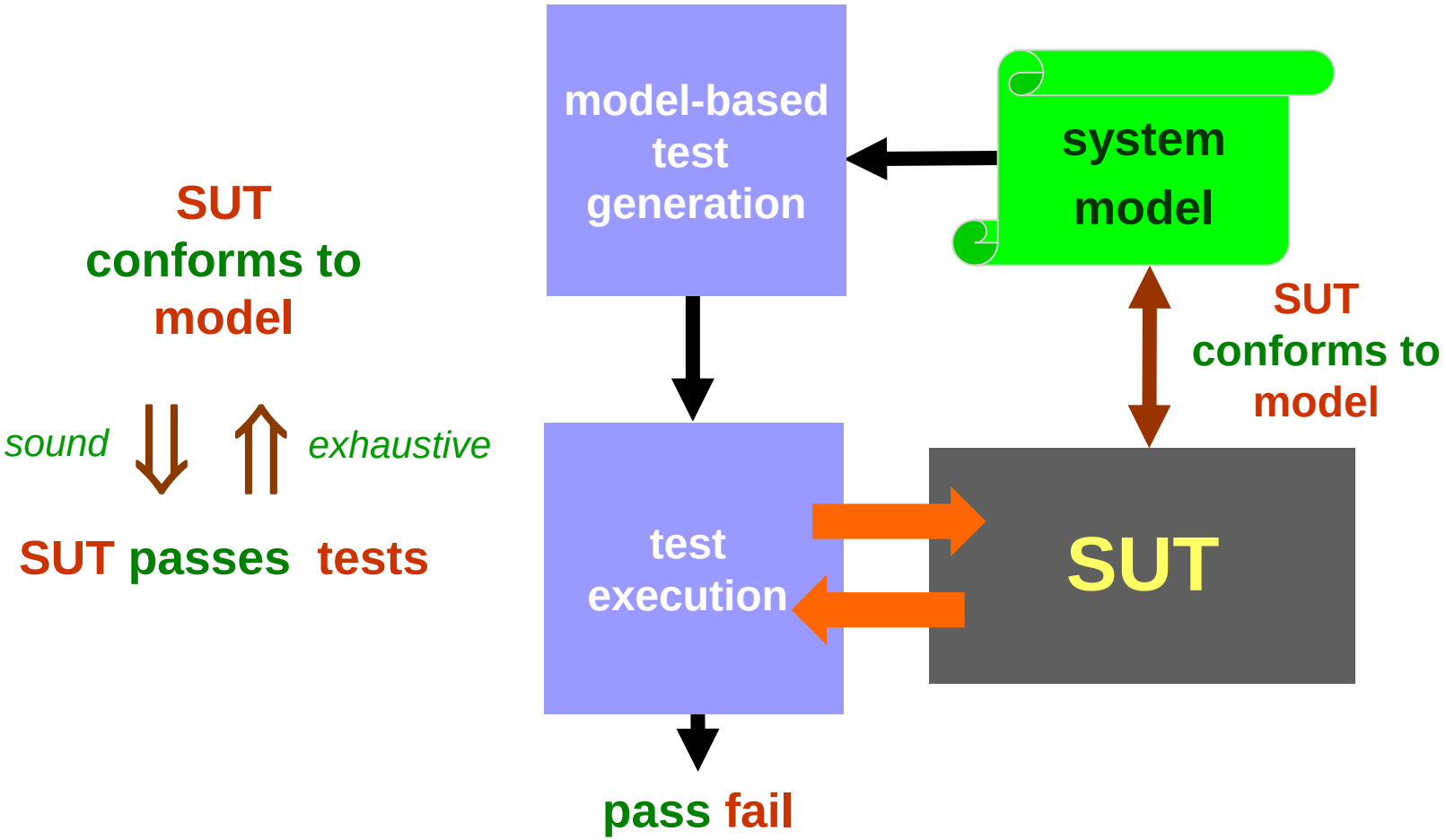
A Theory of Model-Based Testing with Labelled Transition Systems

