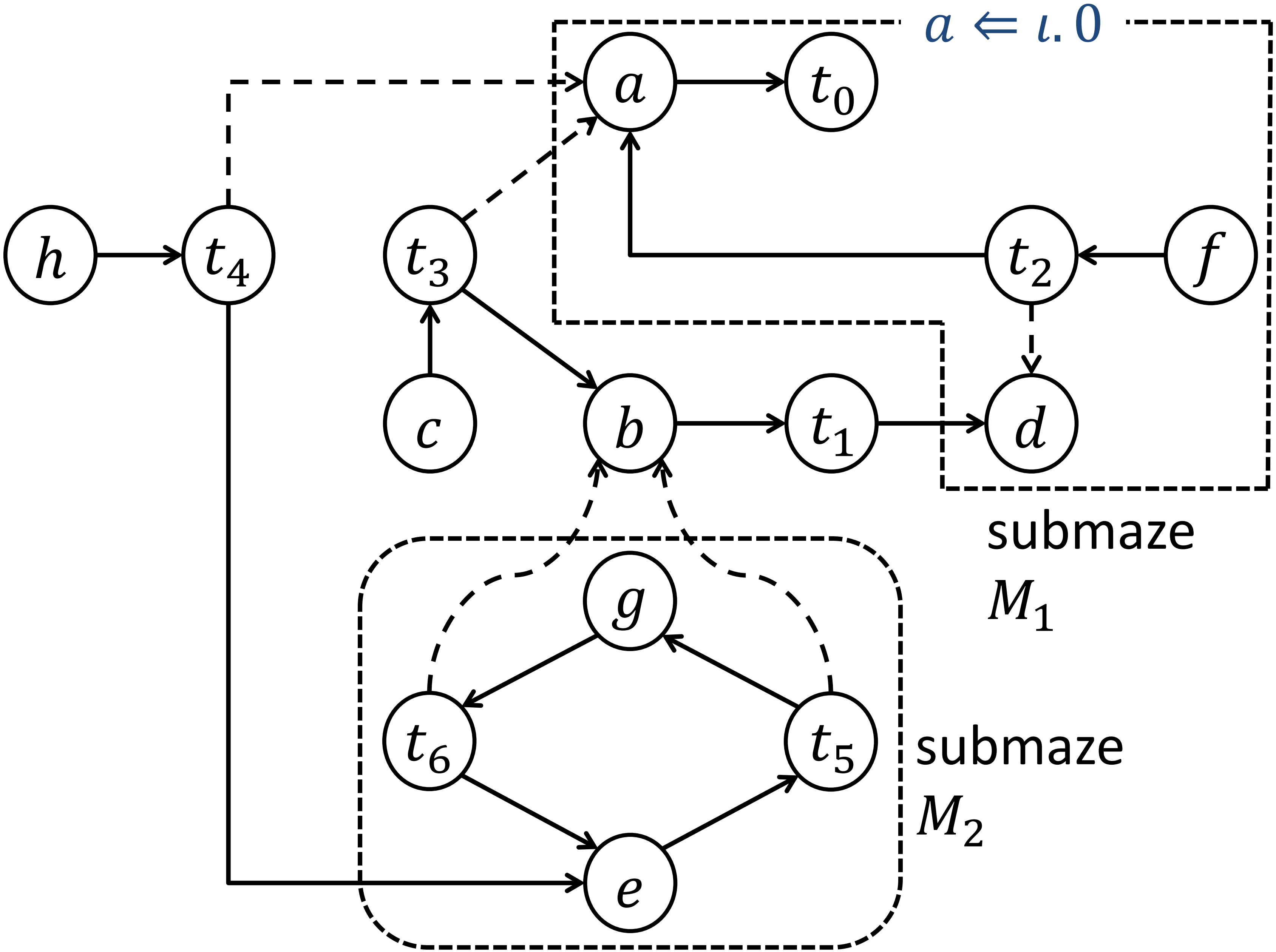
The game-theoretic semantics of maze M requires the introduction of a labelled transition system $\langle \mathcal{M}, \{\iota, \tau\}, \longrightarrow \rangle$ where $\mathcal{M}$ is the set of *rooms*, $\{\iota, \tau\}$ is the *alphabet* and $\longrightarrow$ is the *transition relation* representing corridors defined by:

$$\frac{}{\gamma . m \xrightarrow{\gamma} m} \qquad \frac{m_j \xrightarrow{\gamma} m'_j}{\sum_{i \in \mathrm{I}} m_i \xrightarrow{\gamma} m'_j}\; j \in I$$

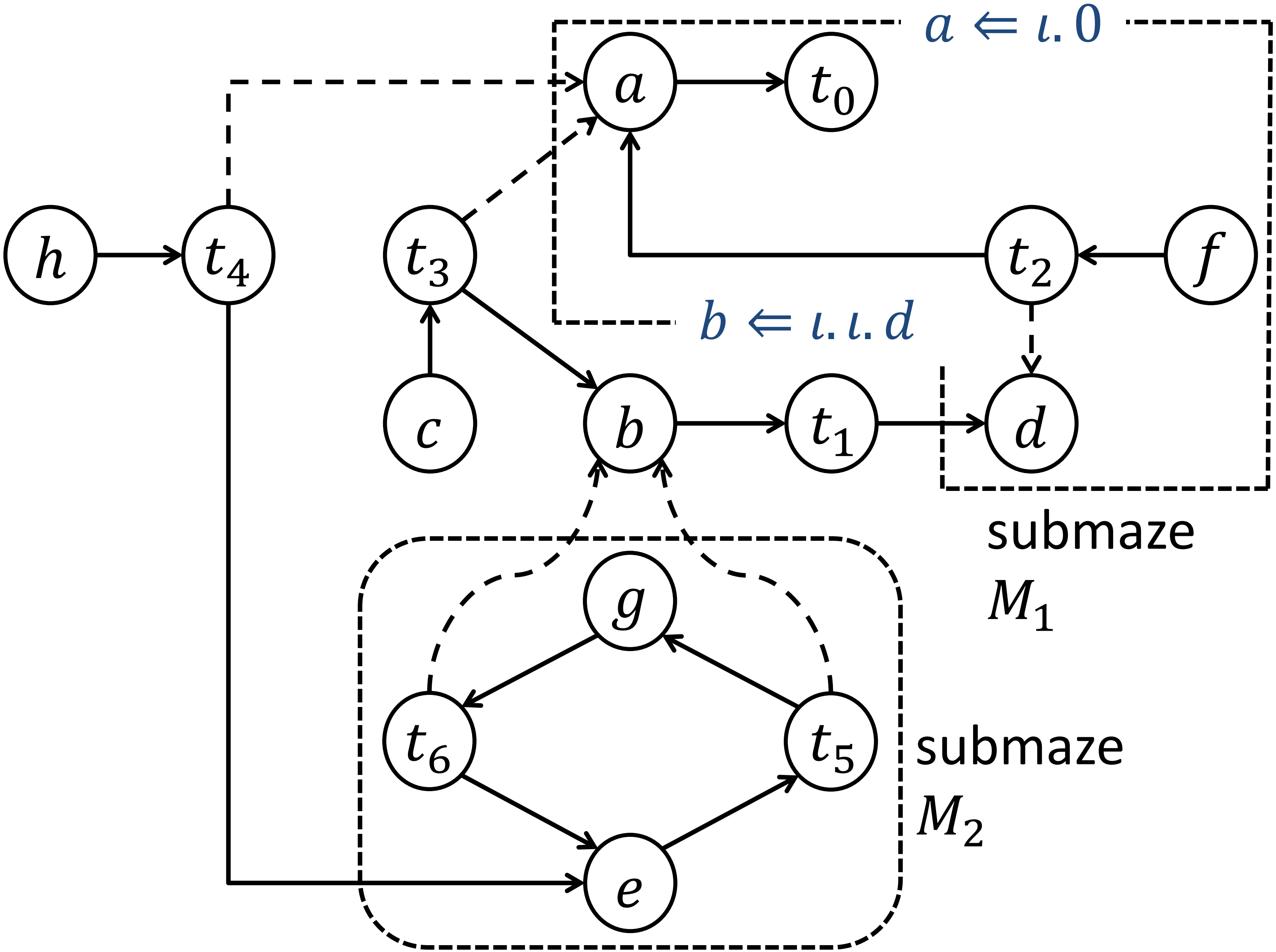
$$\frac{m \xrightarrow{\gamma} m'}{x \xrightarrow{\gamma} m'}\; x \Leftarrow m \qquad \frac{m\{\mu x . m / x\} \xrightarrow{\gamma} m'}{\mu x . m \xrightarrow{\gamma} m'}$$

where γ ranges over $\{\iota, \tau\}$.

