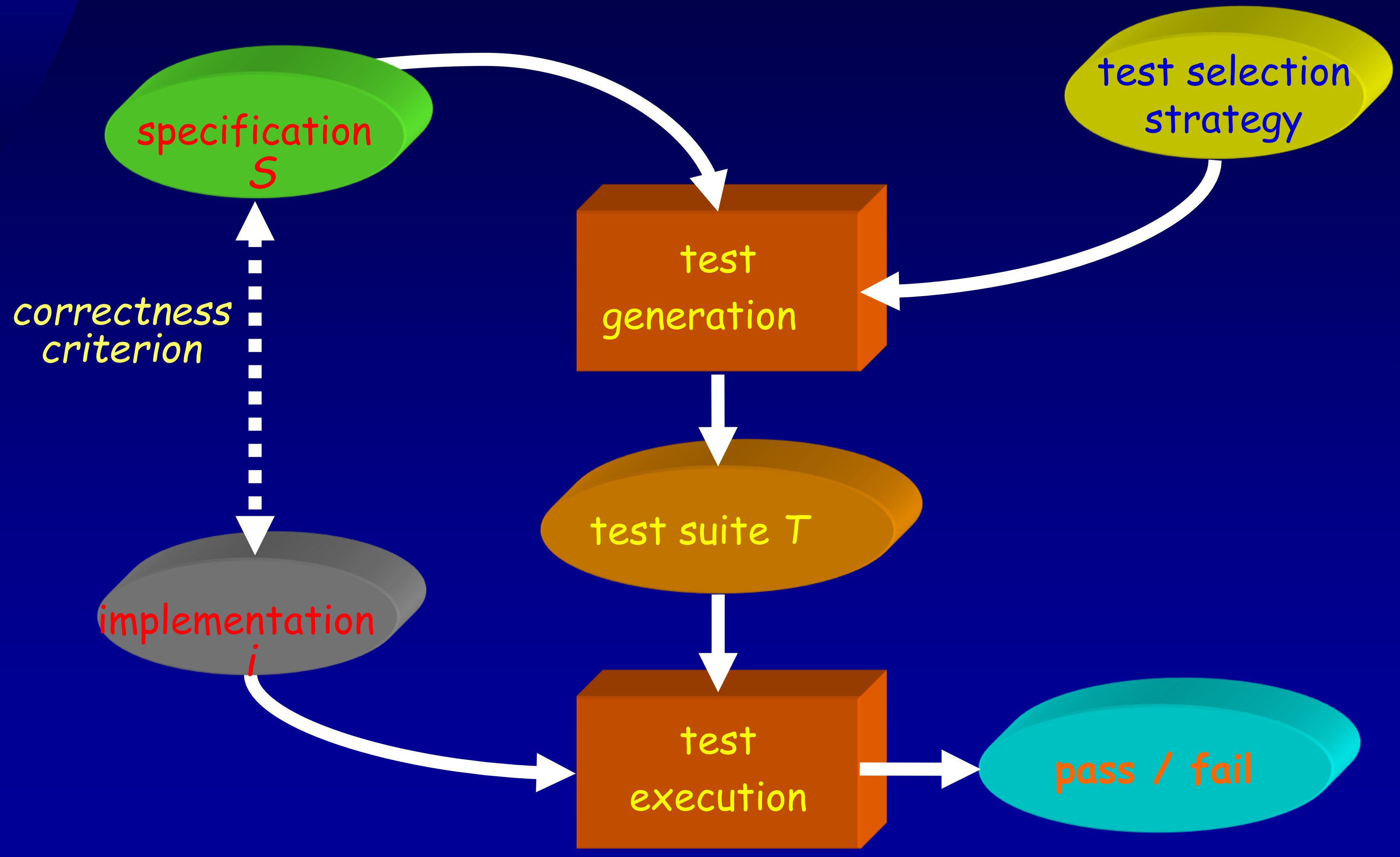
Embedded Systems INSTITUTE

Test Selection

Test Selection



Test Selection

(Infinitely) many sound test cases can be generated,
but : no time and resources to execute them all

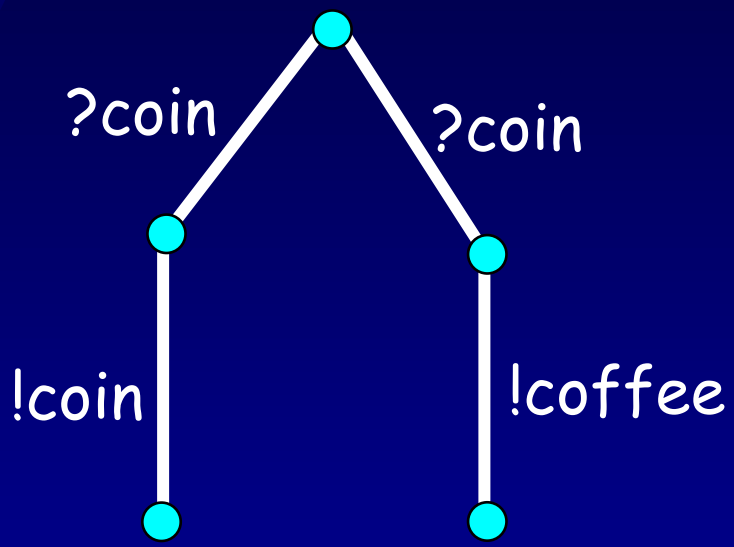
- ☞ Which are the best ones ?
- ☞ How many ?

Problem of Test Selection

Test selection :

- ☞ guided by user : test purposes
- ☞ automatic by test tool : selection strategy
- ☞ bottom line : random

Test Purpose : Example



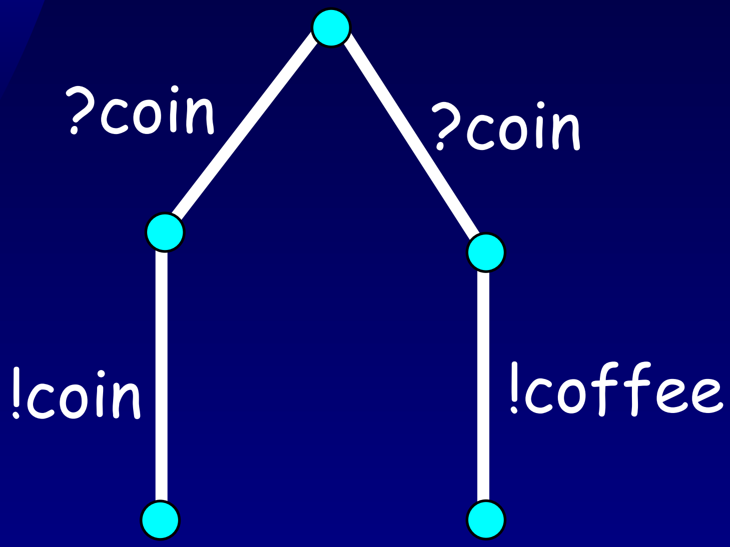
Test: can the machine deliver coffee?