*The **uioco** Theory*

MBT : Model-Based Testing

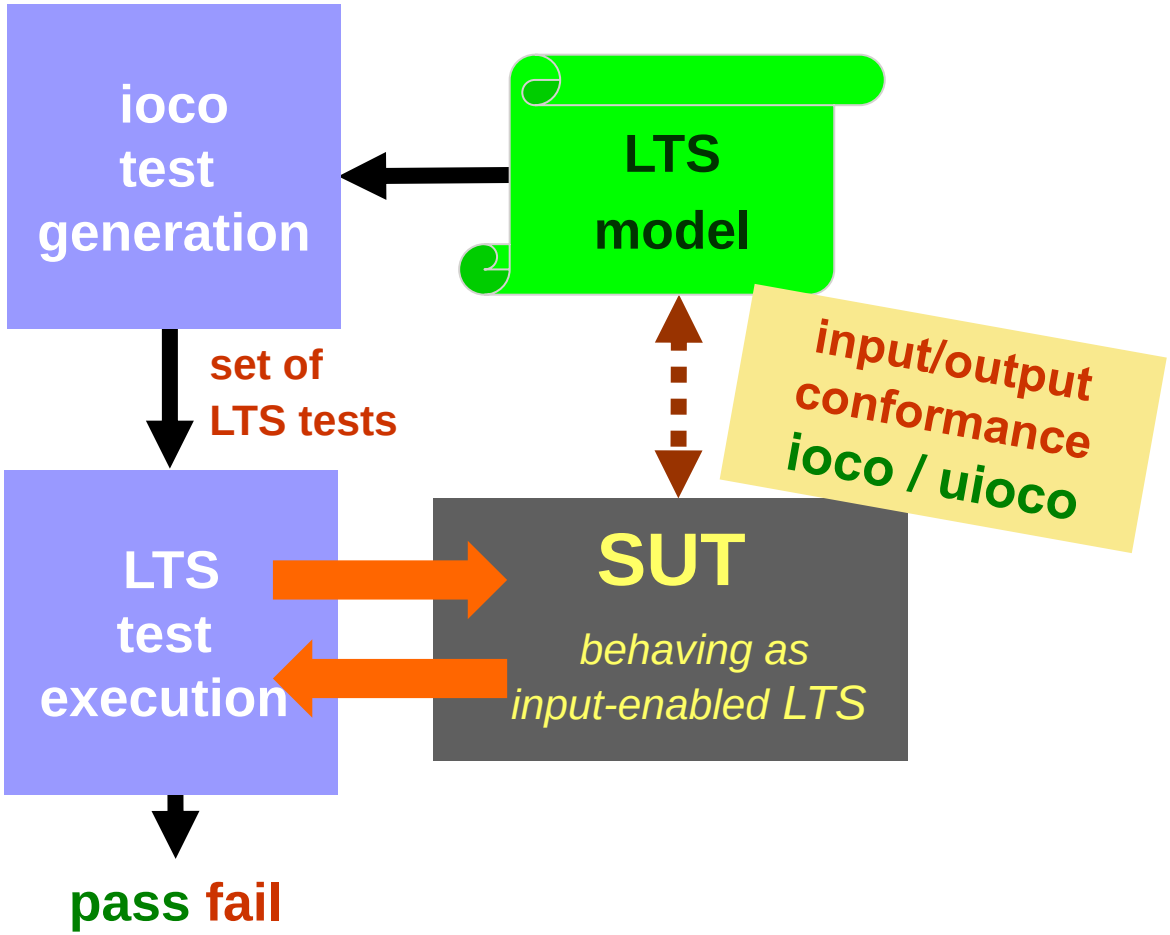


MBT : Labelled Transitions Systems

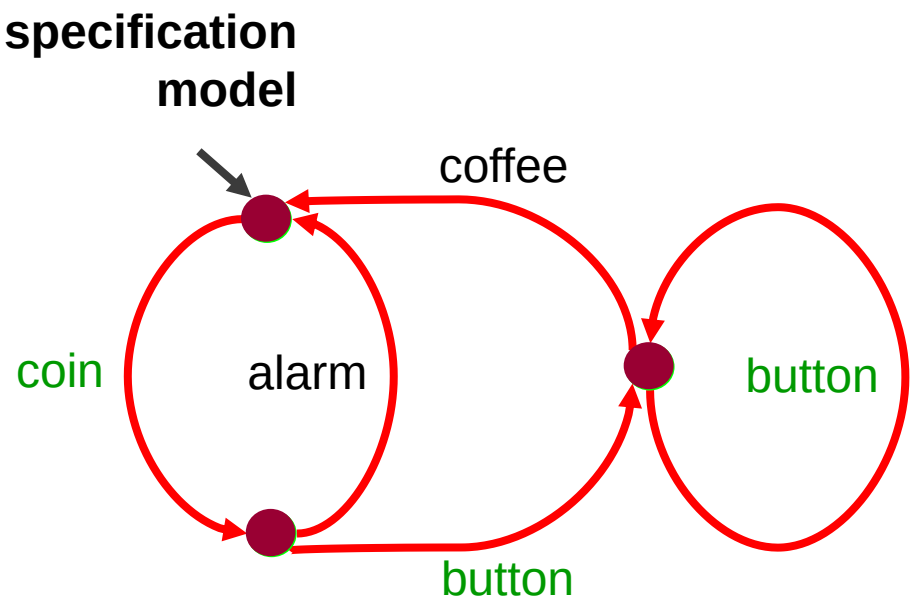
SUT uioco model