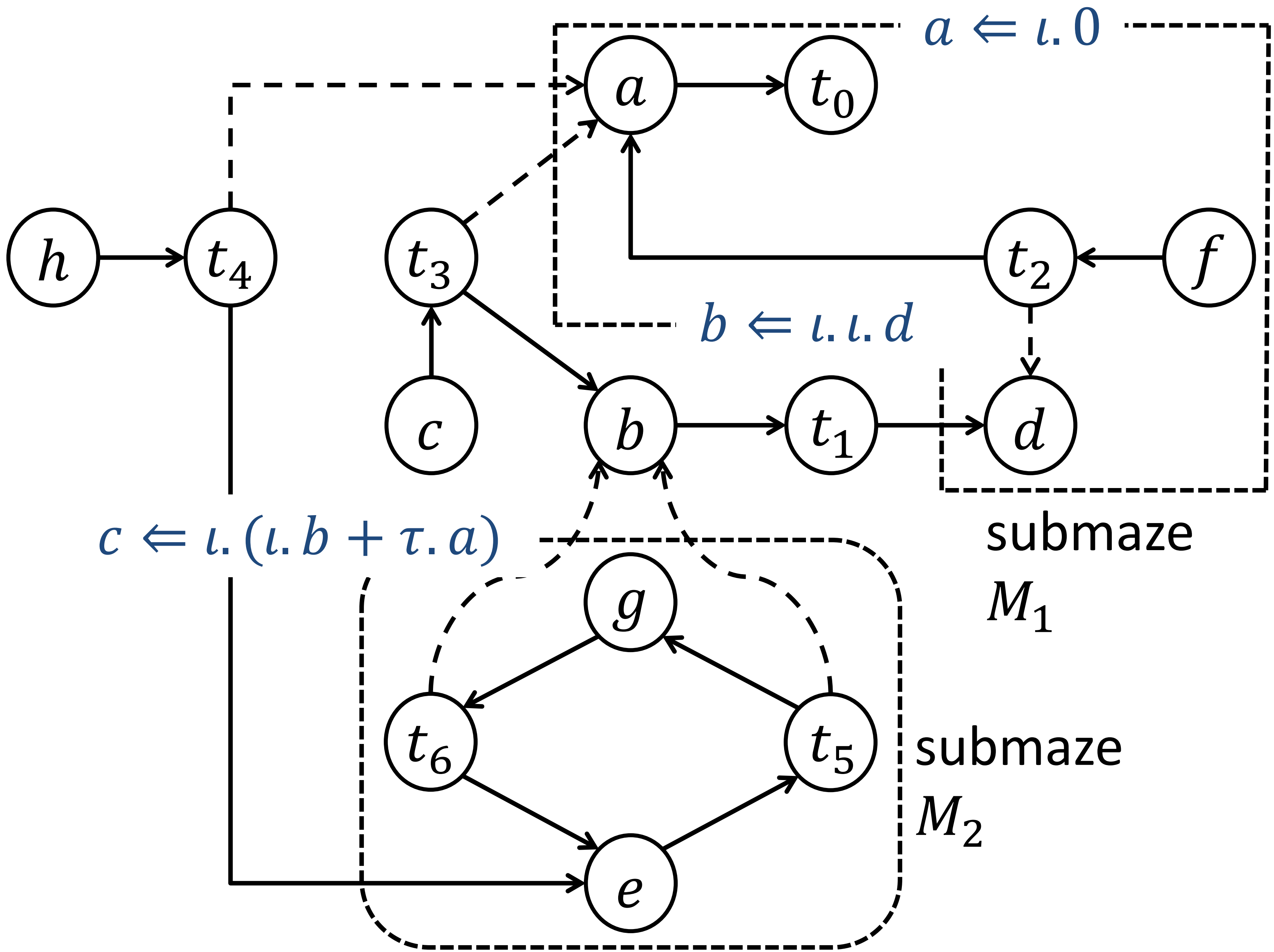
Formalising Mazes



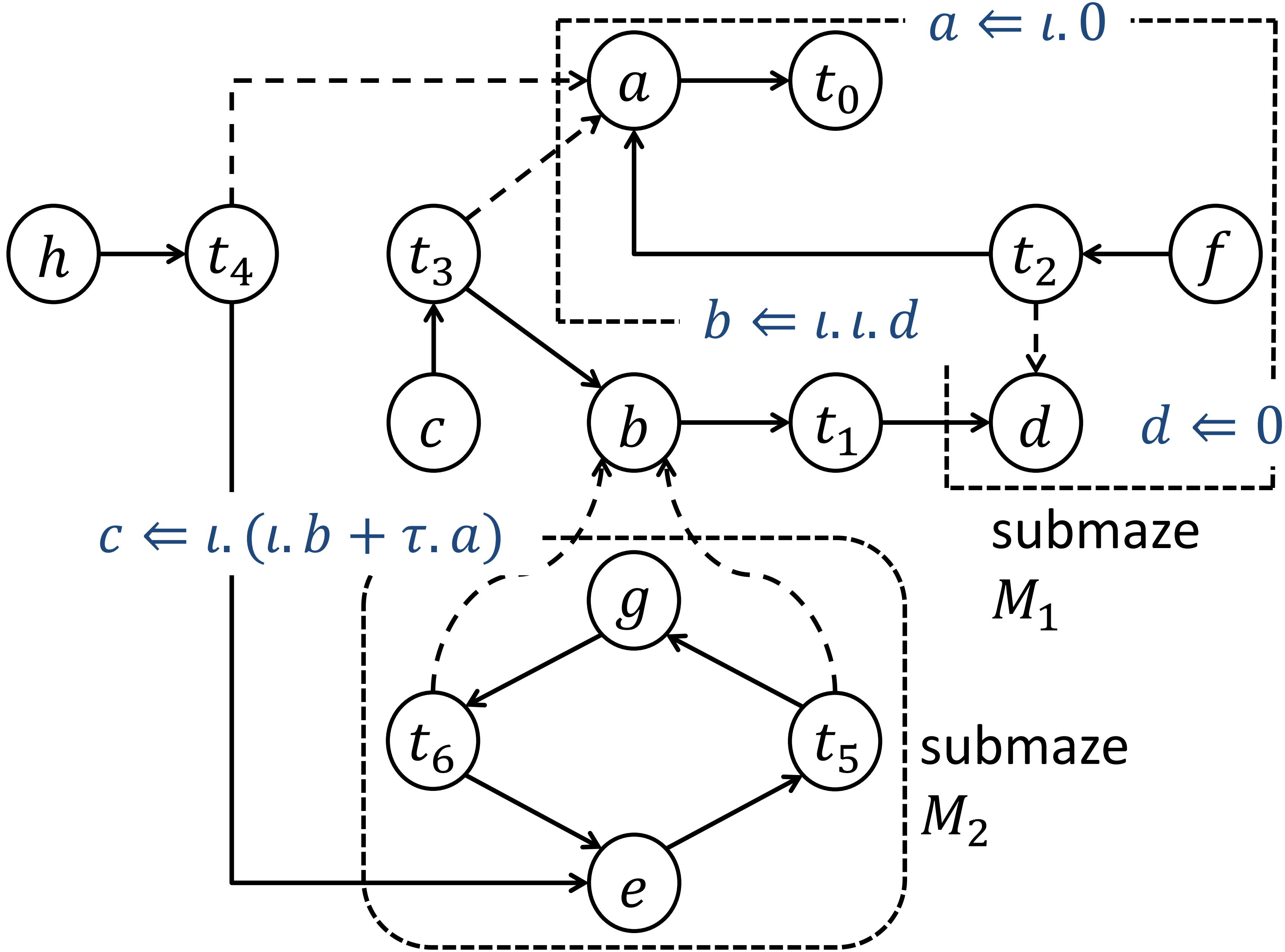
Formalising Mazes



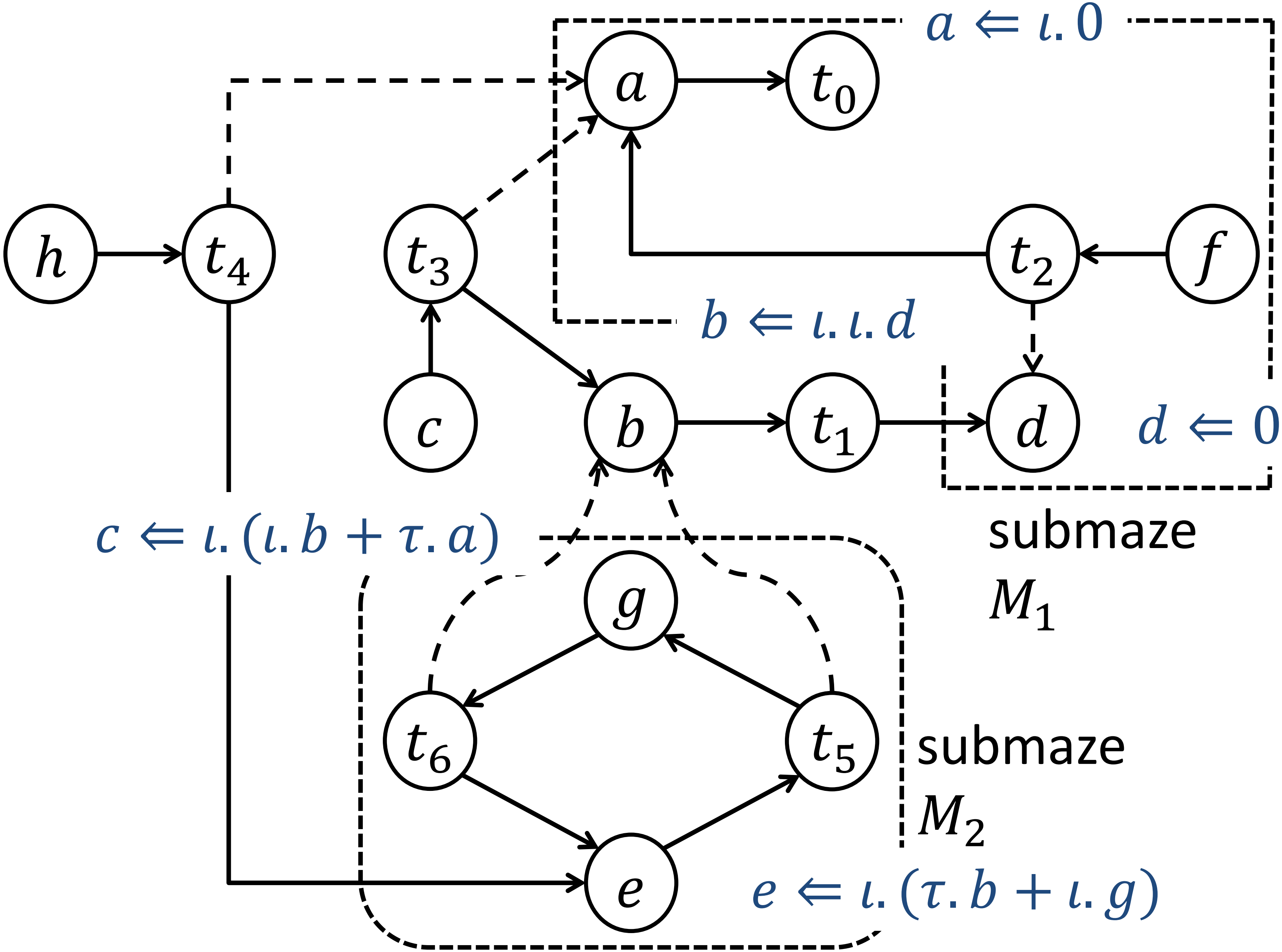
Formalising Mazes



Formalising Mazes



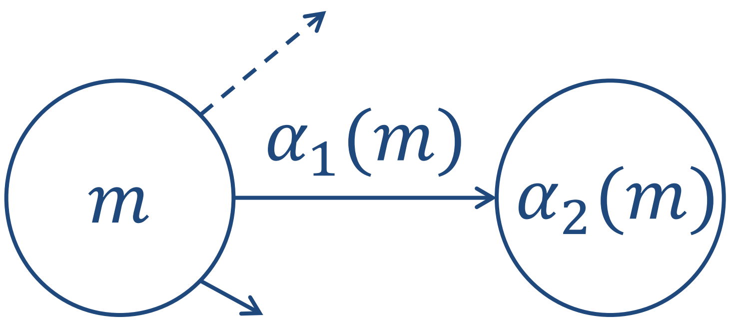
Formalising Mazes



Game Semantics

A maze play is determined by the players' strategies.

A strategy is a (partial) function: $\alpha : \mathcal{M} \rightarrow \{\iota, \tau\} \times \mathcal{M}$ such that , for all $m \in \mathcal{M}$ if $\alpha(m) = (\alpha_1(m), \alpha_2(m))$ is defined then



A strategy does not depend on the opponent's strategy or on a play history.

Game Semantics

Given strategies α and β for players P and O, the play $play(\alpha, \beta, m)$ is the maximal path in M starting in room m with player P.

A player has a *winning strategy*, if he is always able to drive his opponent into a dungeon no matter which strategy his opponent employs and always assuming that P starts the game.

If player P has a winning strategy for room x, then x is a *winning position*. If player O has a winning strategy, then x is a *losing position*.

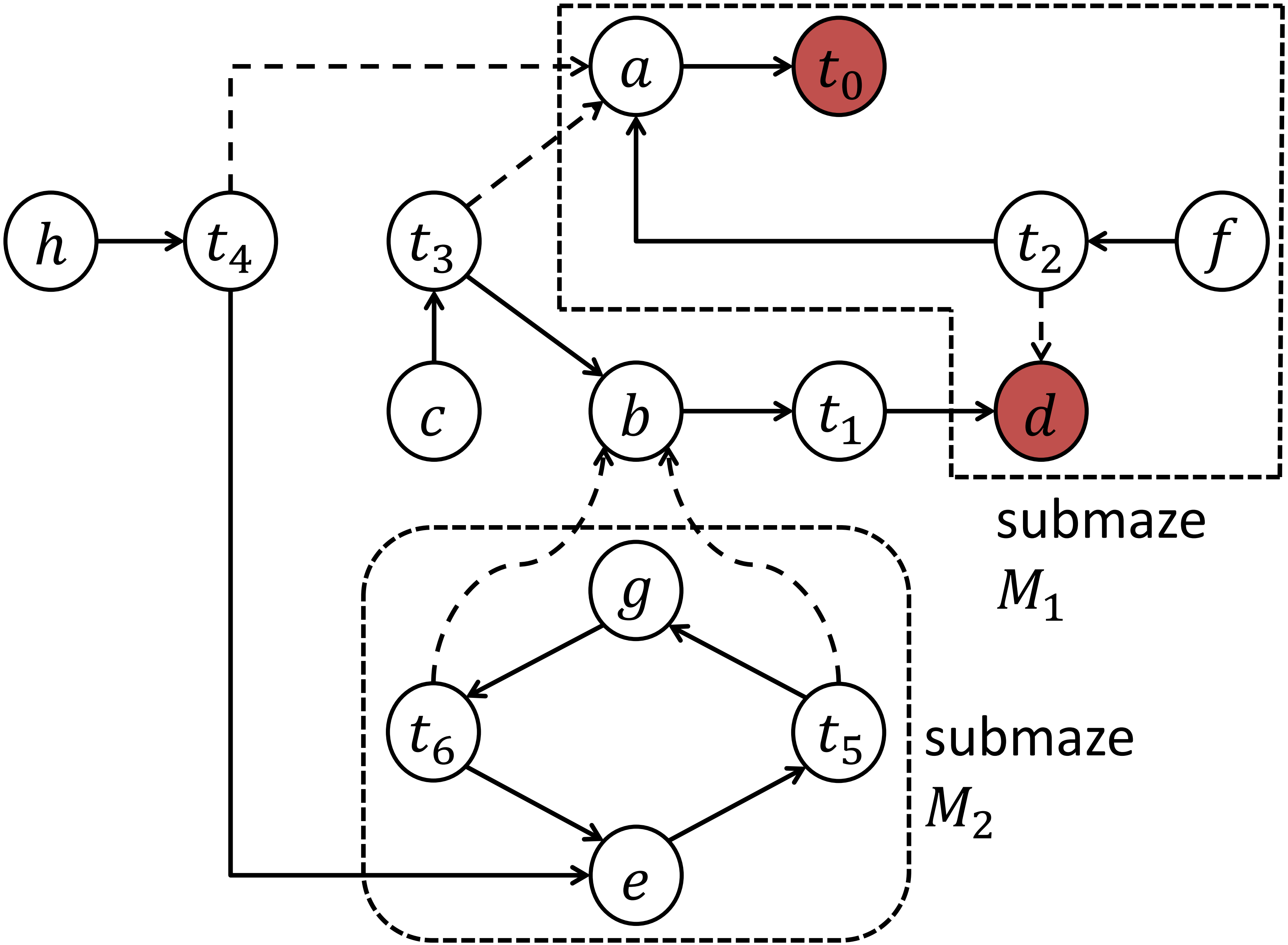
Game Semantics

If both players can always avoid dungeons, thus engaging in infinite plays, neither player wins and the play ends in a *draw*.

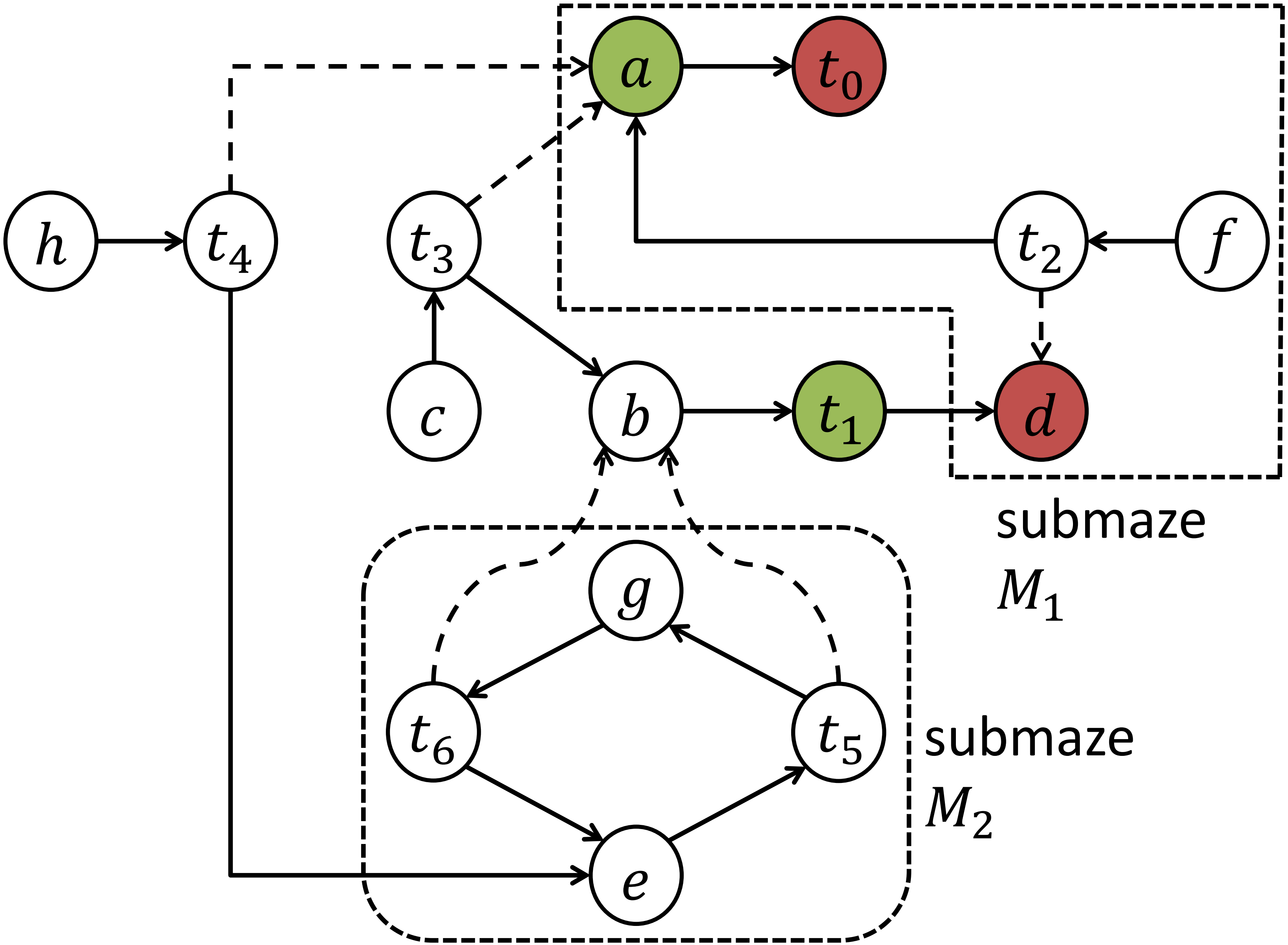
A position that is neither a wining or a losing position is referred to as a *draw position*.

Technically, strategies within a maze M correspond to the must- and cannot-analysis (Esterel) of the associated program, which forms the basis of Esterel causality analysis.