Desired observation:
coffee after coin

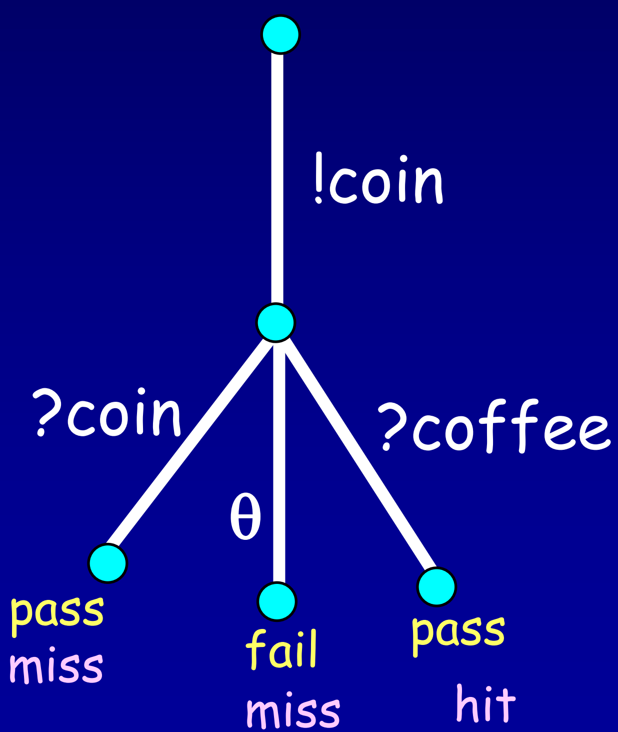
More confidence in correctness

We can only draw conclusions based on observations

Test Purpose : Example



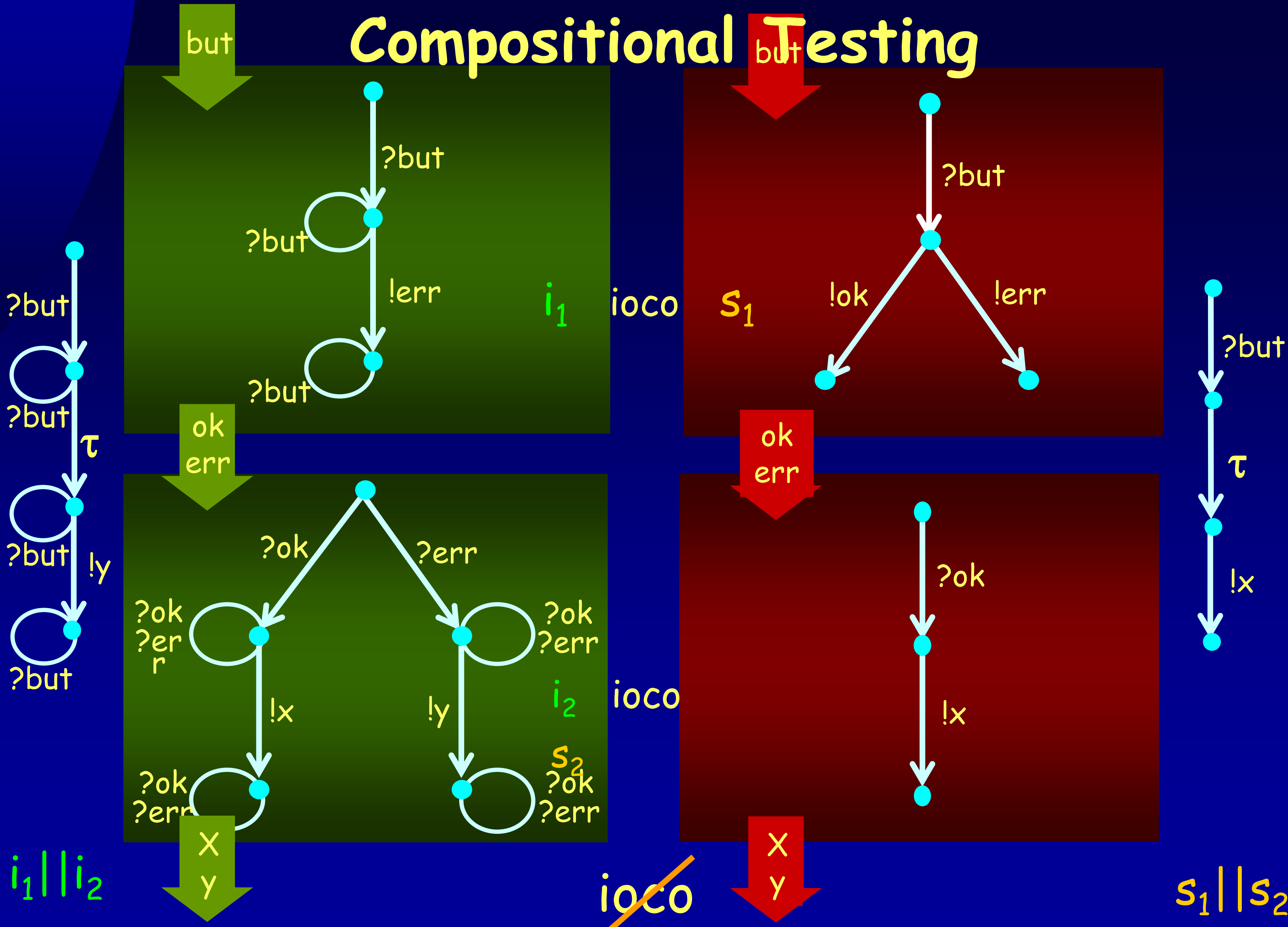
Test purpose:
can the machine deliver coffee?



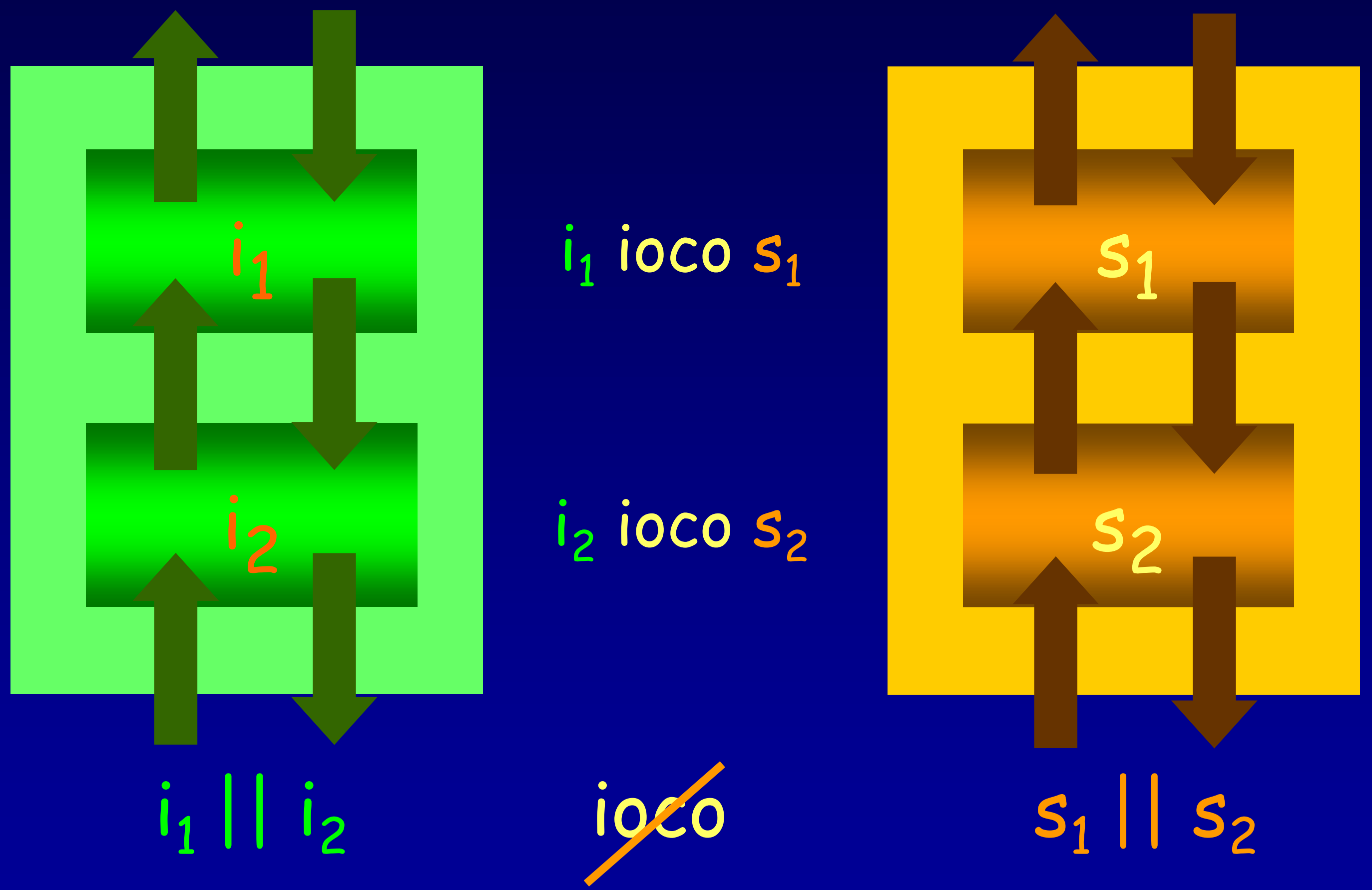
Desired observation
= observation objective :
?coin . !coffee