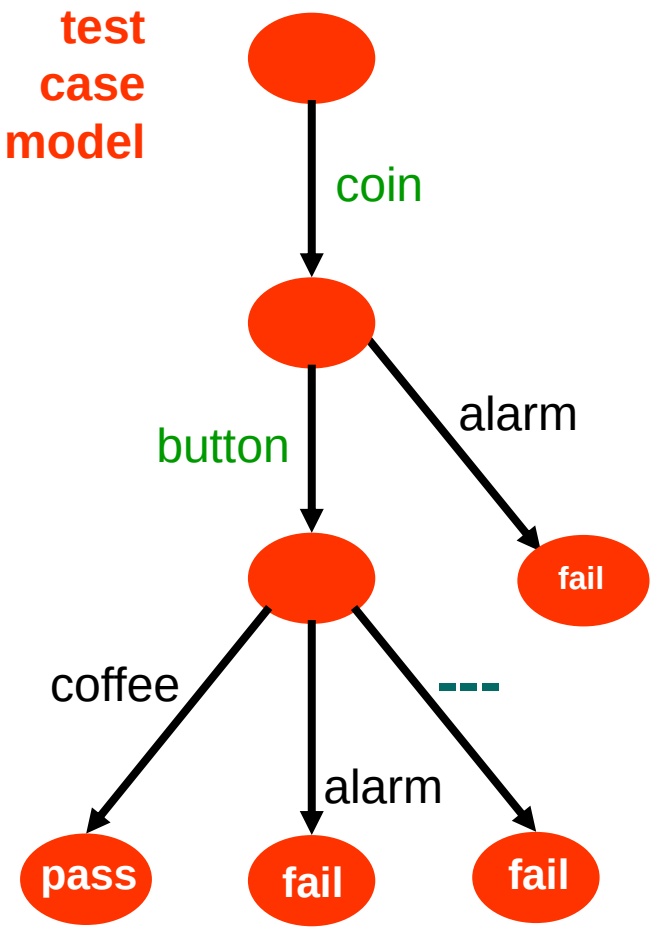
sound ⇓ ⇑ *exhaustive*

SUT passes tests

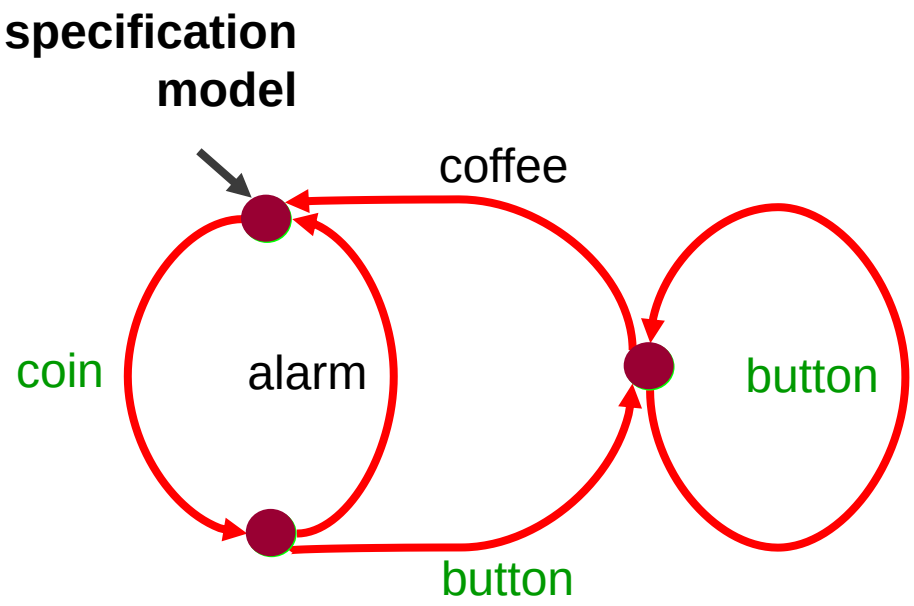


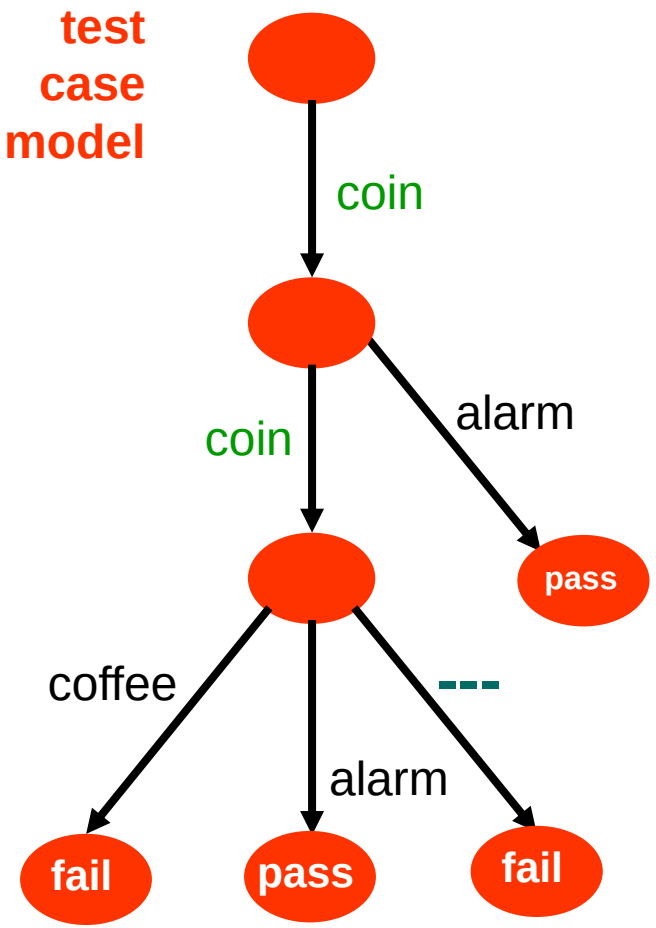
Models: Generation of Test Cases