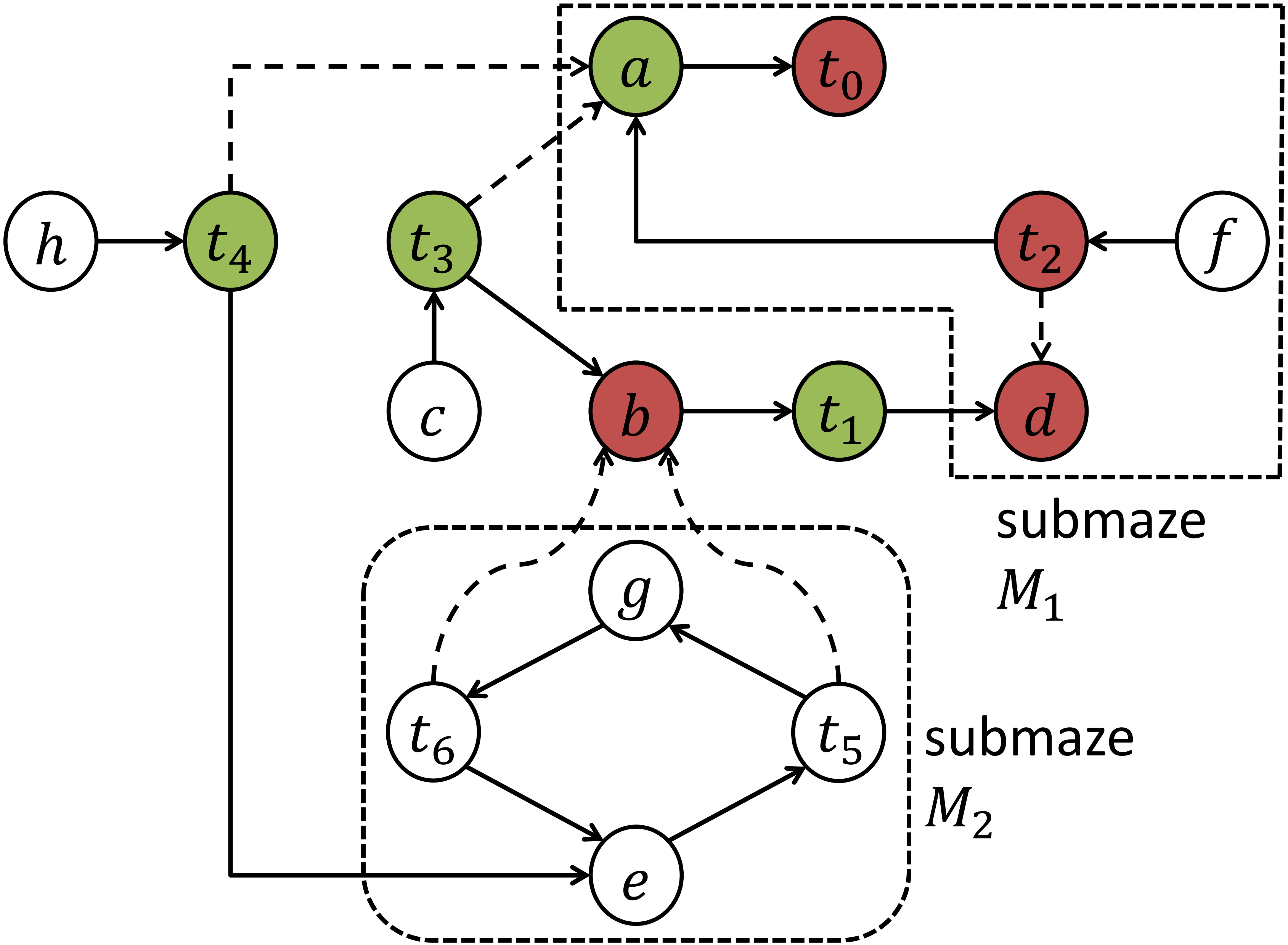
Game Semantics



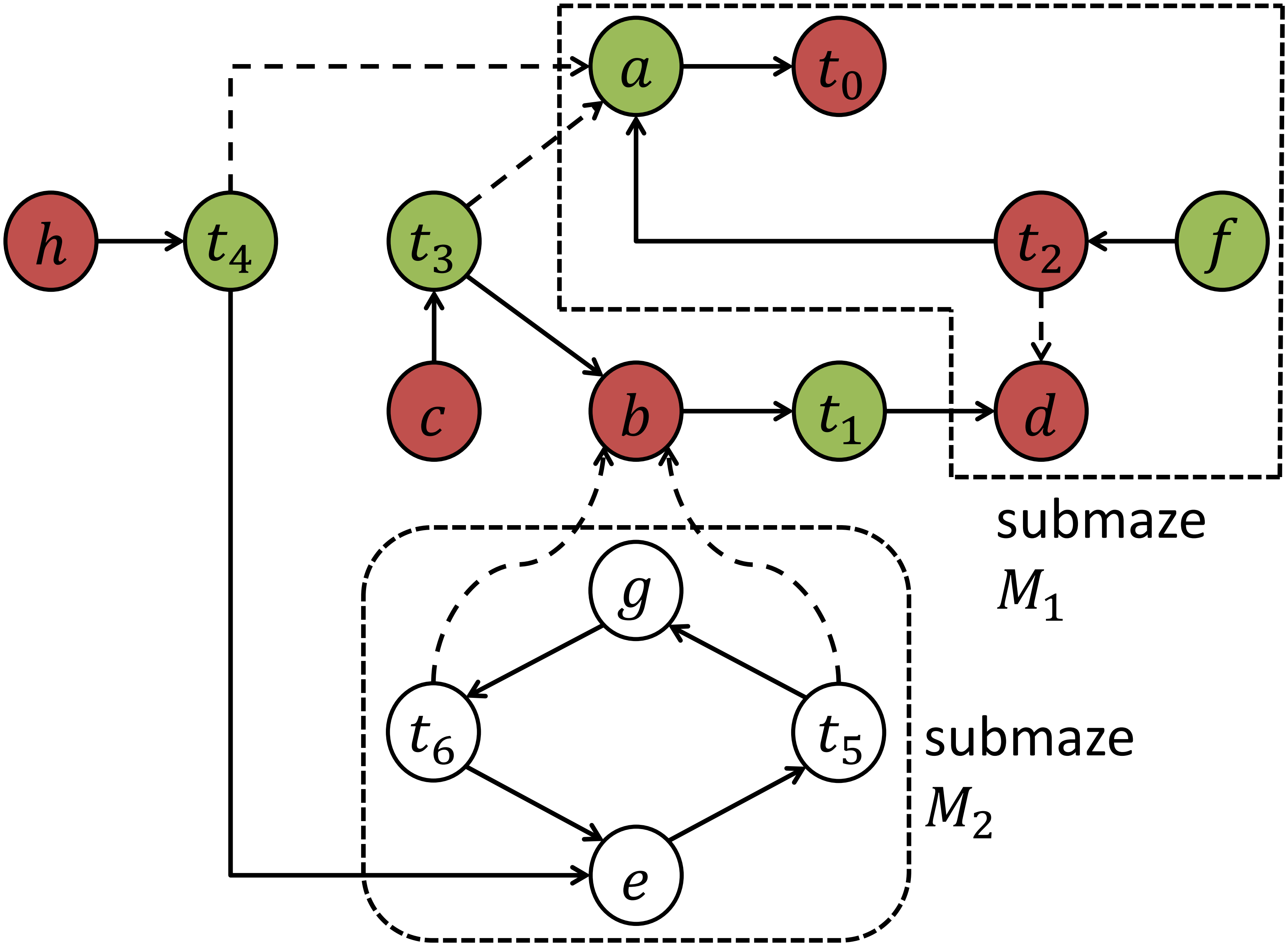
Game Semantics



Game Semantics



Game Semantics



Representing Programs as Mazes

With each program P, we associate a maze:

$$M \coloneqq (a \Leftarrow \langle\langle P \rangle\rangle_a \{0/\delta\})_{a \in S \cup \{\lambda\}}$$

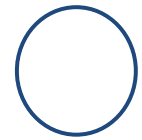
The elements in $S \cup \{\lambda, \delta\}$ play the role of term variables representing rooms.

There is no rule $\delta \Leftarrow \langle\langle P \rangle\rangle_\delta$ for the connecting variable δ.

In general, $\langle\langle P \rangle\rangle_a$ with describes the game conforming to P that can be played starting in room a modulo some conditions (dependencies) yet to be defined where instances of δ appear.

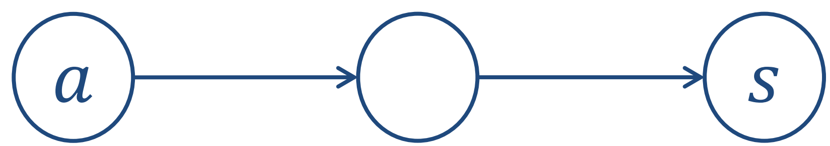
Representing Programs as Mazes

$\langle\langle 0 \rangle\rangle$

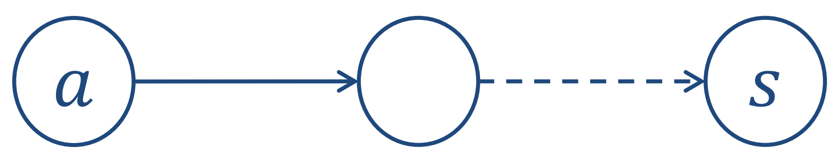


$\langle\langle !\, a \rangle\rangle$

$\langle\langle s^{+}?\, !\, a \rangle\rangle$

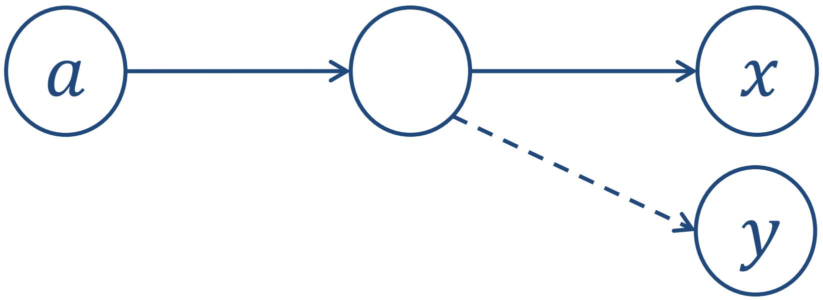


$\langle\langle s^{-}?\, !\, a \rangle\rangle$

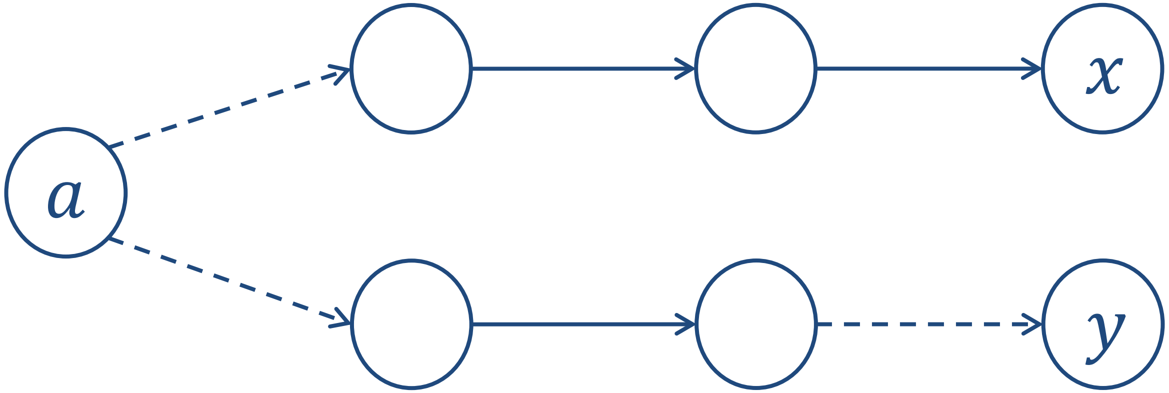


Representing Programs as Mazes

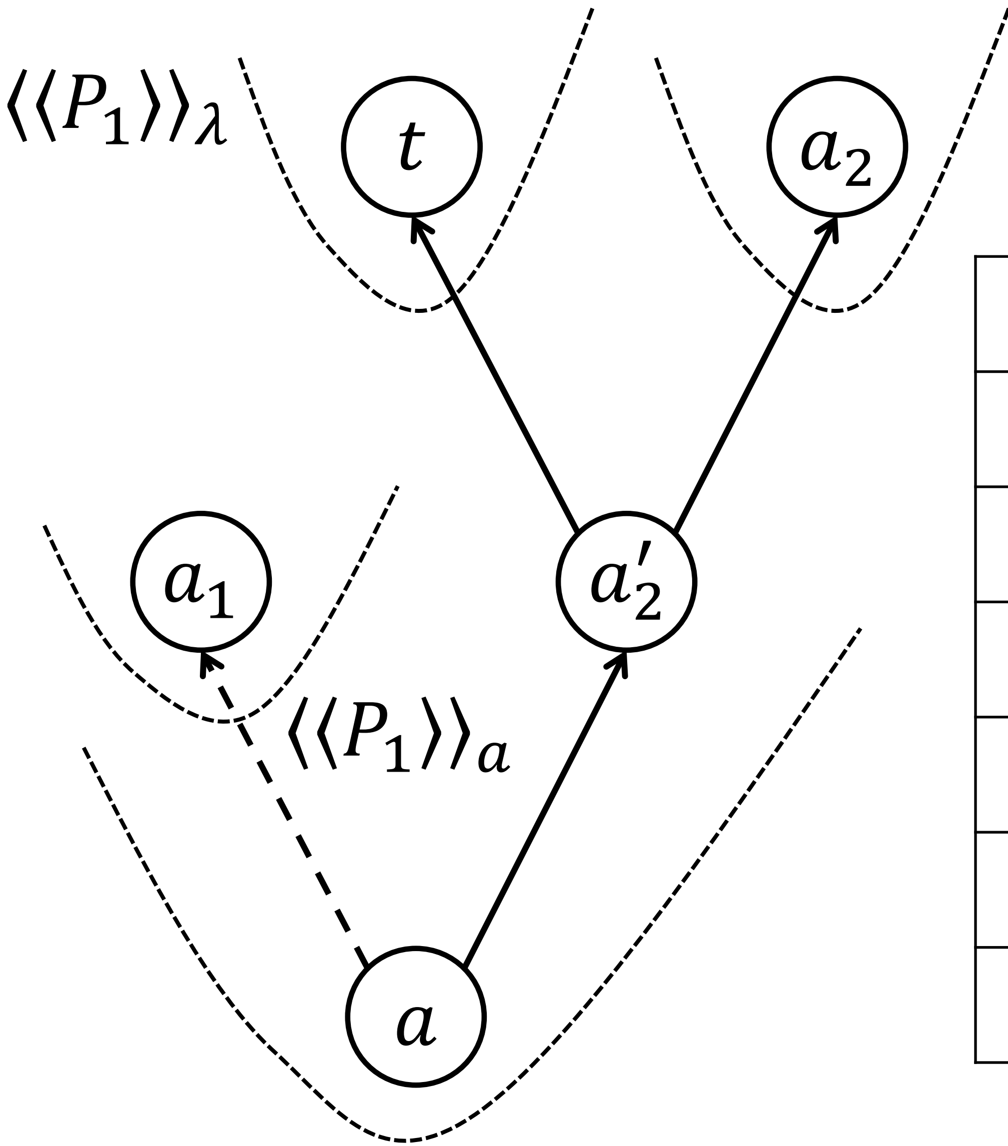
$\langle\langle x^{+}?\, y^{-}?\, !\, a \rangle\rangle$



$\langle\langle\, x^{+}?\, !\, a \mid y^{+}?\, !\, a \,\rangle\rangle$



Representing Programs as Mazes

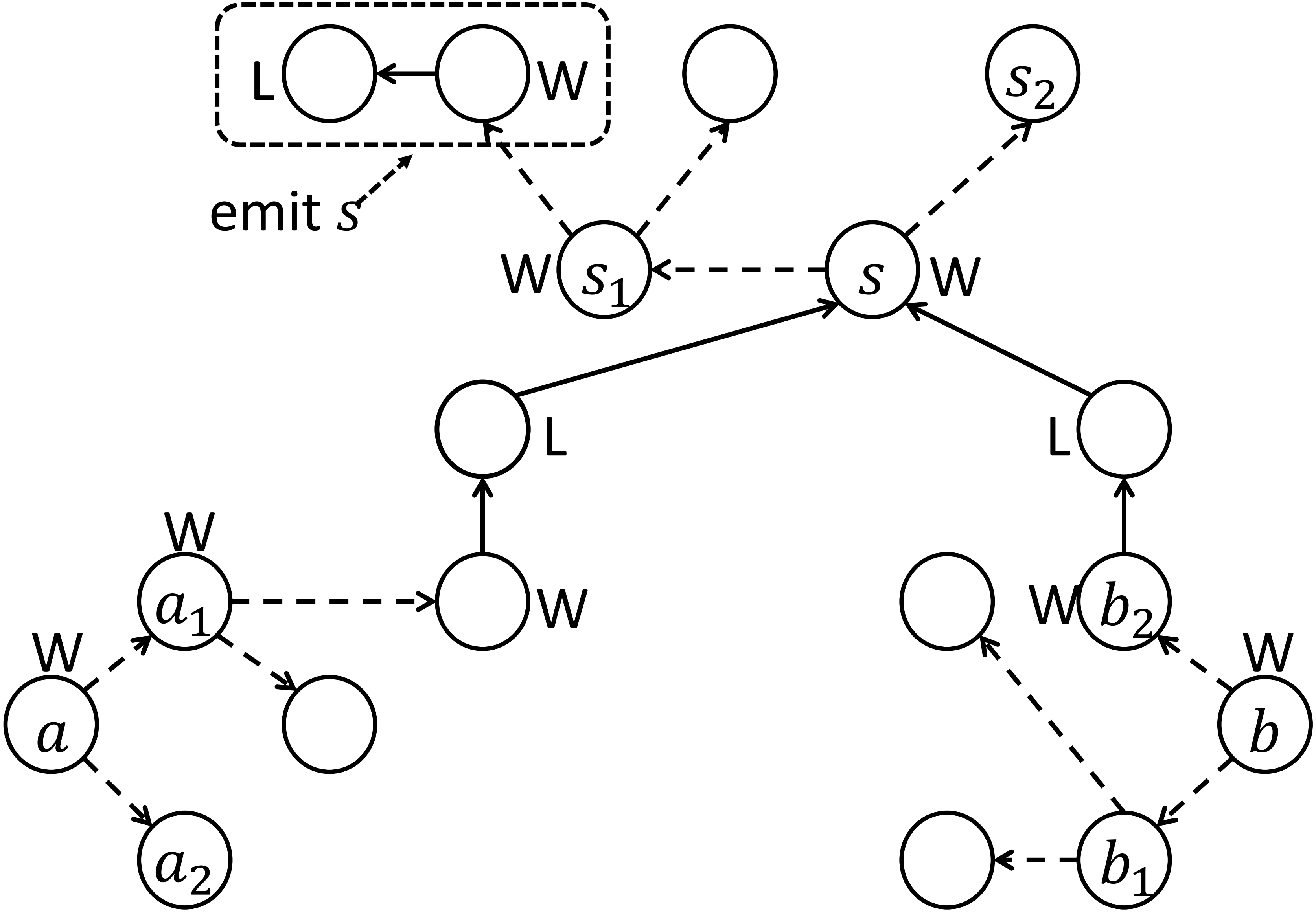


$\boldsymbol{a_1}$	$\boldsymbol{t}$	$\boldsymbol{a_2}$	$\boldsymbol{a}$
W	χ	χ	W
L	W	χ	χ
L	D	W	D
L	D	L	L
L	D	D	D
D	D	χ	D

$$\langle\langle P_1; P_2 \rangle\rangle_a \;\coloneqq\; \tau.\langle\langle P_1 \rangle\rangle_a + \iota.(\iota.\langle\langle P_1 \rangle\rangle_\lambda + \iota.\langle\langle P_2 \rangle\rangle_a)$$