Compositionality

Compositional Testing



Compositional Testing

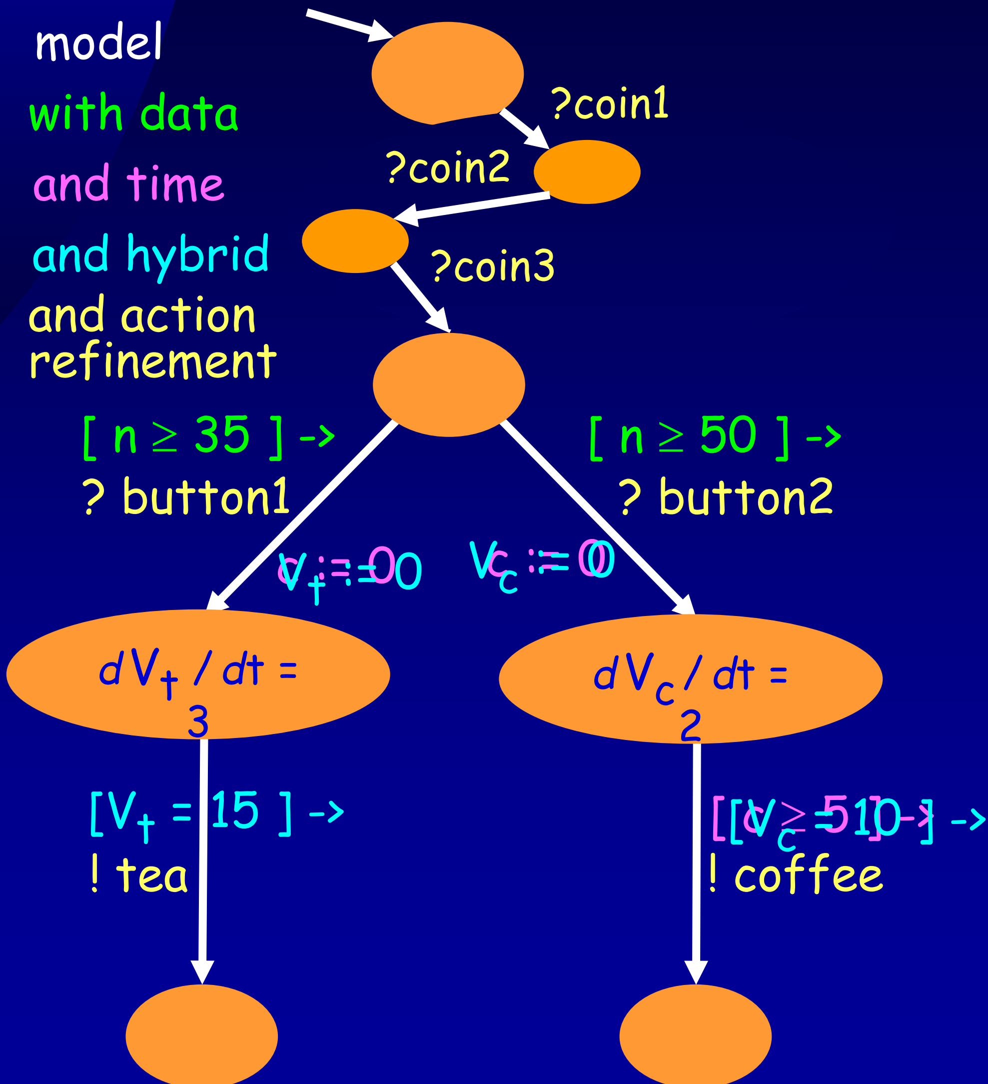


If s_1, s_2 input enabled - $s_1, s_2 \in$ IOTS - then ioco is preserved !

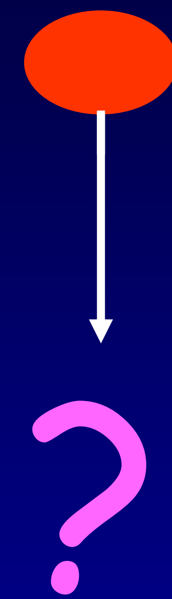
Variations of (u)**ioco**

Testing Transition Systems: Extensions

model
with data
and time
and hybrid
and action
refinement



test case



Variations on a Theme

i **ioco** s $\Leftrightarrow \forall \sigma \in Straces(s) : out\ (\ i\ \mathbf{after}\ \sigma) \subseteq out\ (\ s\ \mathbf{after}\ \sigma)$

i $\leq_{ior}$ s $\Leftrightarrow \forall \sigma \in (\ L \cup \{\delta\}\ )^* : out\ (\ i\ \mathbf{after}\ \sigma) \subseteq out\ (\ s\ \mathbf{after}\ \sigma)$

i **ioconf** s $\Leftrightarrow \forall \sigma \in traces(s) : out\ (\ i\ \mathbf{after}\ \sigma) \subseteq out\ (\ s\ \mathbf{after}\ \sigma)$

i $\mathbf{ioco}_F$ s $\Leftrightarrow \forall \sigma \in F : out\ (\ i\ \mathbf{after}\ \sigma) \subseteq out\ (\ s\ \mathbf{after}\ \sigma)$

i **uioco** s $\Leftrightarrow \forall \sigma \in Utraces(s) : out\ (\ i\ \mathbf{after}\ \sigma) \subseteq out\ (\ s\ \mathbf{after}\ \sigma)$

i **mioco** s multi-channel ioco

i **wioco** s non-input-enabled ioco

i **eco** e environmental conformance

i **sioco** s symbolic ioco

i **(r)tioco** s (real) timed tioco (Aalborg, Twente, Grenoble, Bordeaux,. . . .)

i $\mathbf{ioco}_r$ s refinement ioco

i **hioco** s hybrid ioco

i **qioco** s quantified ioco

. . . .

$\mathbf{ioco}_F$: Varying S-trace sets

Variations on a Theme

$i \; \mathbf{ioco}_F \; s \;\; \Leftrightarrow \;\; \forall \sigma \in F : \quad out\,(\, i \;\mathbf{after}\; \sigma) \;\subseteq\; out\,(\, s \;\mathbf{after}\; \sigma)$

$i \; \mathbf{ioco} \; s \;\; \Leftrightarrow \;\; \forall \sigma \in Straces(s) : out\,(\, i \;\mathbf{after}\; \sigma) \;\subseteq\; out\,(\, s \;\mathbf{after}\; \sigma)$

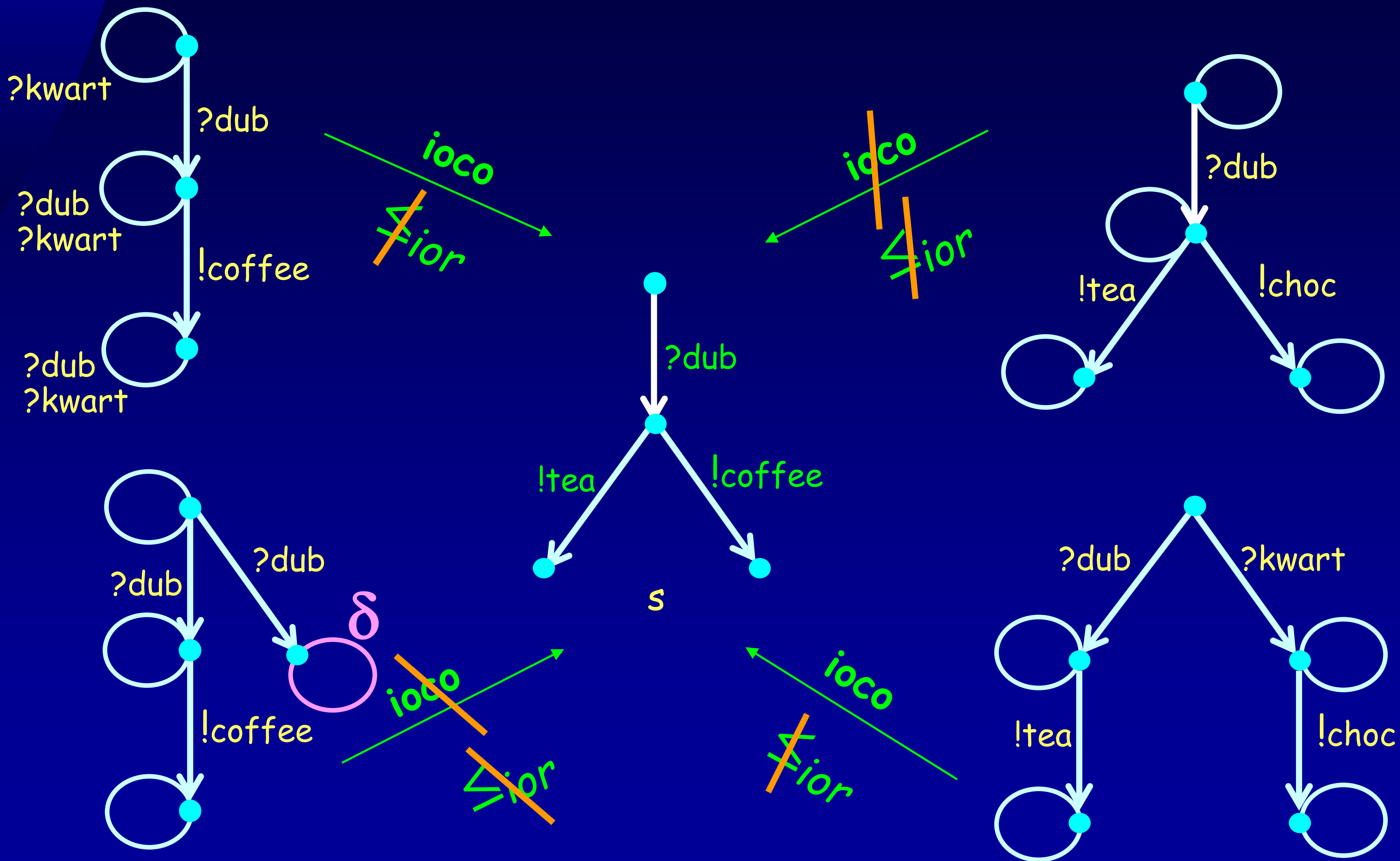
$i \; \leq_{iot} \; s \;\; \Leftrightarrow \;\; \forall \sigma \in L^* : \quad out\,(\, i \;\mathbf{after}\; \sigma) \;\subseteq\; out\,(\, s \;\mathbf{after}\; \sigma)$