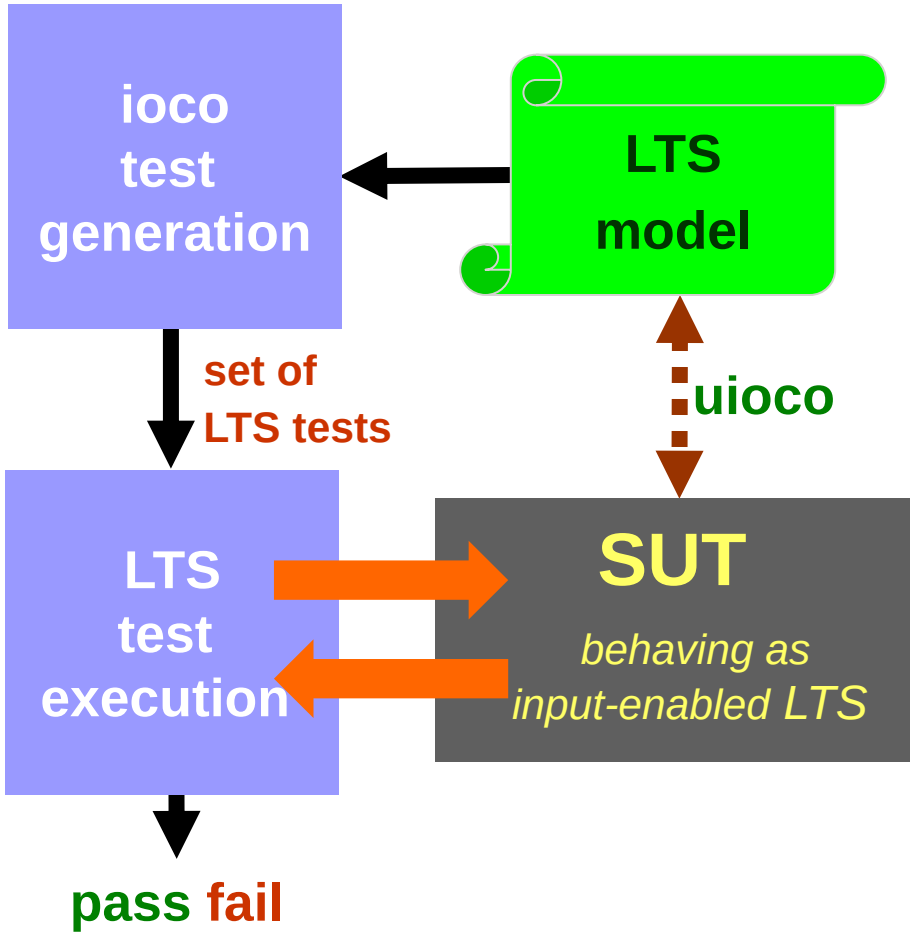


Models: Generation of Test Cases





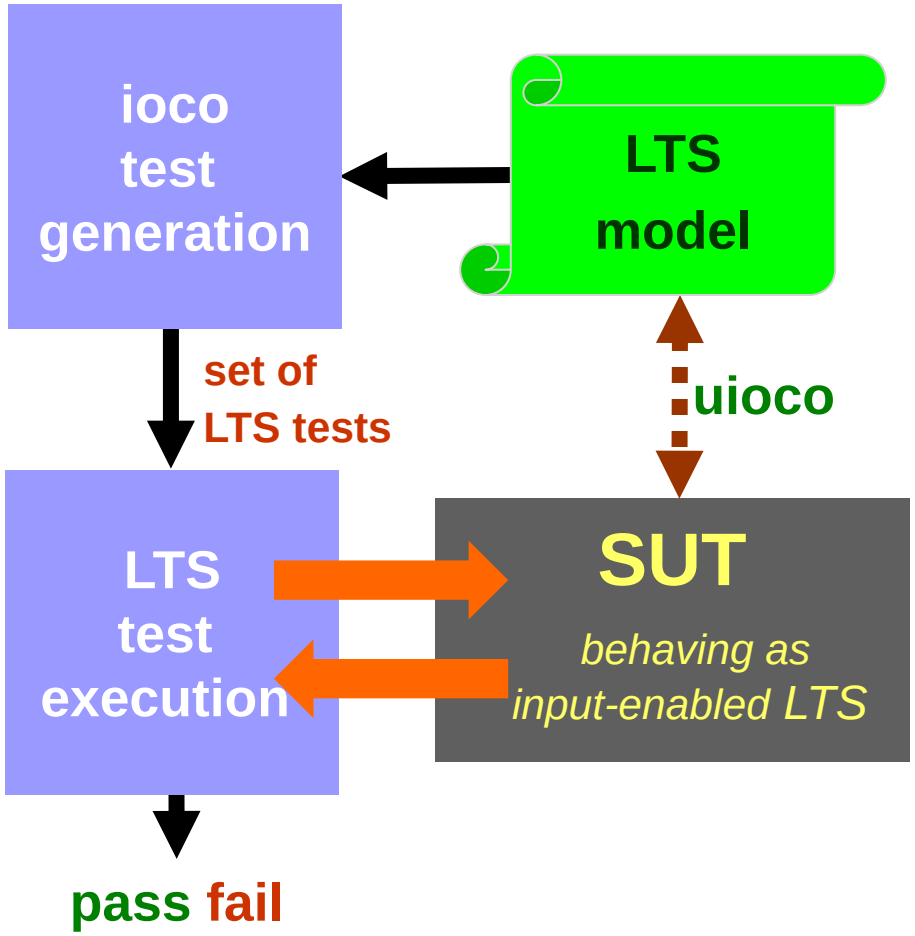
MBT : Labelled Transitions Systems



MBT with LTS topics:

- specification model
- implementation (SUT)