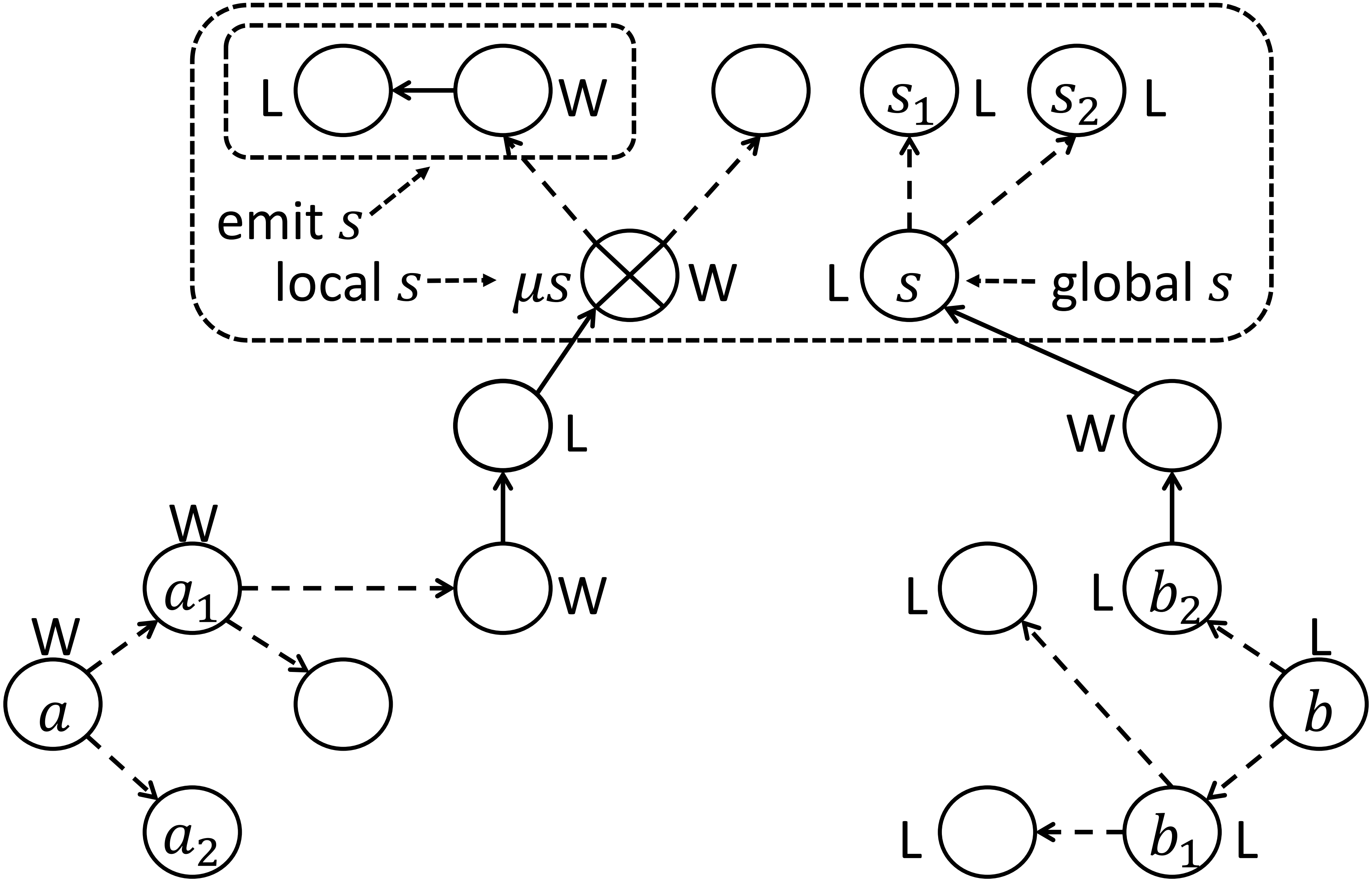
Representing Programs as Mazes

$\langle\!\langle\, (s^{+}?\,!\,a \mid !\,s) \setminus s \mid s^{+}?\,!\,b \,\rangle\!\rangle$



Representing Programs as Mazes

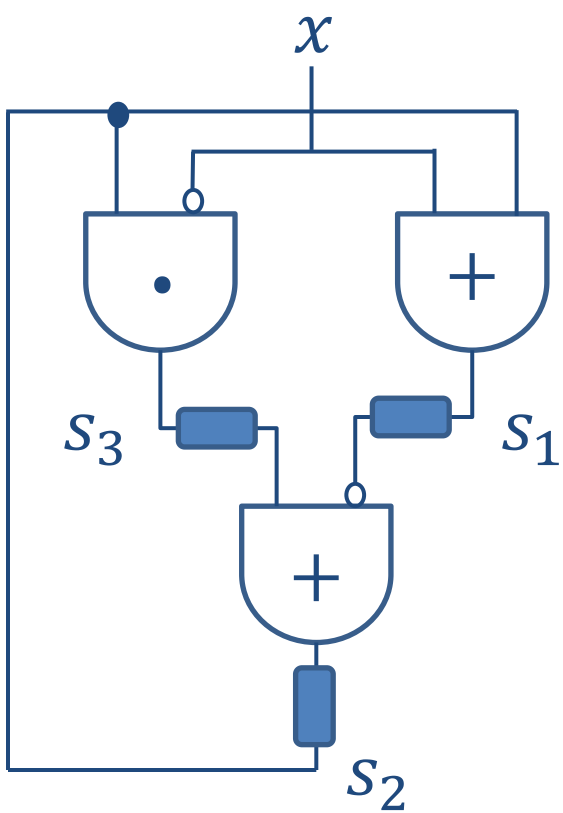
$\langle\!\langle\, (s^{+}?\,!\,a \mid !\,s) \setminus s \mid s^{+}?\,!\,b \,\rangle\!\rangle$

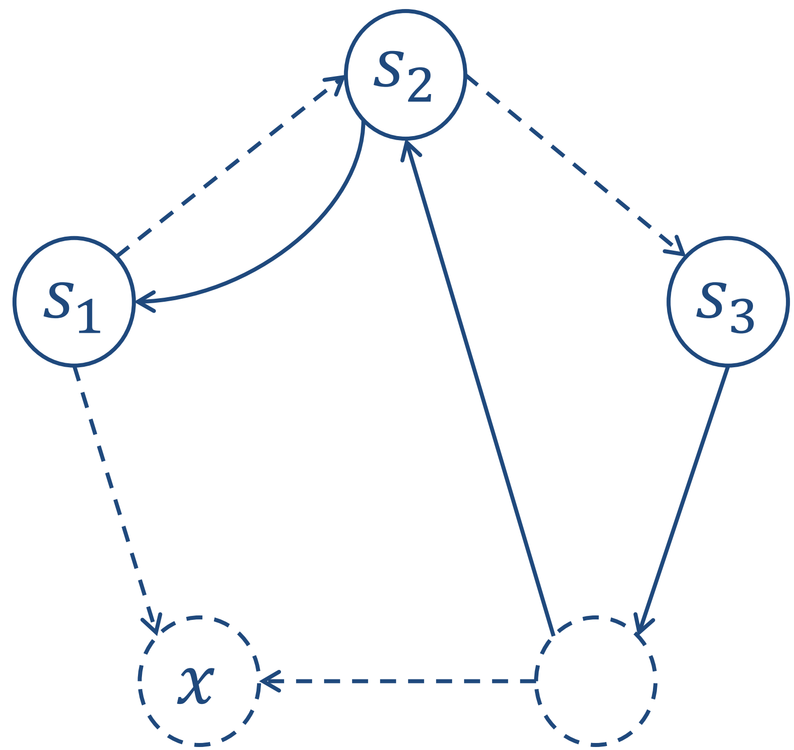


Non-combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?(!\,s_1) \mid x^+?(!\,s_1) \mid s_1^-?(!\,s_2) \mid$$
$$s_3^+?(!\,s_2) \mid x^-?\,s_2^+?(!\,s_3)$$

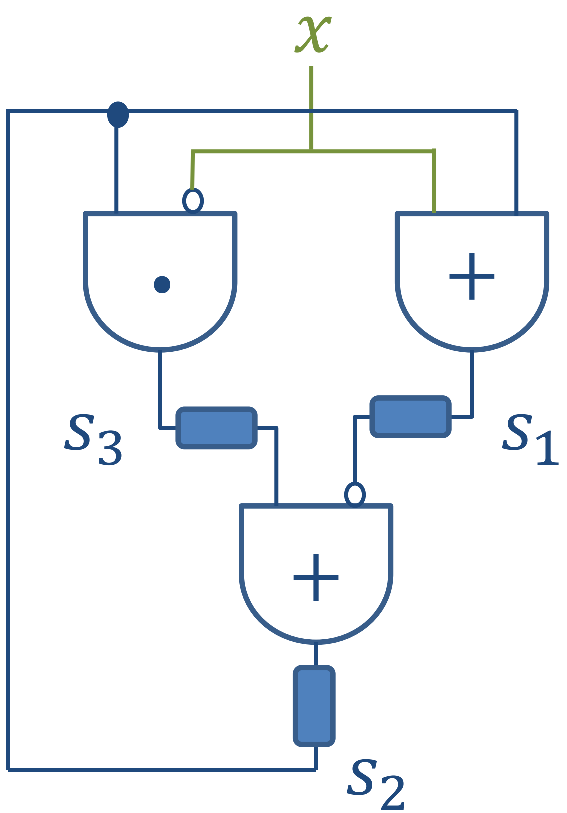


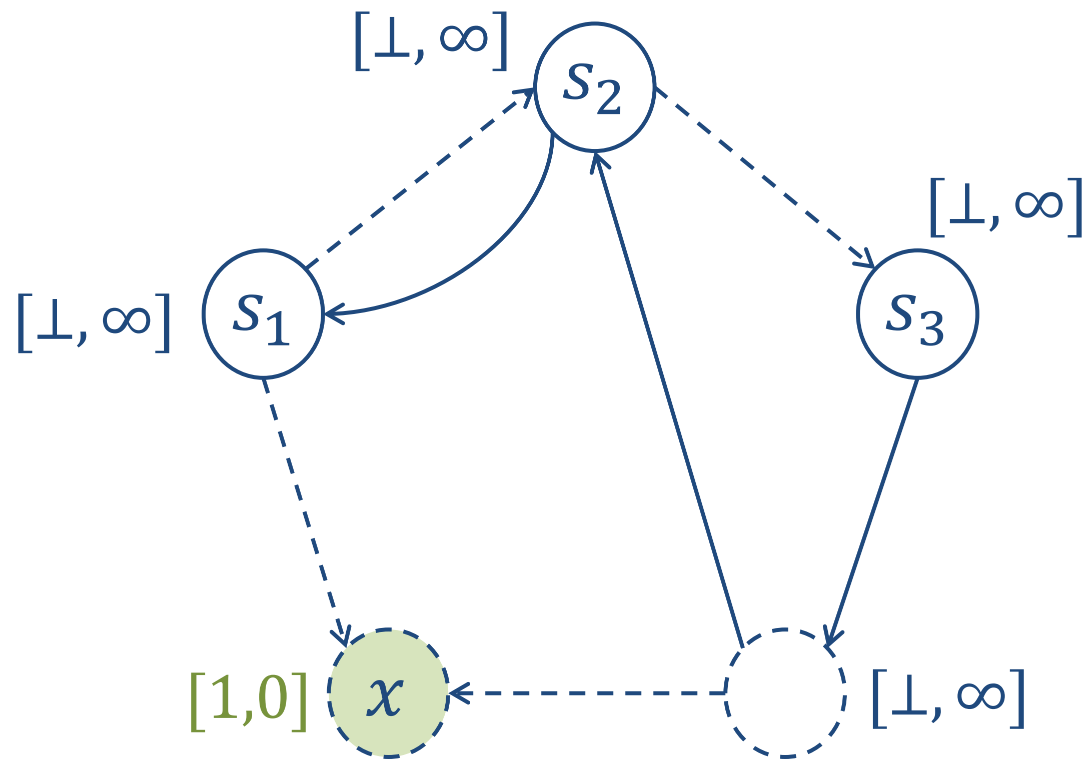


Non-combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid$$
$$s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

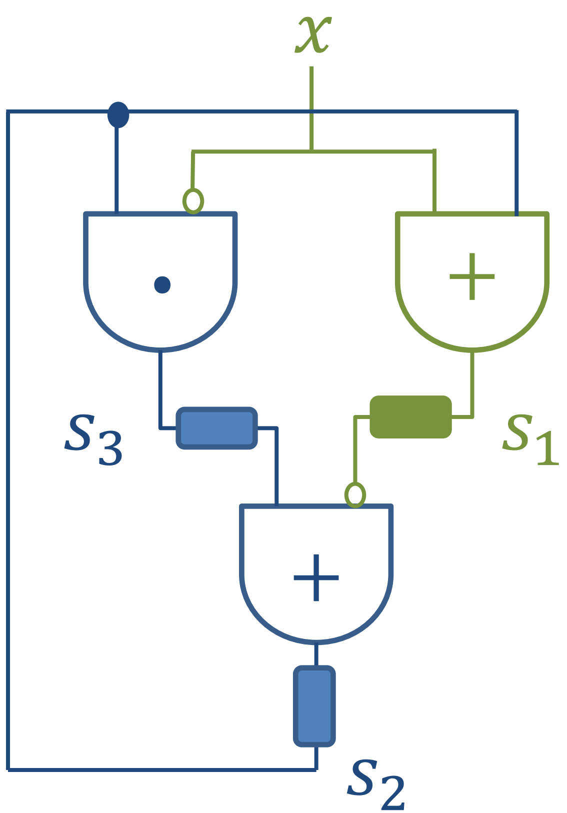


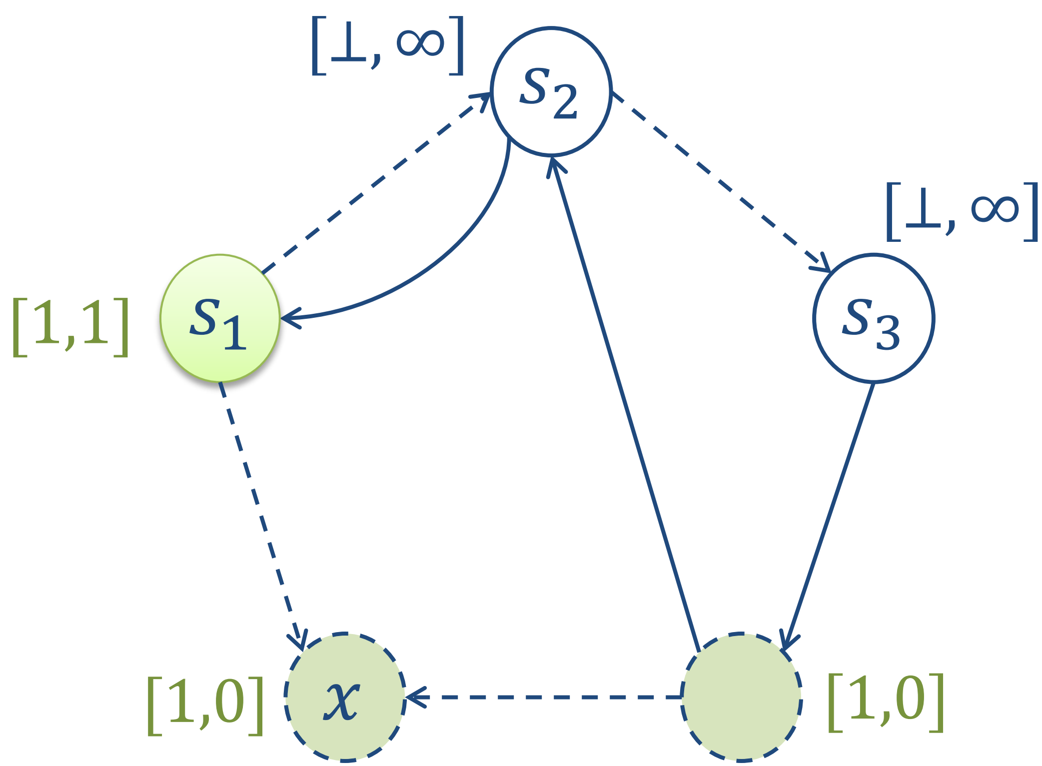


Non-combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+ ?\,(!\,s_1) \mid x^+ ?\,(!\,s_1) \mid s_1^- ?\,(!\,s_2) \mid s_3^+ ?\,(!\,s_2) \mid x^- ?\, s_2^+ ?\,(!\,s_3)$$