$i \; \leq_{ior} \; s \;\; \Leftrightarrow \;\; \forall \sigma \in (\, L \cup \{\delta\}\,)^* : out\,(\, i \;\mathbf{after}\; \sigma) \;\subseteq\; out\,(\, s \;\mathbf{after}\; \sigma)$

$i \; \mathbf{ioconf} \; s \;\; \Leftrightarrow \;\; \forall \sigma \in traces(s) : out\,(\, i \;\mathbf{after}\; \sigma) \;\subseteq\; out\,(\, s \;\mathbf{after}\; \sigma)$

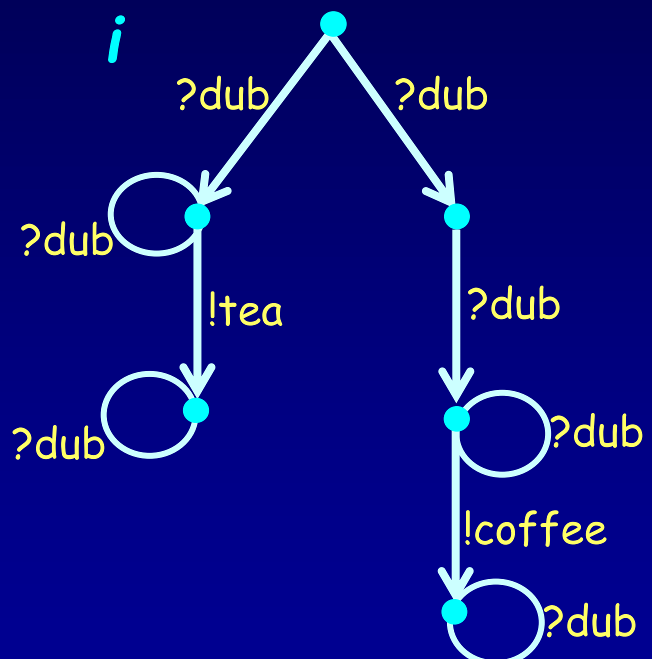
$i \; \mathbf{uioco} \; s \;\; \Leftrightarrow \;\; \forall \sigma \in Utraces(s) : out\,(\, i \;\mathbf{after}\; \sigma) \;\subseteq\; out\,(\, s \;\mathbf{after}\; \sigma)$

Implementation Relation $\leq_{ior}$



Implementation Relation ioco(nf)

$i \ \textbf{ioconf} \ s \ =_{def} \ \forall \sigma \in traces(s) : \ out(i \ \textbf{after} \ \sigma) \subseteq out(s \ \textbf{after} \ \sigma)$

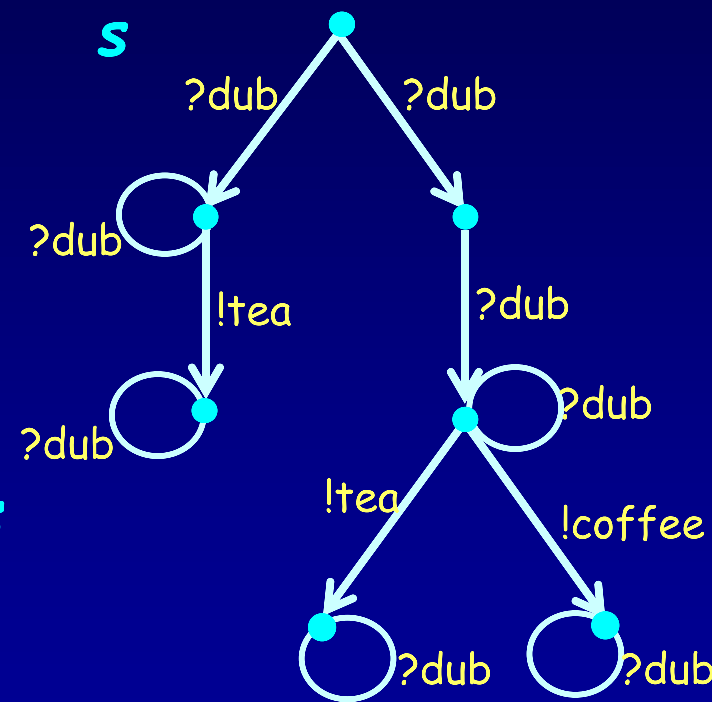


i **ioco** s

s $\cancel{\textbf{ioco}}$ i

i **ioconf** s

s **ioconf** i

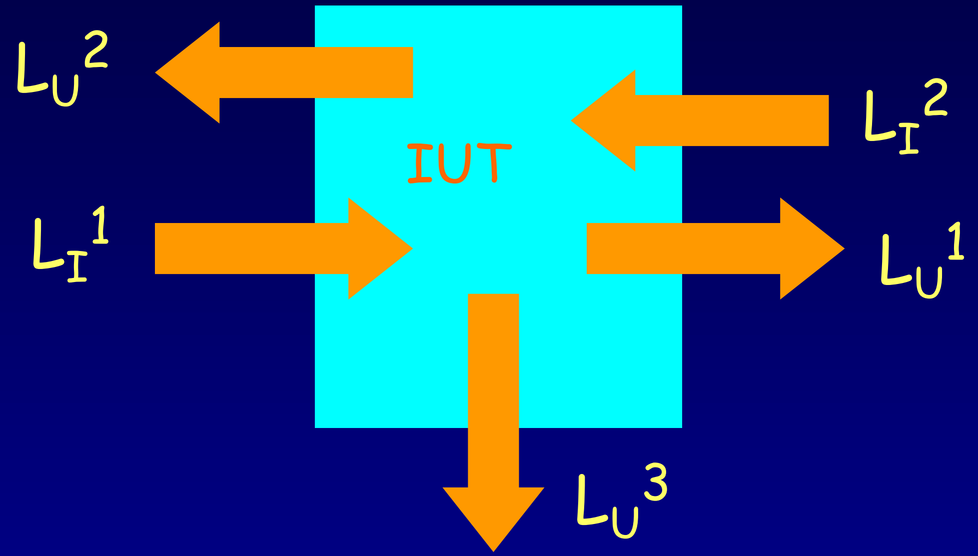


$out(i \ \textbf{after} \ ?dub.?dub) = out(s \ \textbf{after} \ ?dub.?dub) = \{ \ !tea, !coffee \ \}$

$out(i \ \textbf{after} \ ?dub.\delta.?dub) = \{ \ !coffee \ \} \neq out(s \ \textbf{after} \ ?dub.\delta.?dub) = \{ \ !tea, !coffee \ \}$

mioco : Multiple Channels

Variations on a Theme: mioco



$$i \ \textbf{mioco} \ s \iff \forall \sigma \in \mathit{Straces}'(s) : \mathit{out}'(\, i \ \textbf{after} \ \sigma) \subseteq \mathit{out}'(\, s \ \textbf{after} \ \sigma)$$