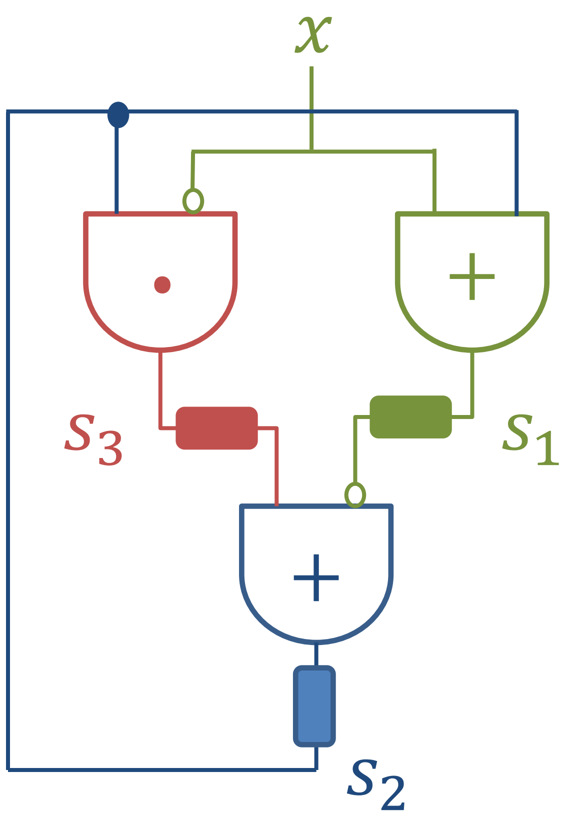


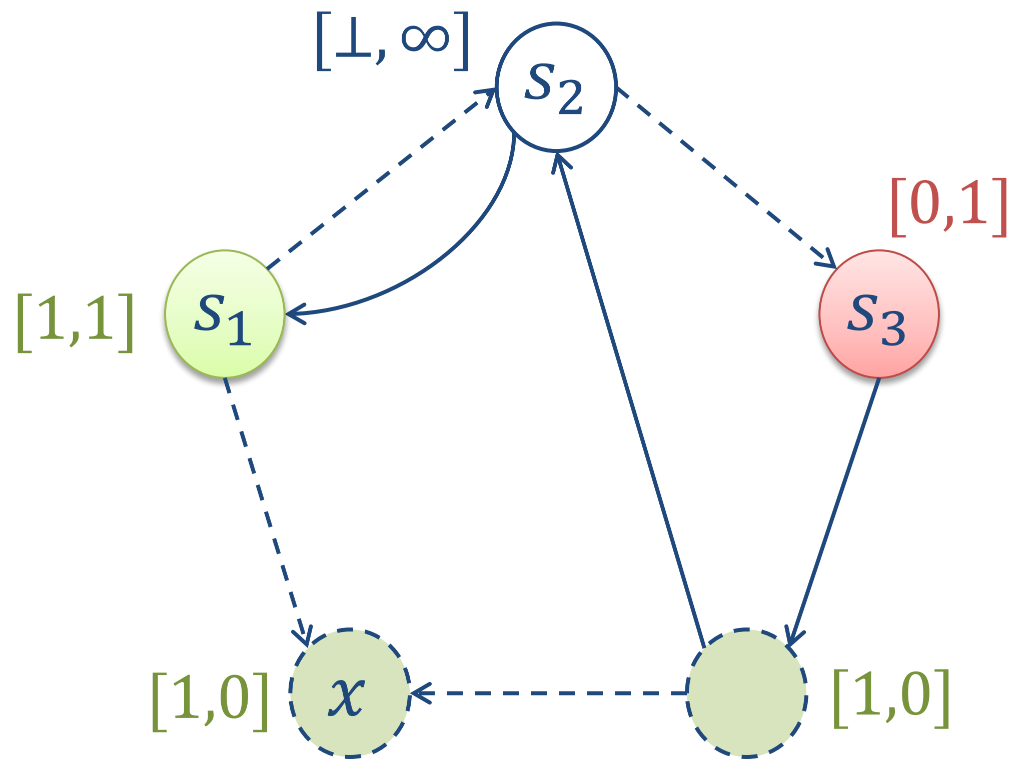


Non-combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

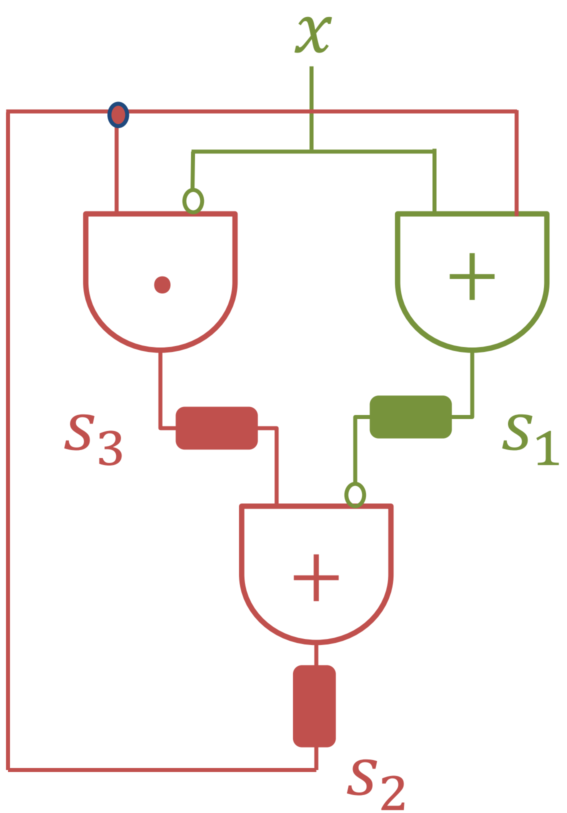


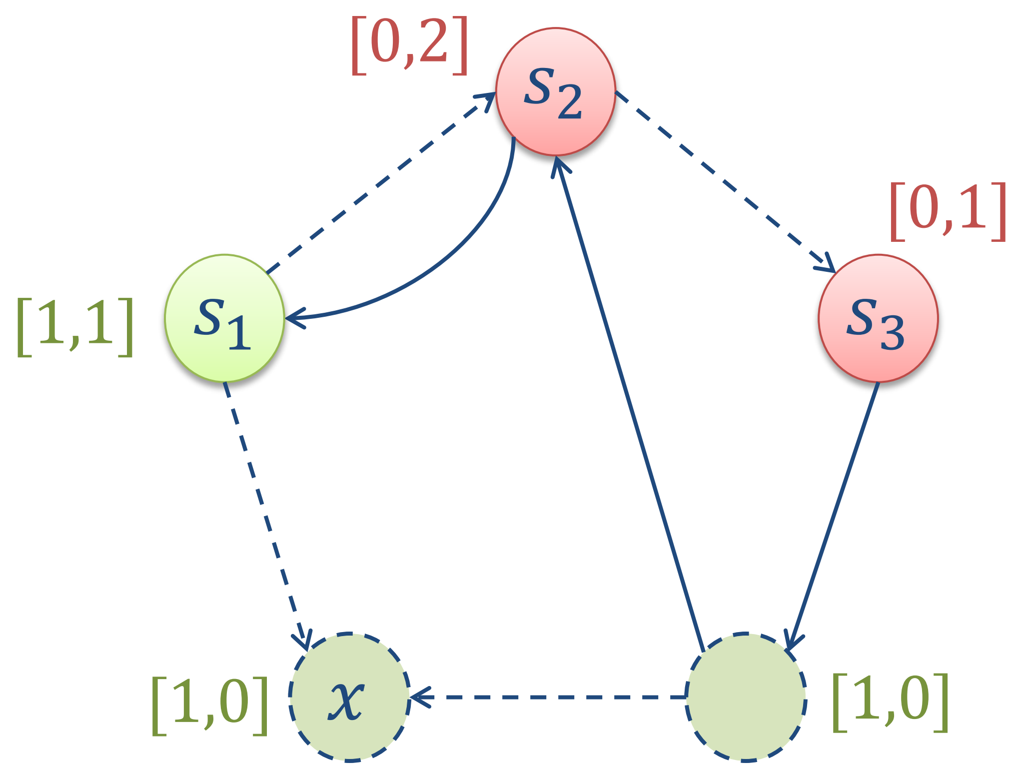


Non-combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+ ?\,(!\,s_1) \mid x^+ ?\,(!\,s_1) \mid s_1^- ?\,(!\,s_2) \mid s_3^+ ?\,(!\,s_2) \mid x^- ?\, s_2^+ ?\,(!\,s_3)$$

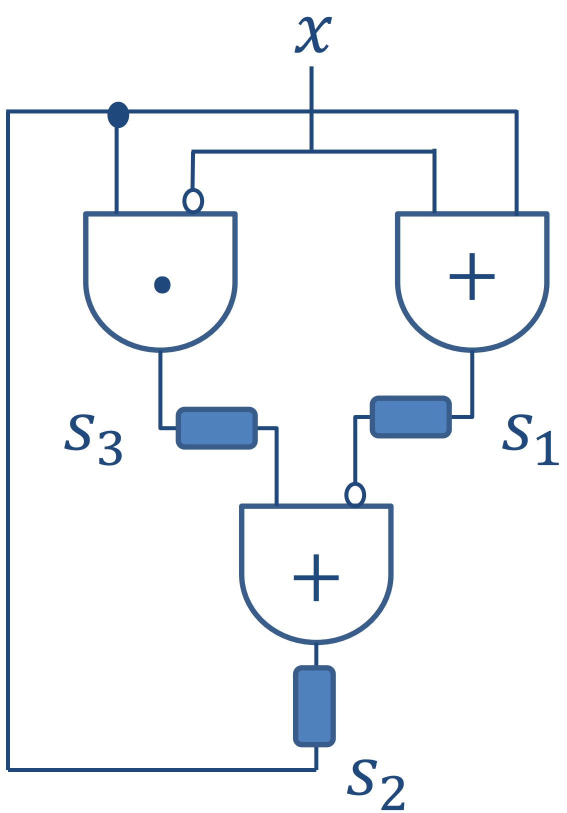


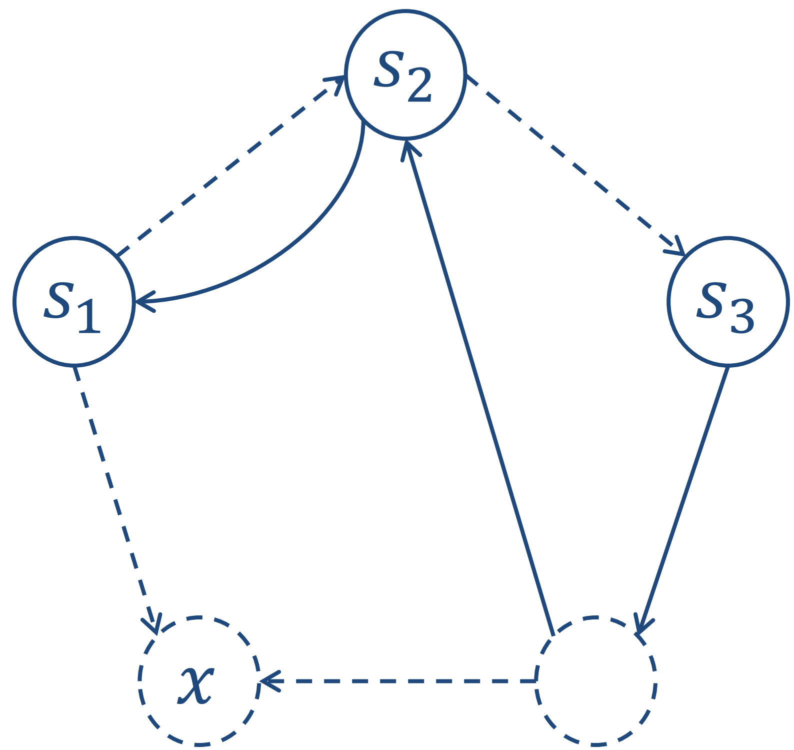


Non-combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

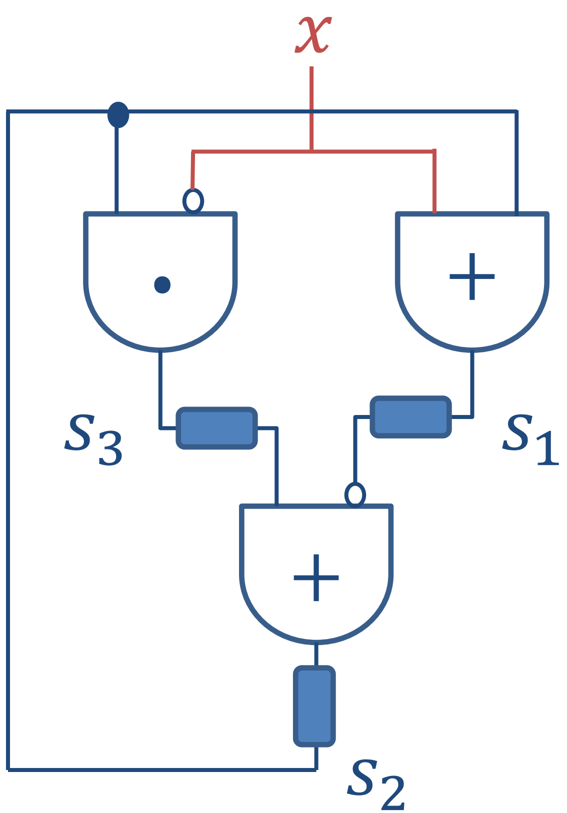


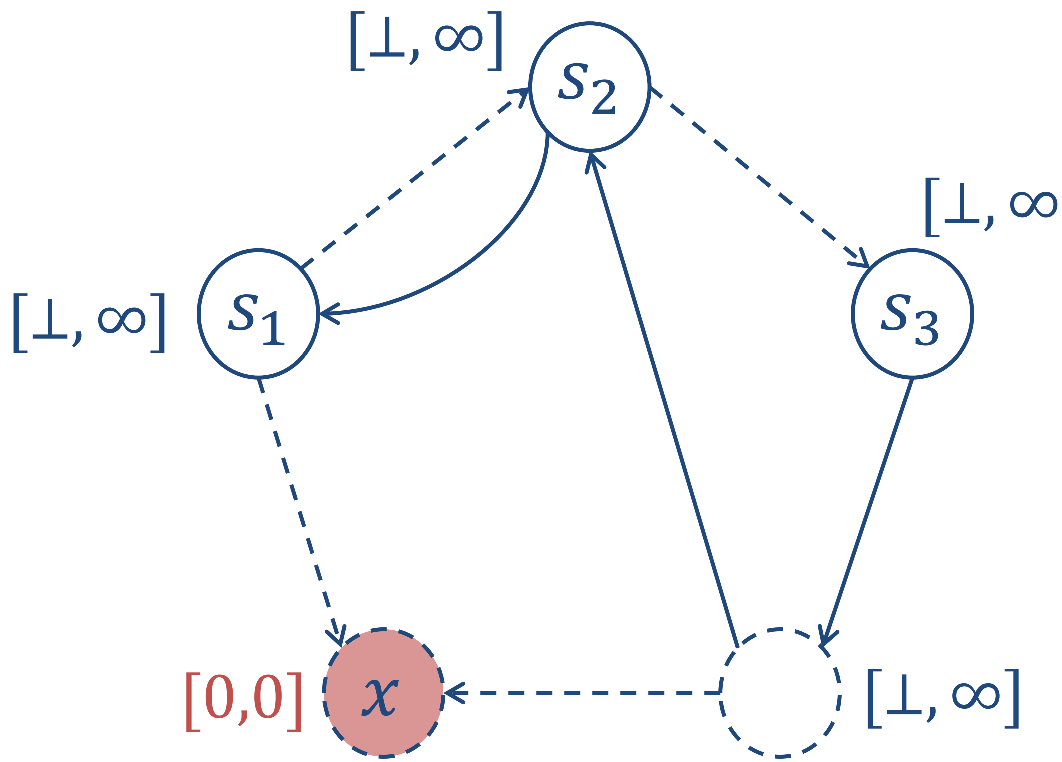


Non-combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+ ? (! s_1) \mid x^+ ? (! s_1) \mid s_1^- ? (! s_2) \mid s_3^+ ? (! s_2) \mid x^- ? s_2^+ ? (! s_3)$$