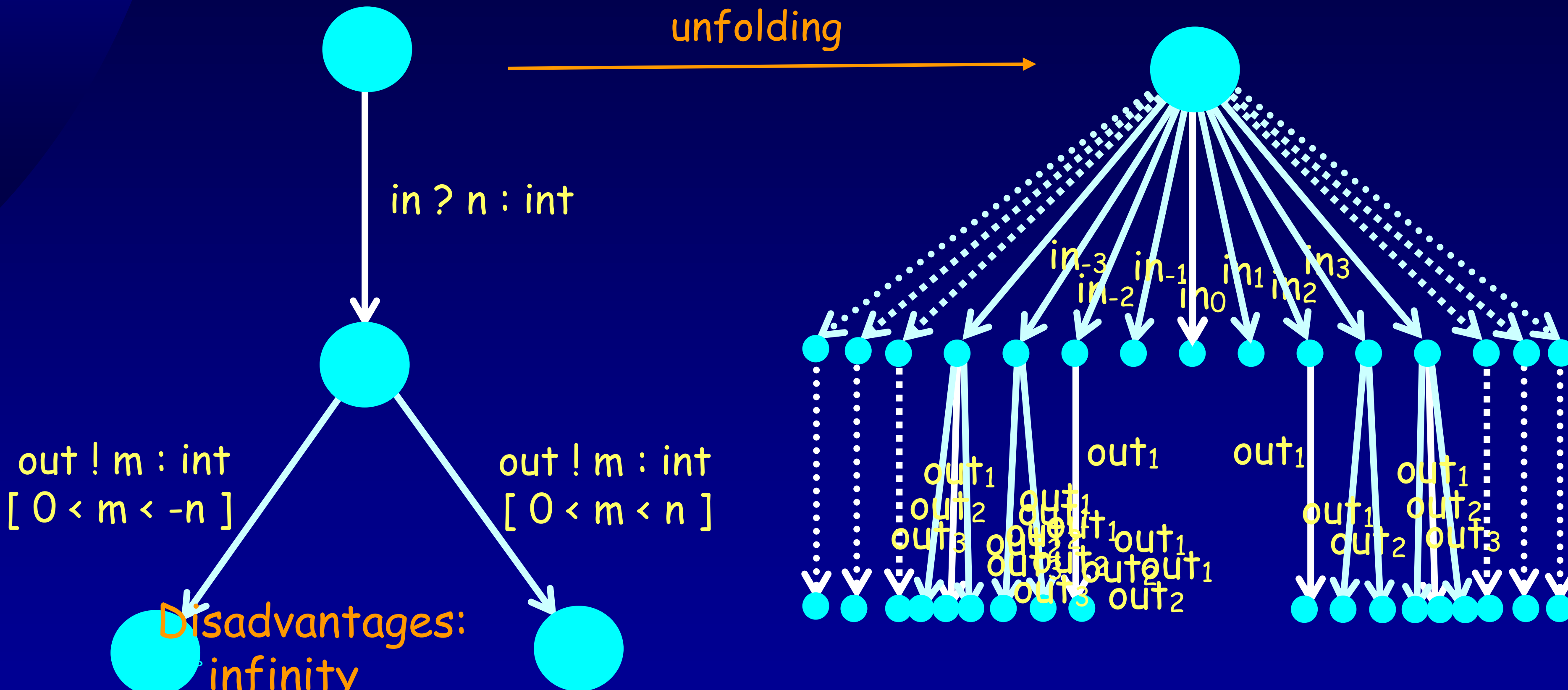
$$p \xrightarrow{\delta_k} p \quad = \quad \forall \ !x \in L^k \cup \{\tau\} \,.\ p \overset{!x}{\not\longrightarrow}$$

$$\mathit{Straces}'(\, s\,) \quad = \quad \{\ \sigma \ \in (L \cup \{\delta_k\})^* \mid s \overset{\sigma}{\Longrightarrow} \}$$

$$\mathit{out}'(\, P\,) \quad = \{\ !x \in L_U \mid p \xrightarrow{!x} ,\ p \in P \} \ \cup \ \{\ \delta_k \mid p \xrightarrow{\delta_k} p,\ p \in P \}$$

sioco : ioco with data

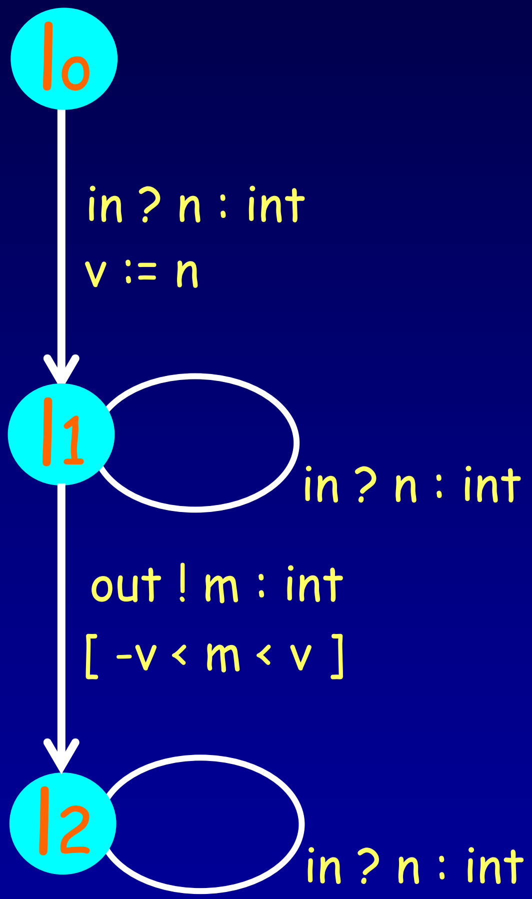
Transition System with Data



Disadvantages:
- ☞ infinity
- ☞ loss of information
  (e.g. for test selection)

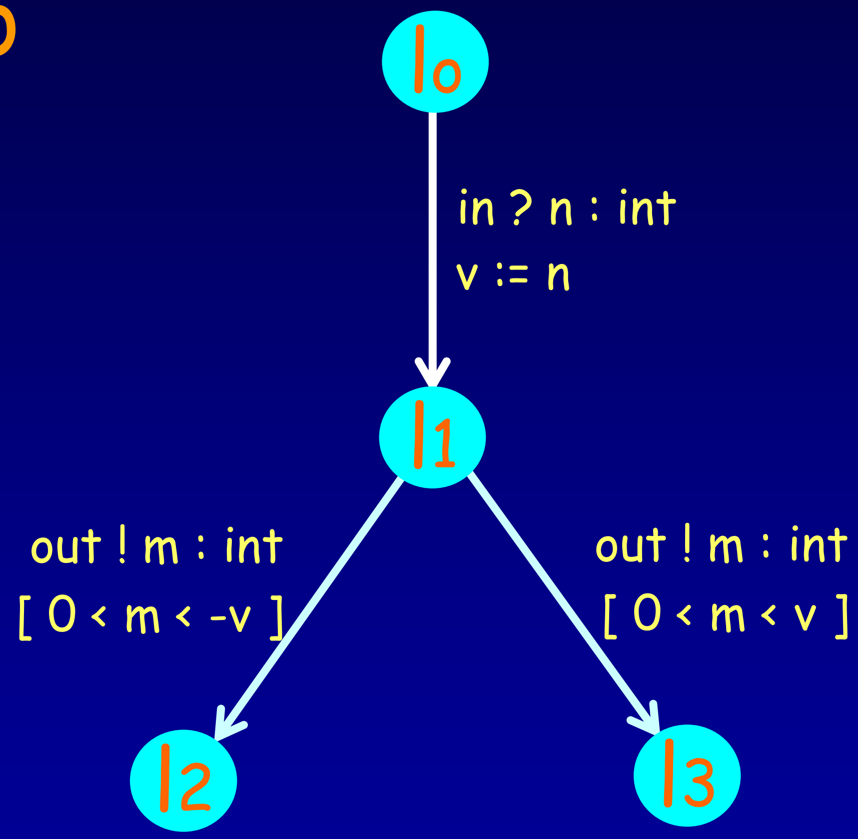
Disadvantages:
- ☞ infinity
- ☞ loss of information
  (e.g. for test selection)

Symbolic ioco



sioco

?