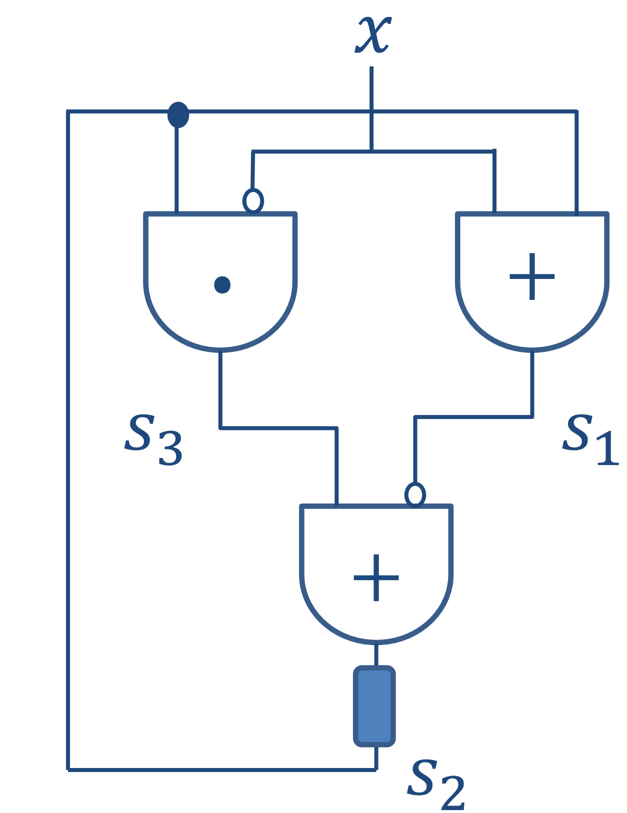


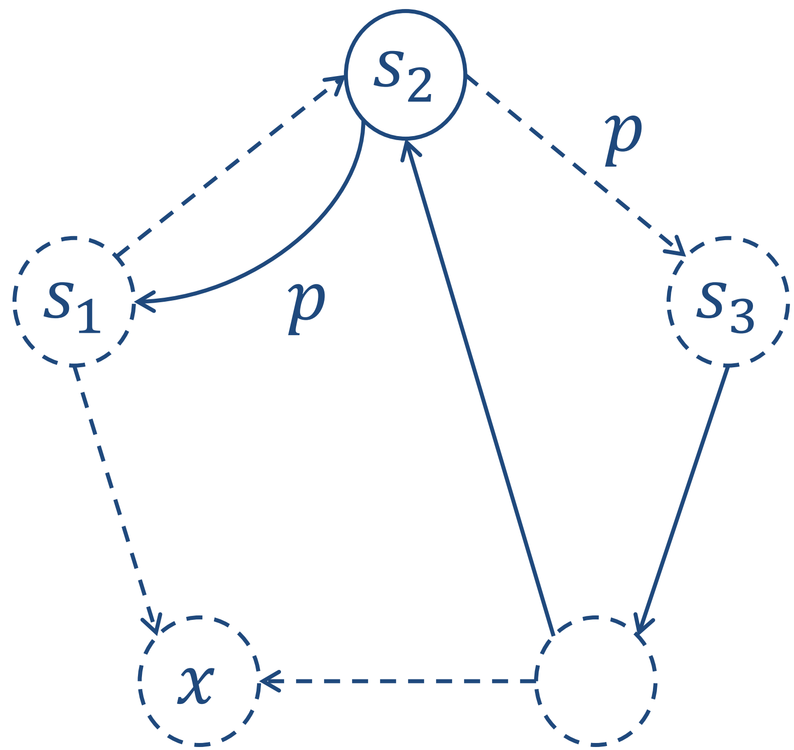


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$



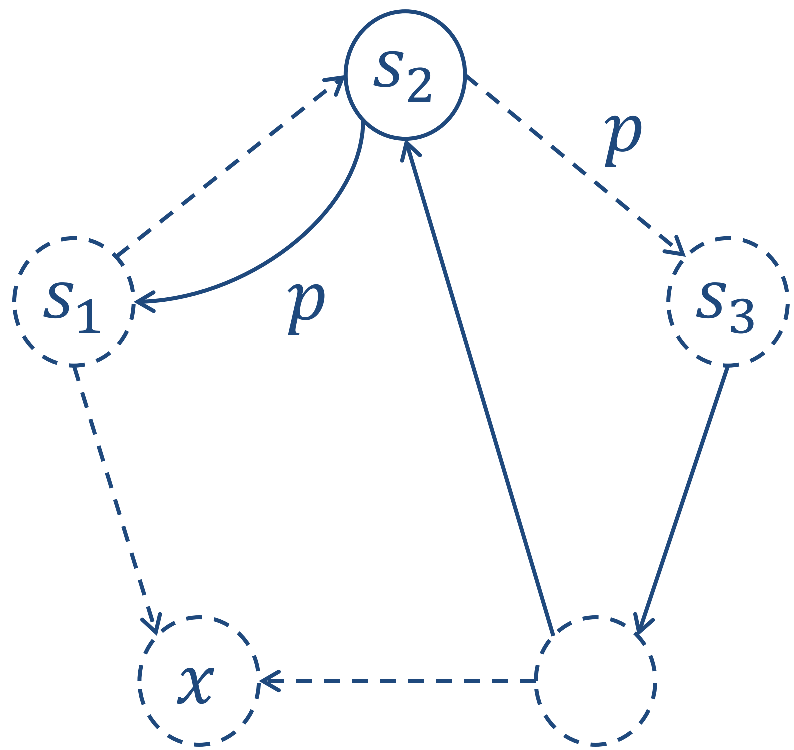


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

$$s_2 = \overline{(s_2 + x)} + (\overline{x} \cdot s_2)$$
$$s_2 = (\overline{s_2} \cdot \overline{x}) + (\overline{x} \cdot s_2)$$
$$s_2 = \overline{x} \cdot (\overline{s_2} + s_2) = \overline{x}$$

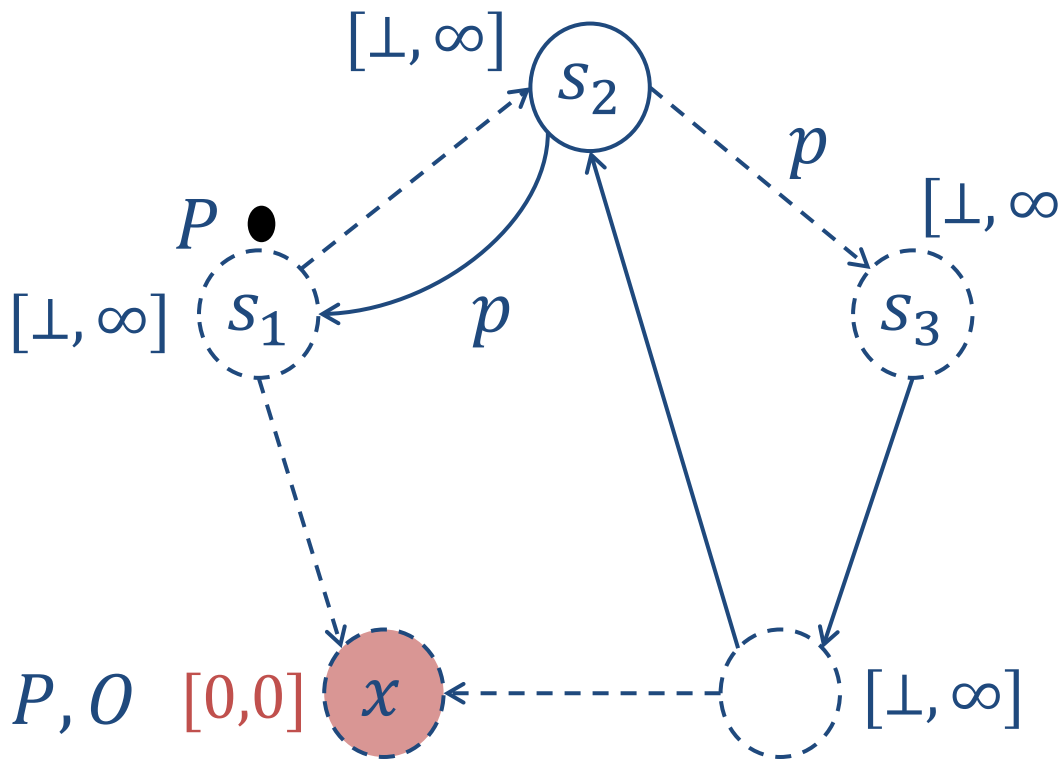


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid$$
$$s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

$$s_2 = \overline{x}$$

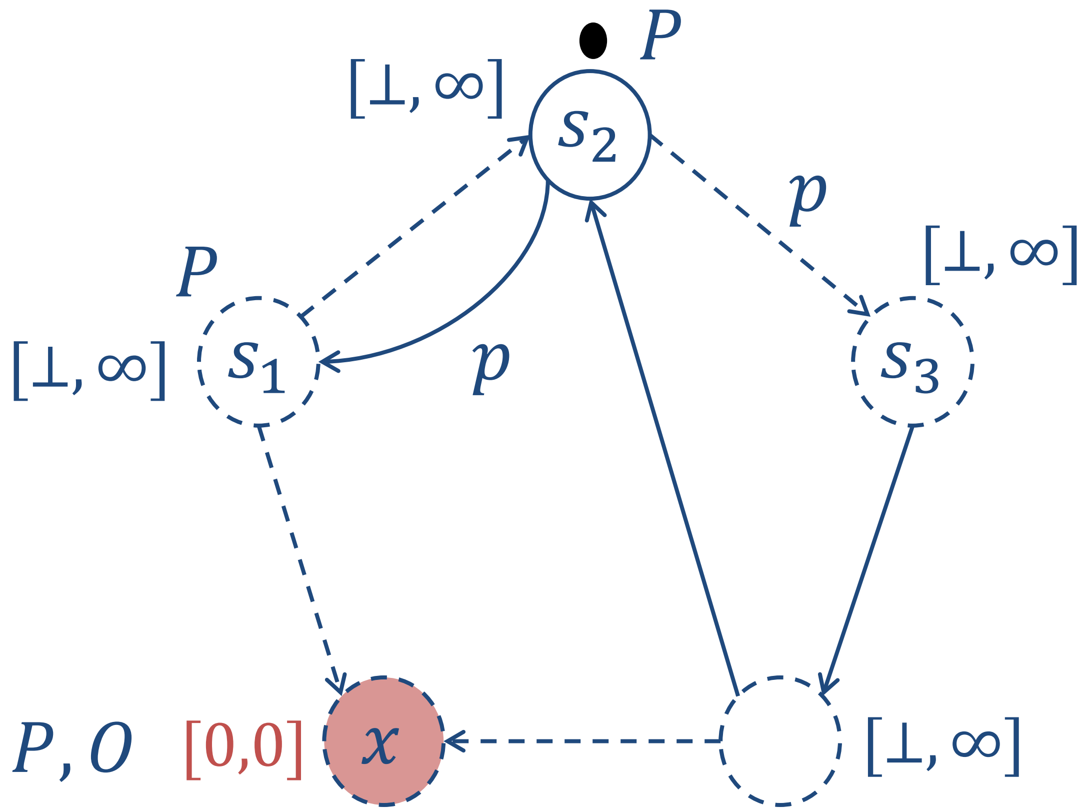


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?(!\, s_1) \mid x^+?(!\, s_1) \mid s_1^-?(!\, s_2) \mid s_3^+?(!\, s_2) \mid x^-?\, s_2^+?(!\, s_3)$$