Symbolic ioco

Specification: IOSTS $\mathcal{S}(\iota_S) = \langle L_S, l_S, \mathcal{V}_S, \mathcal{I}, \Lambda, \rightarrow_S \rangle$
Implementation: IOSTS $\mathcal{P}(\iota_P) = \langle L_P, l_P, \mathcal{V}_P, \mathcal{I}, \Lambda, \rightarrow_P \rangle$
both initialised, implementation input-enabled, $\mathcal{V}_S \cap \mathcal{V}_P = \emptyset$
$\mathcal{F}_s$: a set of symbolic extended traces satisfying $[\![\mathcal{F}_s]\!]_{\iota_S} \subseteq \mathit{Straces}((l_0, \iota))$;

$\mathcal{P}(\iota_P)$ $\mathbf{sioco}_{\mathcal{F}_s}$ $\mathcal{S}(\iota_S)$ iff

$$\forall (\sigma, \chi) \in \mathcal{F}_s \; \forall \lambda_\delta \in \Lambda_U \cup \{\delta\} : \iota_P \cup \iota_S \models \overline{\forall}_{\widehat{\mathcal{I}} \cup \mathcal{I}} (\Phi(l_P, \lambda_\delta, \sigma) \wedge \chi \rightarrow \Phi(l_S, \lambda_\delta, \sigma))$$

$$\text{where } \Phi(\xi, \lambda_\delta, \sigma) = \bigvee \{\varphi \wedge \psi \mid (\lambda_\delta, \varphi, \psi) \in \mathbf{out}_s((\xi, \top, \mathsf{id})_0 \mathbf{after}_s(\sigma, \top))\}$$

Theorem 1.

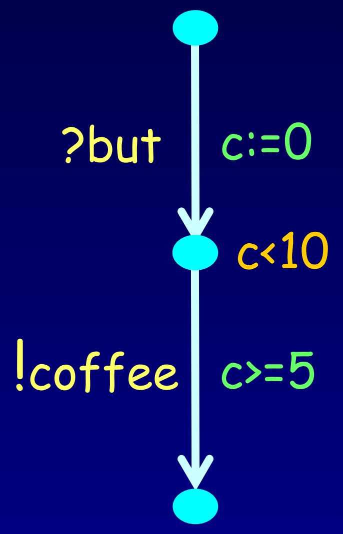
$$\mathcal{P}(\iota_P) \;\mathbf{sioco}_{\mathcal{F}_s}\; \mathcal{S}(\iota_S) \quad \mathit{iff} \quad [\![\mathcal{P}]\!]_{\iota_P} \;\mathbf{ioco}_{[\![\mathcal{F}_s]\!]_{\iota_S}}\; [\![\mathcal{S}]\!]_{\iota_S}$$

(r)tioco : ioco with time

Timed Model-Based Testing

- In many systems real-time properties are crucial
- Approach:
  - Extension of IOTS/**ioco** theory
    - Timed Input Output Transition Systems (TIOTS)
    - Timed Implementation Relations: build on **ioco**
- Challenges:
  - Is time input or output ?
  - Quiescence: How long is there never eventually no output?

Timed Input-Output Transition Systems



TIOTS : $\langle$ **Q**, L_I, L_U, $\mathbf{R}_{\geq 0}$, **T**, q_0 $\rangle$

Observable actions: L_I, L_U

delay $d \in \mathbf{R}_{\geq 0}$

Unobservable action: τ

Specifications are TIOTS

Implementations are assumed

to behave as input-enabled TIOTS

The Untimed Implementation Relation ioco

$$i \; \mathbf{ioco} \; s \;=_{def}\; \forall \sigma \in \mathit{Straces}(s) : \; \mathit{out}(i \; \mathbf{after} \; \sigma) \subseteq \mathit{out}(s \; \mathbf{after} \; \sigma)$$

$\downarrow$ *Straces* $\quad\downarrow$ **after** $\quad\downarrow$ *out*

$$\delta(p) \;=\; \forall \; !x \in L_U \cup \{\tau\} . \;\; p \overset{!x}{\not\longrightarrow}$$

$$\mathit{Straces}(s) \;=\; \{ \; \sigma \in (L \cup \{\delta\})^* \;|\; s \overset{\sigma}{\Longrightarrow} \; \}$$

$$\mathit{out}(p) \;=\; \{ \; !x \in L_U \;|\; p \overset{!x}{\Longrightarrow} \; \} \;\cup\; \{ \delta \;|\; \delta(p) \; \}$$

$$\mathit{out}(P) \;=\; \bigcup \{ \; \mathit{out}(p) \;|\; p \in P \; \}$$

$$p \; \mathbf{after} \; \sigma \;=\; \{ \; p' \;|\; p \overset{\sigma}{\Longrightarrow} p' \; \}$$

Some Timed Implementation Relations

$$i \ \textbf{ioco} \ s \ =_{def} \ \forall \sigma \in Straces(s) : \ out(i \ \textbf{after} \ \sigma) \subseteq out(s \ \textbf{after} \ \sigma)$$

$$\downarrow \qquad \downarrow \qquad \downarrow \qquad \downarrow$$

$$\textbf{tioco}_X \qquad ? \qquad ? \qquad ?$$

A Timed Implementation Relation

$i \ \mathbf{ioco} \ s \ =_{def} \ \forall \sigma \in \mathit{Straces}(s) : \ \mathit{out}(i \ \mathbf{after} \ \sigma) \subseteq \mathit{out}(s \ \mathbf{after} \ \sigma)$

$\downarrow$ **tioco** $\quad$ $\downarrow$ *ttraces* $\quad$ $\downarrow$ $\mathbf{after}_t$ $\quad$ $\downarrow$ out_{AG}

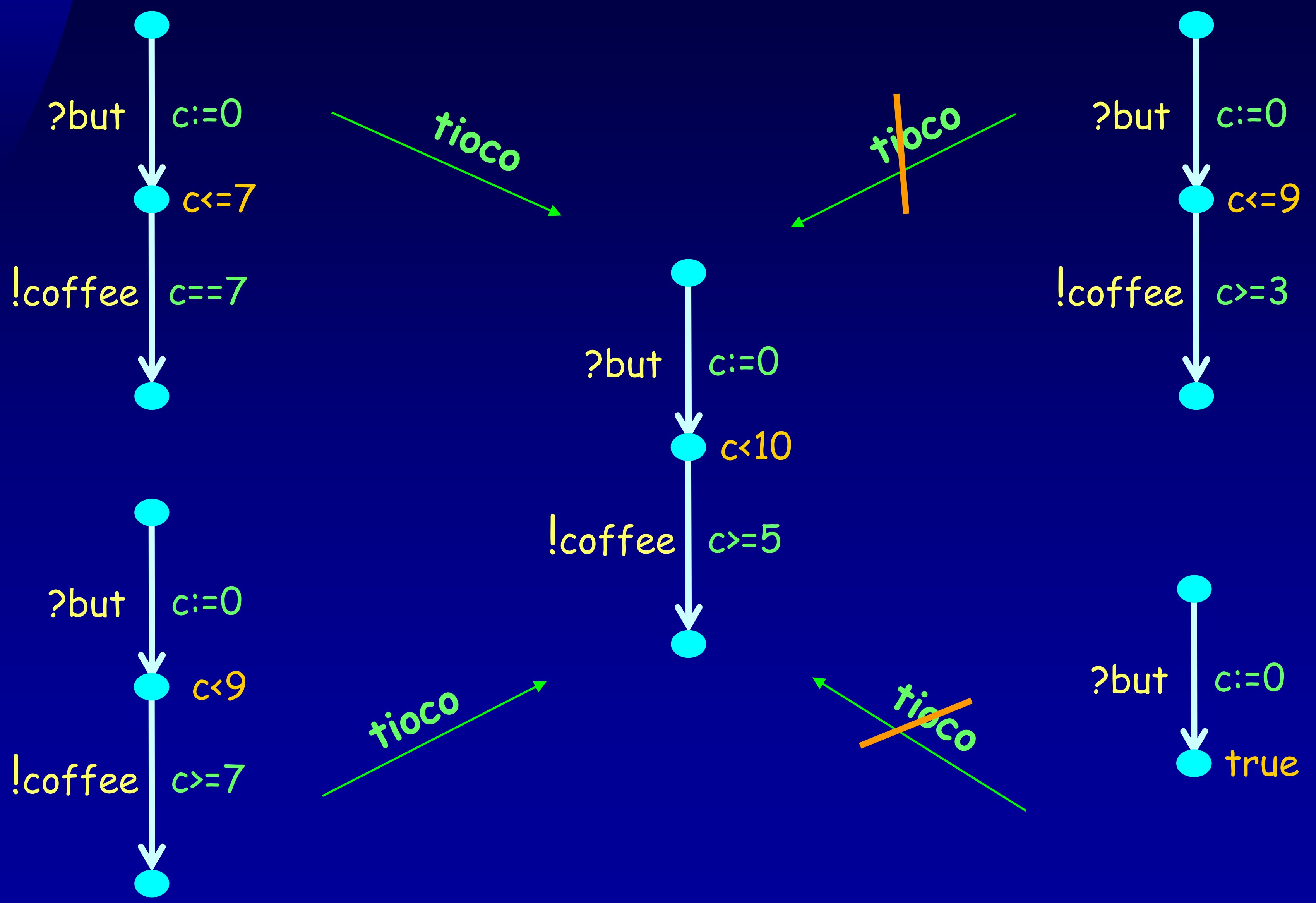
$$\delta(p) = \times$$

$$\mathit{ttraces}(s) = \{ \sigma \in (L \cup \mathbf{R}_{\geq 0})^* \mid s \stackrel{\sigma}{\Longrightarrow} \}$$

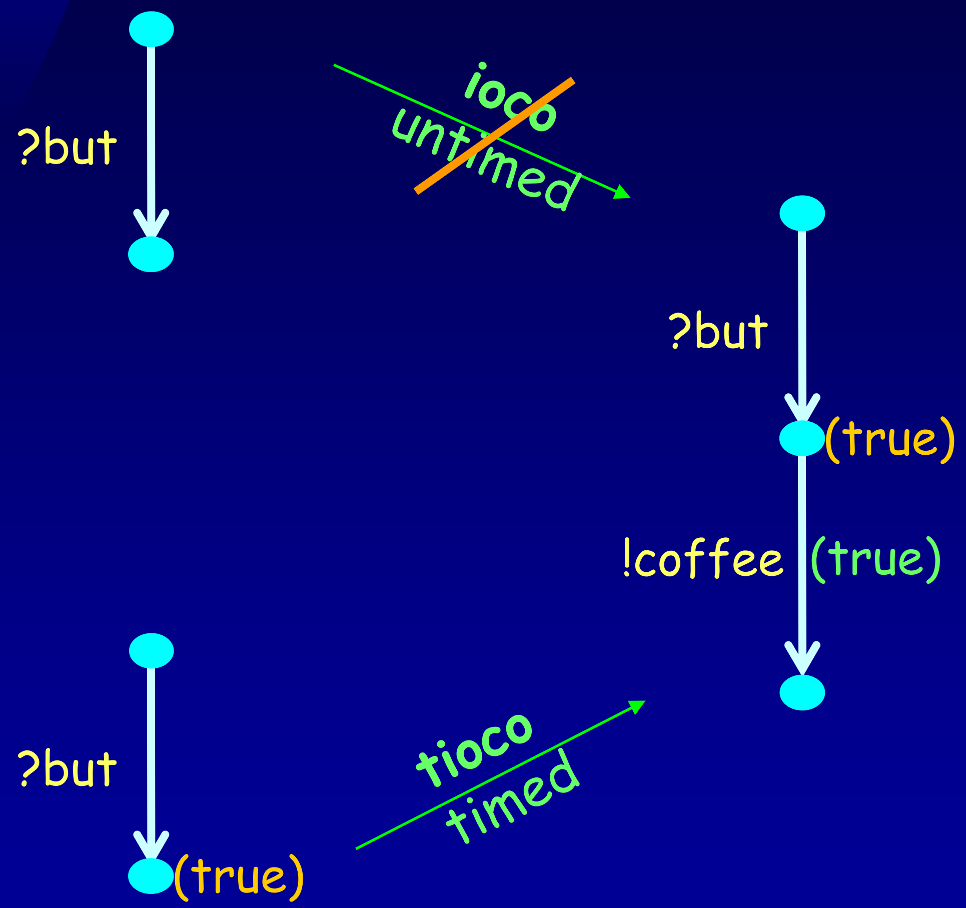
$$\mathit{out}_{AG}(p) = \{ x \in L_U \cup \mathbf{R}_{\geq 0} \mid p \stackrel{x}{\Longrightarrow} \}$$

$$p \ \mathbf{after}_t \ \sigma = \{ p' \mid p \stackrel{\sigma}{\Longrightarrow} p', \ \sigma \in (L \cup \mathbf{R}_{\geq 0})^* \}$$

A Timed Implementation Relation tioco



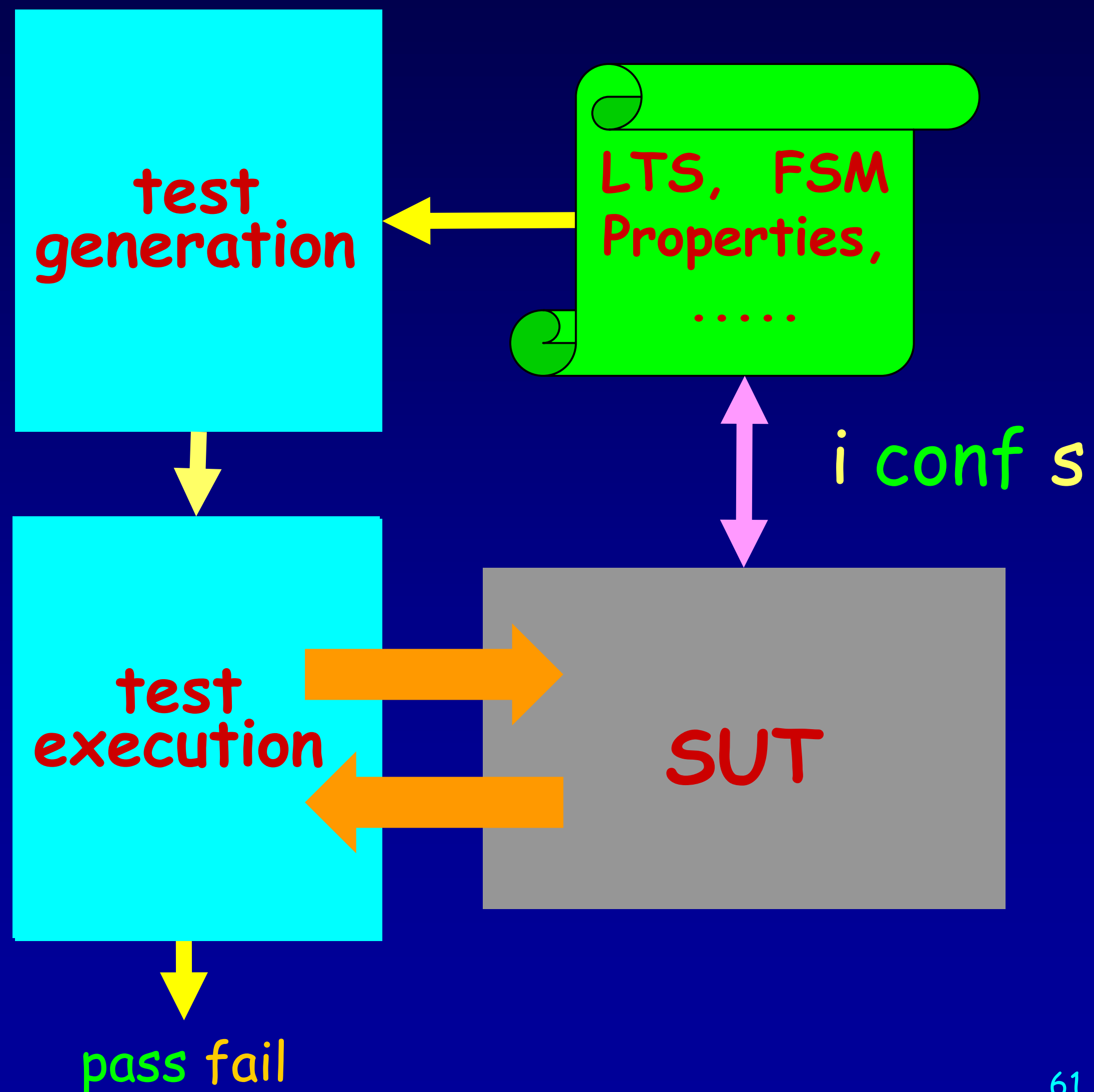
Unbounded Delay



- And suppose you wish to reject this IUT: how long would you wait ?
- Untimed ioco: quiescence to express that there eventually is !coffee
- But when is eventually ?

But There is More than LTS . . .