$$s_2 = \overline{x}$$

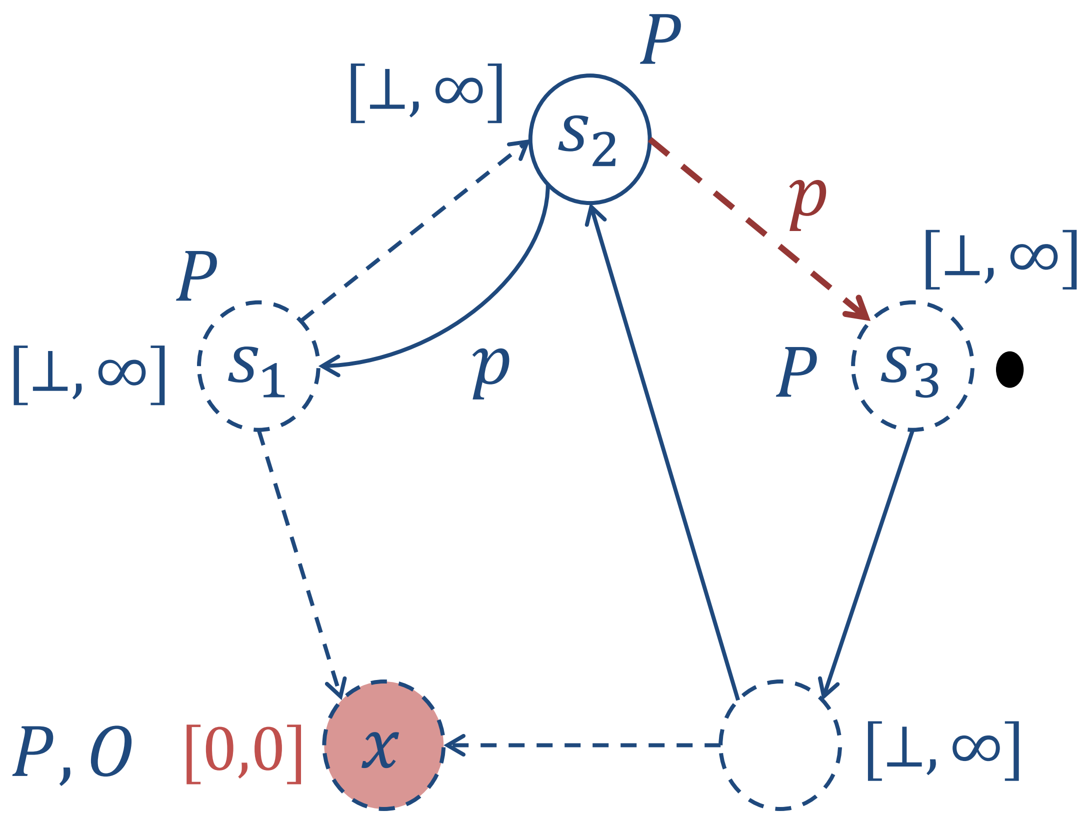


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

$$s_2 = \overline{x}$$

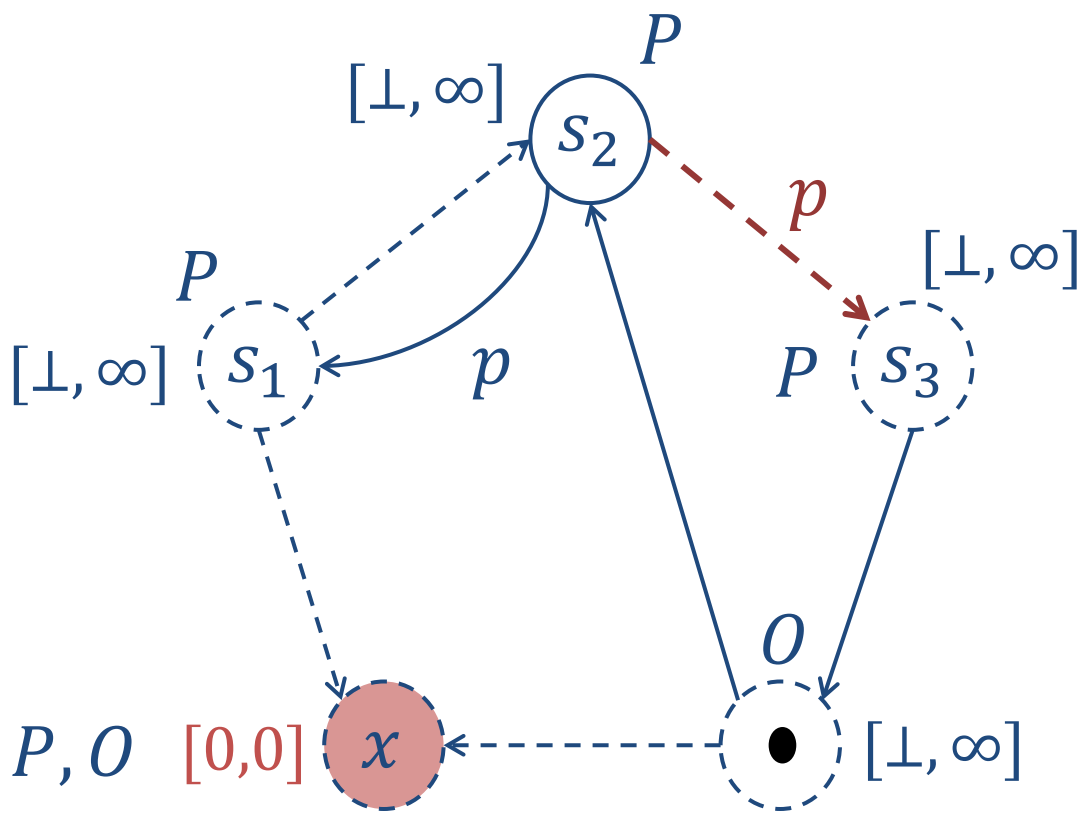


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

$$s_2 = \overline{x}$$

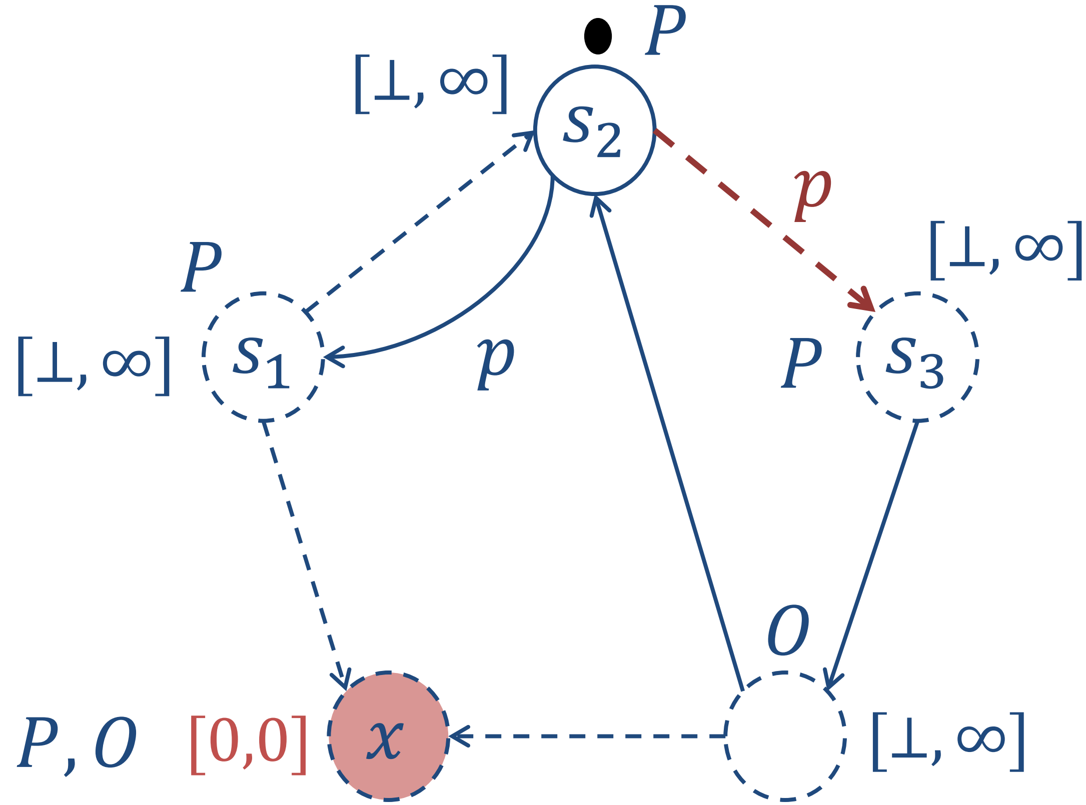


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+ ?\,(!\,s_1) \mid x^+ ?\,(!\,s_1) \mid s_1^- ?\,(!\,s_2) \mid s_3^+ ?\,(!\,s_2) \mid x^- ?\,s_2^+ ?\,(!\,s_3)$$

$$s_2 = \overline{x}$$

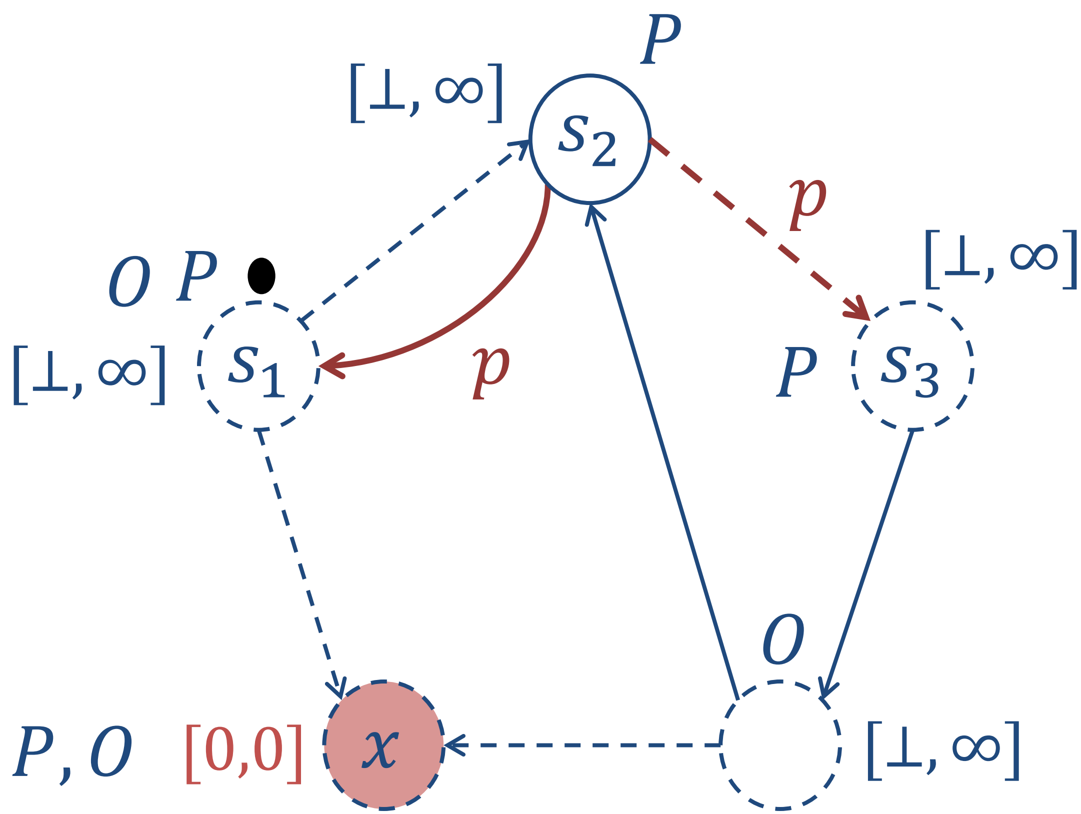


Combinational (Ternary, UNI) System

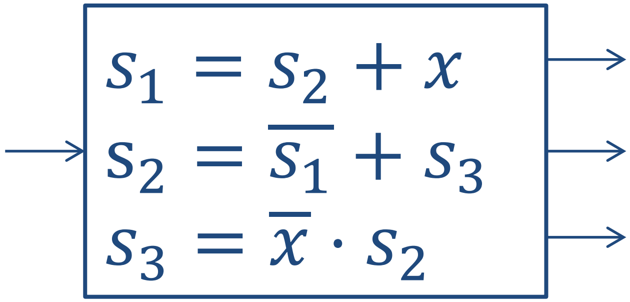
$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+ ? \, (! \, s_1) \mid x^+ ? \, (! \, s_1) \mid s_1^- ? \, (! \, s_2) \mid s_3^+ ? \, (! \, s_2) \mid x^- ? \, s_2^+ ? \, (! \, s_3)$$

$$s_2 = \overline{x}$$



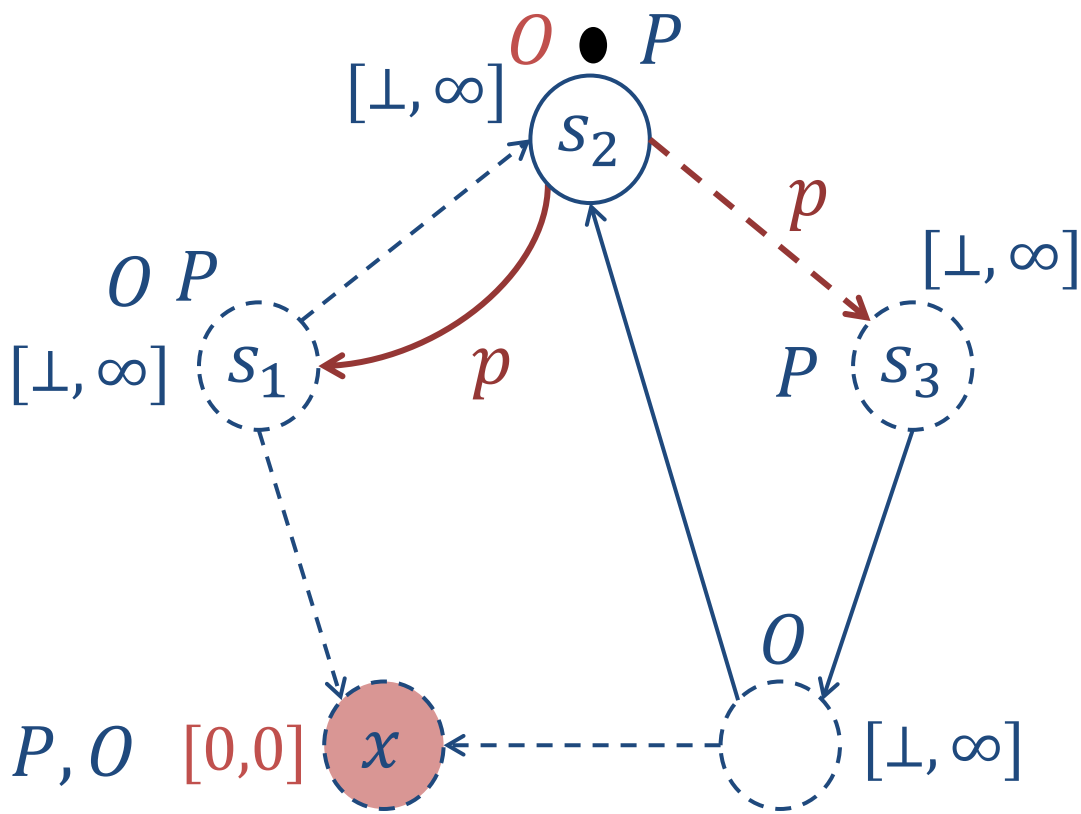
Combinational (Ternary, UNI) System



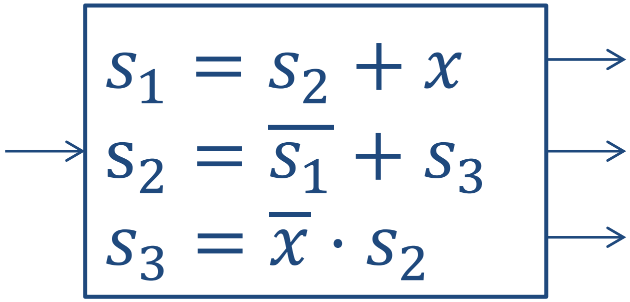
$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+ ? (! s_1) \mid x^+ ? (! s_1) \mid s_1^- ? (! s_2) \mid s_3^+ ? (! s_2) \mid x^- ? s_2^+ ? (! s_3)$$

$$s_2 = \overline{x}$$



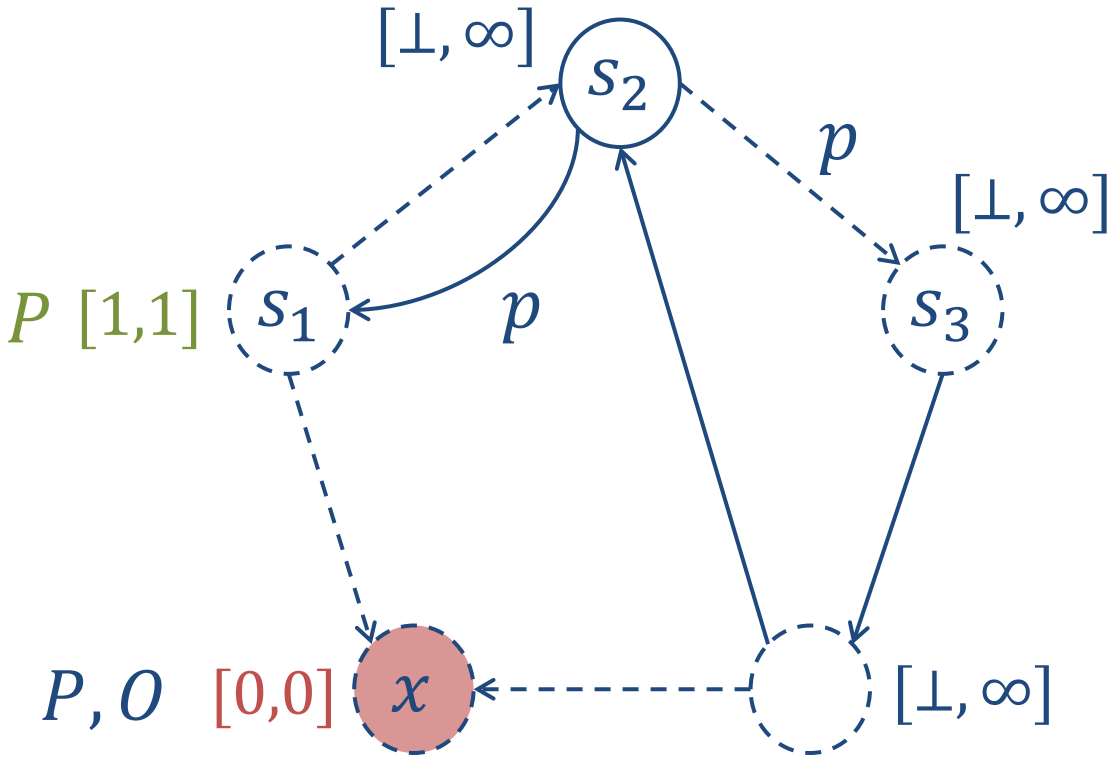
Combinational (Ternary, UNI) System



$s_2^+ ? (! s_1) \mid x^+ ? (! s_1) \mid s_1^- ? (! s_2) \mid$
$s_3^+ ? (! s_2) \mid x^- ? s_2^+ ? (! s_3)$

$s_2 = \overline{x}$

$s_1 = \overline{x} + x = 1$

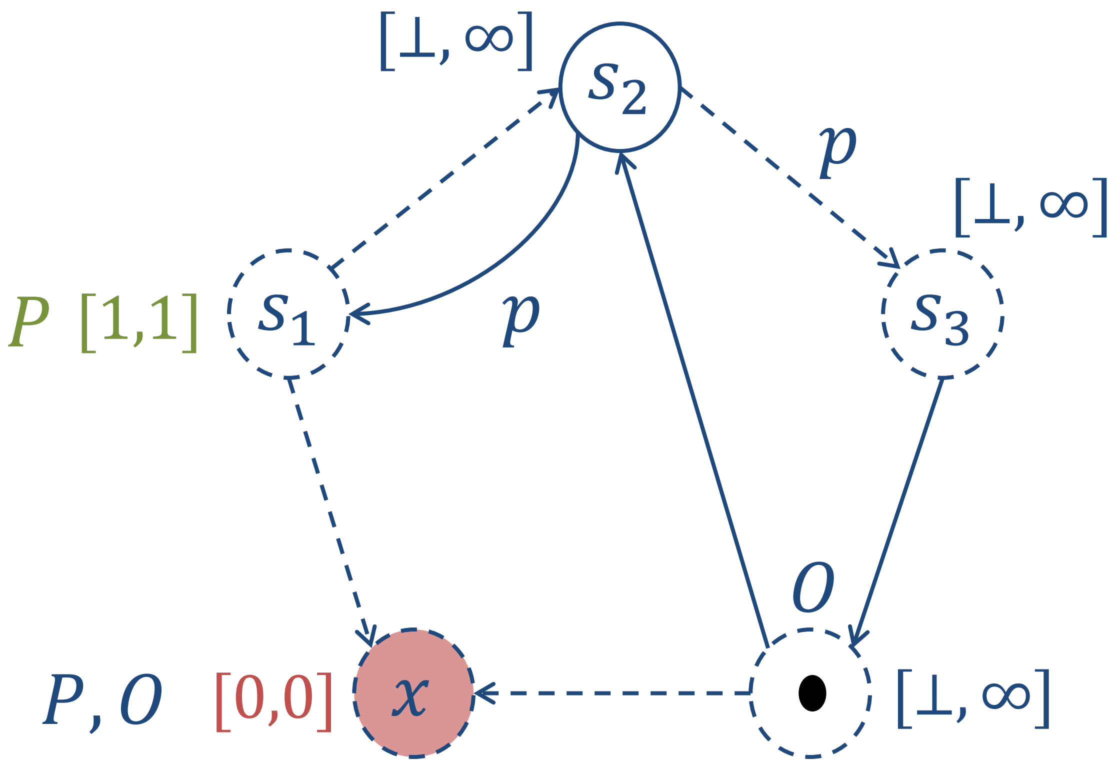


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?(!\,s_1) \mid x^+?(!\,s_1) \mid s_1^-?(!\,s_2) \mid s_3^+?(!\,s_2) \mid x^-?\,s_2^+?(!\,s_3)$$