Model Based Testing

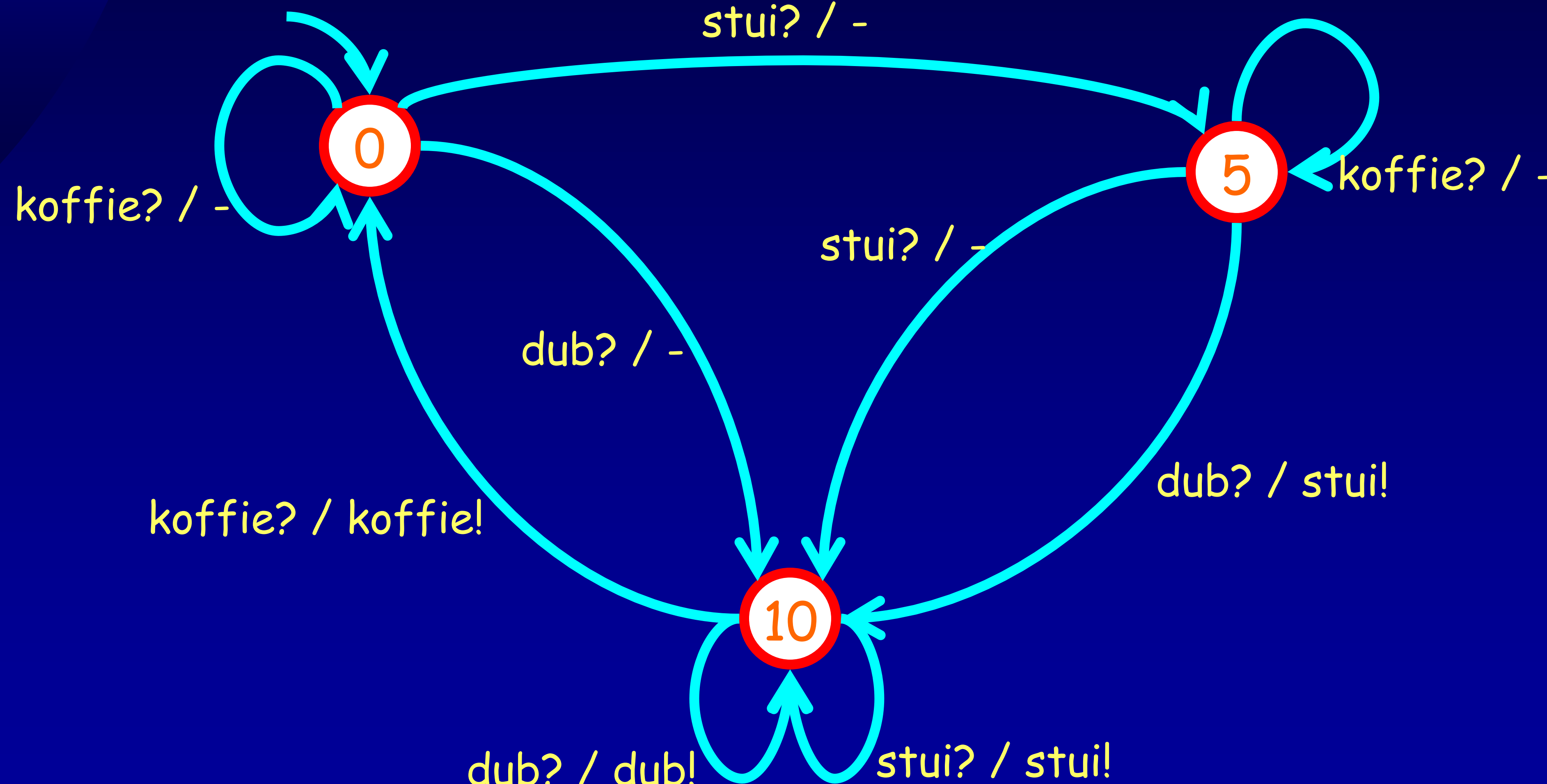


i conf s

exhaustive $\Uparrow$ $\Downarrow$ *sound*

i || TS $\rightarrow$ pass

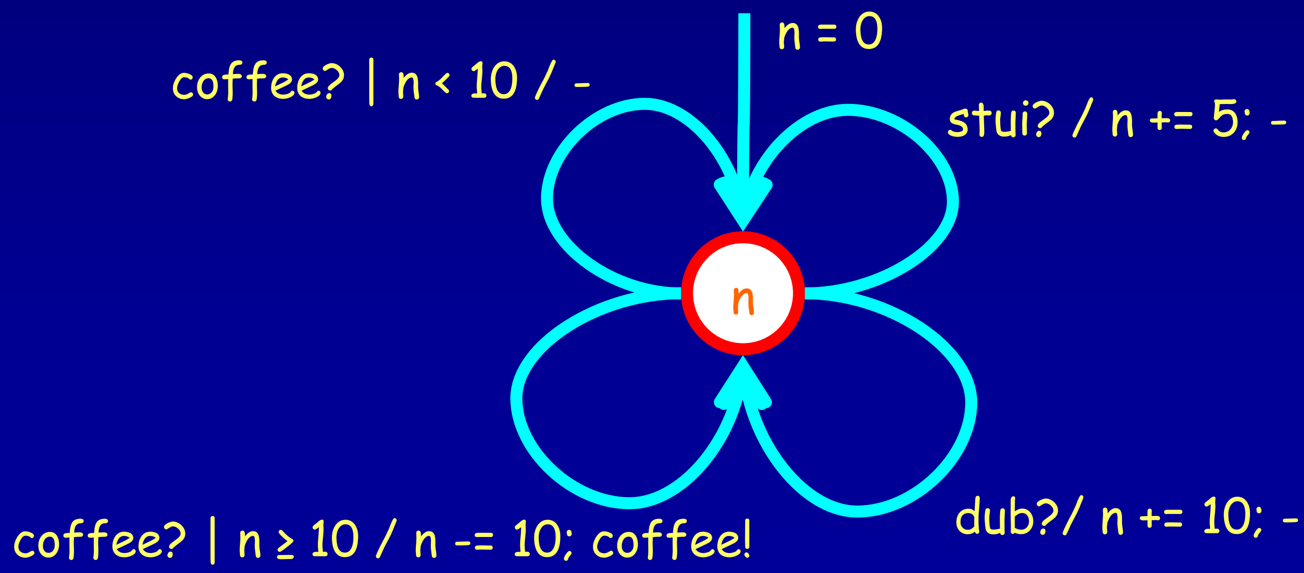
Finite State Machine (FSM)
Mealy Machine



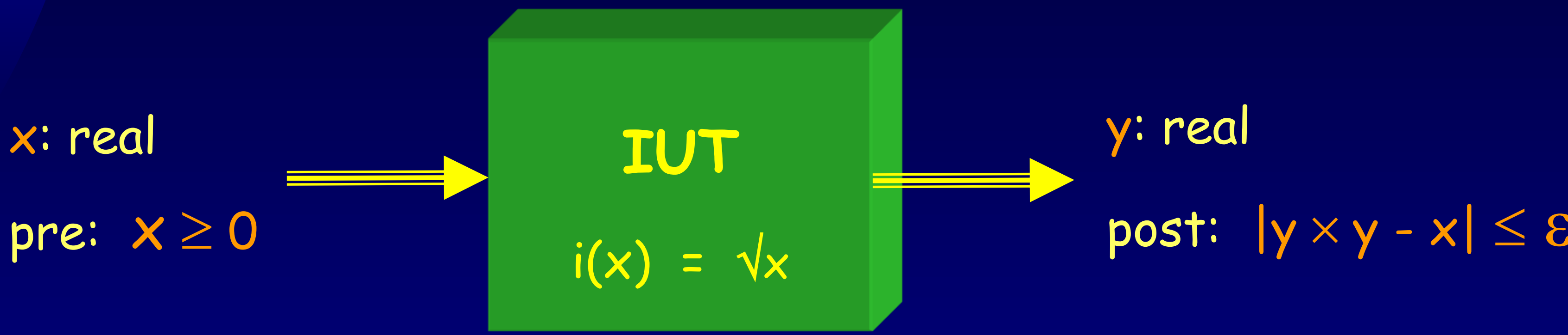
Coffee Machine: FSM with Data

☞ use a variable to record the amount of money inserted

- note this is a different machine; no money back
- you can insert money for 3 coffee before retrieving coffee
  - dub? dub? dub? coffee? coffee? coffee?
- infinitely many states; same inputs and outputs



Testing Properties of Sequential Input/Output Programs



- ☞ Specification: property over x and y
  - ♦ $\text{property}(x,y) \;=\; x \geq 0 \;\Rightarrow\; |y \times y - x| \leq \varepsilon$
- ☞ Implementation is function $i :: X \rightarrow Y$
- ☞ Test set $T \subseteq X$
  - ♦ Tools like G∀ST and QuickCheck generate thousands of tests by systematic traversal of all values of type X
  - ♦ But still: what is a "good" set ?