$$s_2 = \overline{x}$$
$$s_1 = \overline{x} + x = 1$$

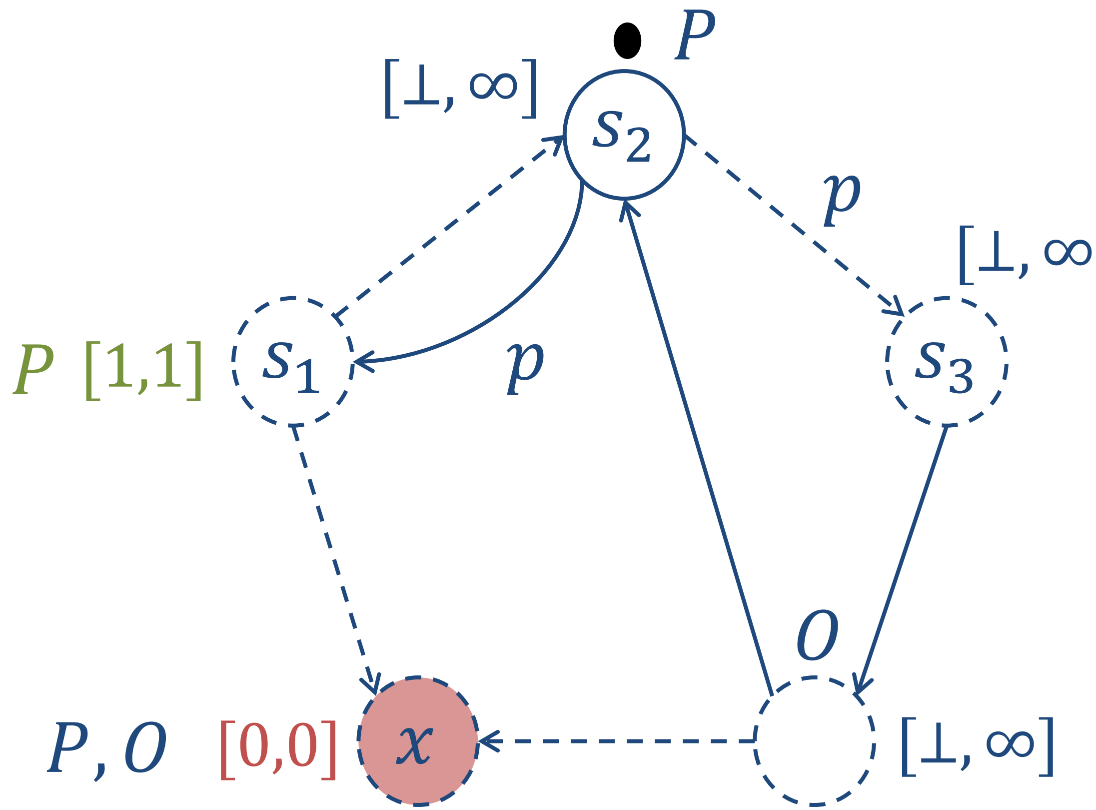


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

$$s_2 = \overline{x}$$
$$s_1 = \overline{x} + x = 1$$

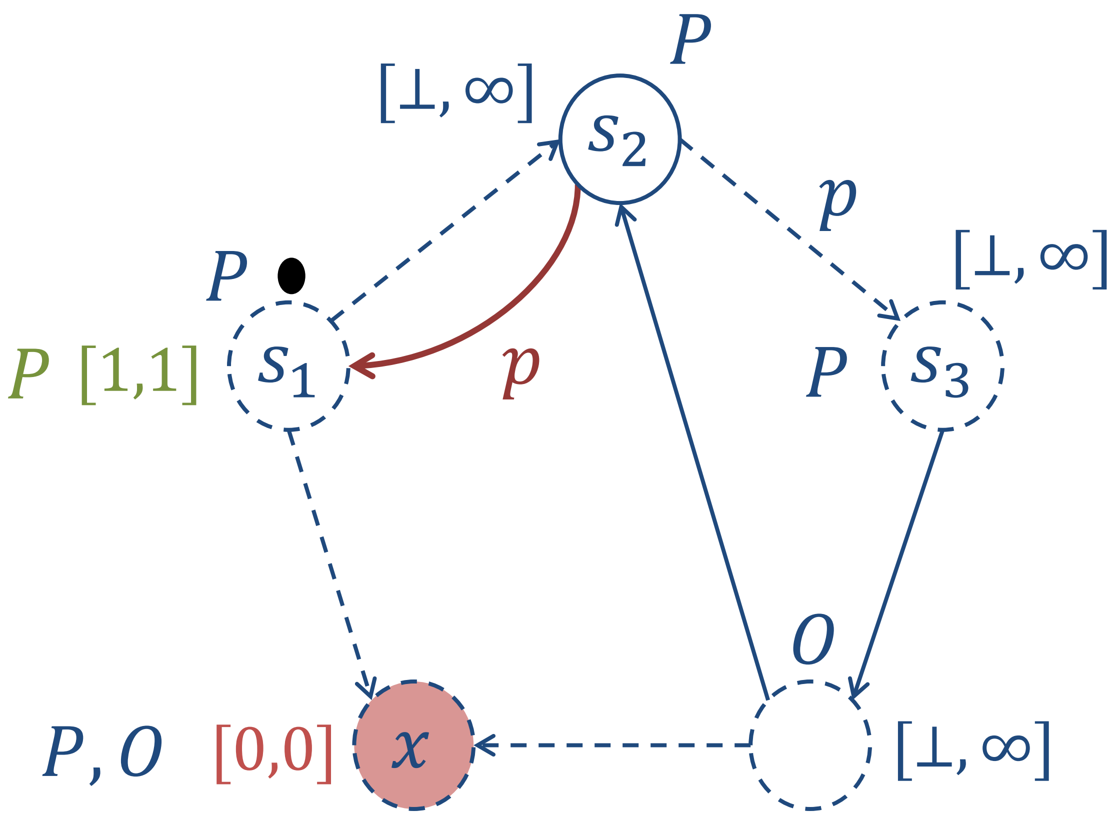


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

$$s_2 = \overline{x}$$
$$s_1 = \overline{x} + x = 1$$

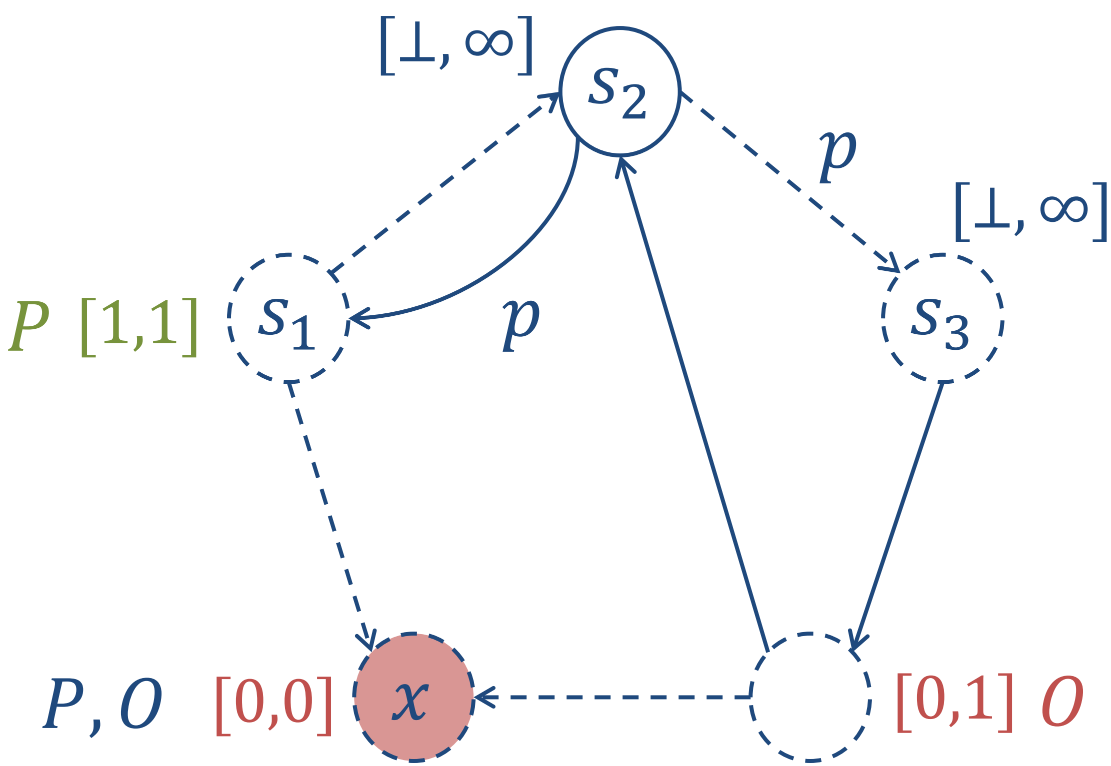


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$s_2^+?(!\,s_1) \mid x^+?(!\,s_1) \mid s_1^-?(!\,s_2) \mid$
$s_3^+?(!\,s_2) \mid x^-?\,s_2^+?(!\,s_3)$

$$s_2 = \overline{x}$$
$$s_1 = \overline{x} + x = 1$$

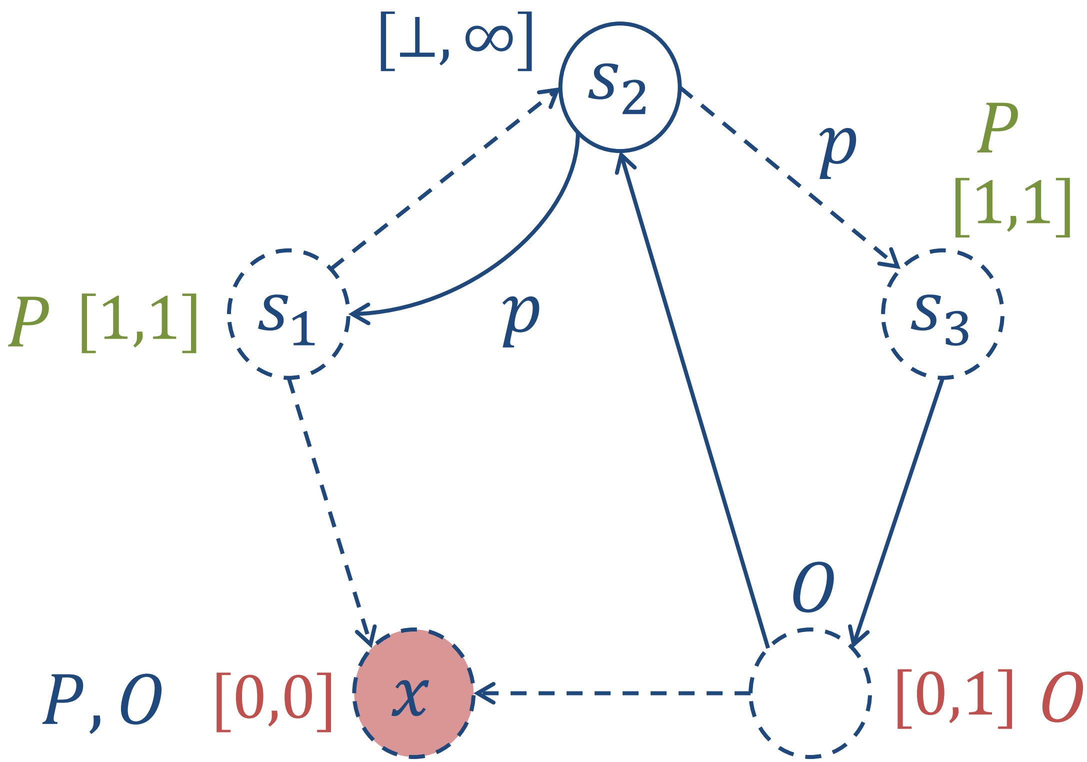


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid$$
$$s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

$$s_2 = \overline{x}$$
$$s_1 = \overline{x} + x = 1$$
$$s_3 = \overline{x} \cdot \overline{x} = \overline{x}$$

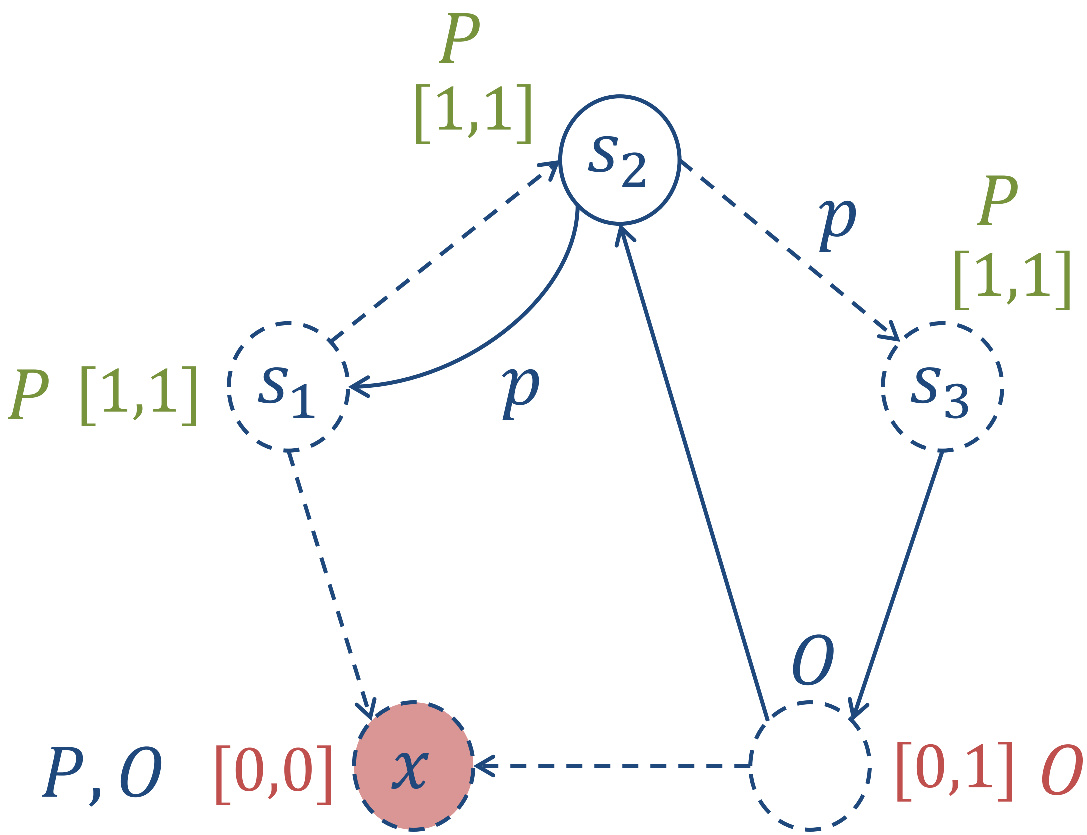


Combinational (Ternary, UNI) System

$$s_1 = s_2 + x$$
$$s_2 = \overline{s_1} + s_3$$
$$s_3 = \overline{x} \cdot s_2$$

$$s_2^+?\,(!\,s_1) \mid x^+?\,(!\,s_1) \mid s_1^-?\,(!\,s_2) \mid s_3^+?\,(!\,s_2) \mid x^-?\,s_2^+?\,(!\,s_3)$$

$$s_2 = \overline{x}$$
$$s_1 = \overline{x} + x = 1$$
$$s_3 = \overline{x} \cdot \overline{x} = \overline{x}$$



Discussion

Is Timing Analysis a Refinement of Causality Analysis?

Timed Esterel Games are computationally equivalent to Timed Ternary Simulation.

Timing Analysis = Causality Analysis + Time?

Because Mazes are logic specifications of execution schedules that can be used together with time information.

Discussion

What is WCRT anyway?

It is the **minimal** upper bound on the reaction time.

By definition WCRT determines **exact** timing.

$$WCRT_1 = min\{t \mid \exists \alpha.\, \forall h.\, \forall s \geq t.\, h(s) = \alpha\}$$
$$WCRT_2 = min\{t \mid \forall h.\, \exists \alpha.\, \forall s \geq t.\, h(s) = \alpha\}$$
$$WCRT_3 = min\{t \mid \forall h.\, \exists \alpha.\, \text{if } h \text{ stabilises} \Rightarrow \forall s \geq t.\, h(s) = \alpha\}$$

Discussion

What is WCRT anyway?

MUST prove that a WCRT algorithm delivers exact results.

But with respect to what model (abstraction)?

Mazes are flexible structures that allows us to consider different levels of atomicity, constraints on the schedule, etc.

Discussion

What is WCRT anyway?

WCRT no schedule takes more time, this requires/involves causality. It is neither simpler, in principle, nor more difficult than causality analysis.

However, Causality/combinational property depends on level of abstraction & atomicity of scheduling, scheduling model (concurrent, multithreading, …)

HERE: Esterel-style WCRT concurrent, non-inertial delays („Chaos“ [Burch‘92]) algorithms based on maze games.