- implementation model
- conformance **uioco**
- test cases
- test generation
- test execution
- test result analysis
- sound & exhaustive

MBT : Labelled Transitions Systems



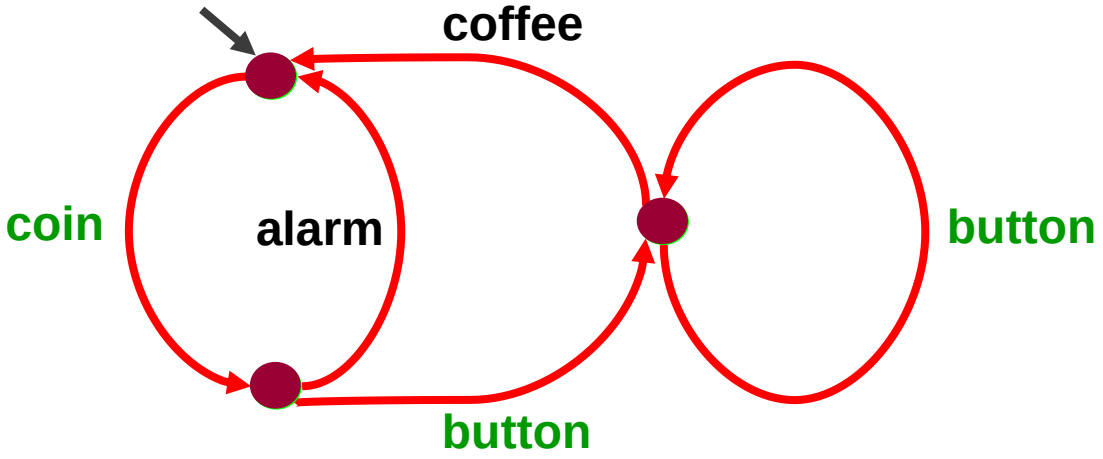
MBT with LTS topics:

- **specification model**
- implementation (SUT)
- implementation model
- conformance **uioco**
- test cases
- test generation
- test execution
- test result analysis
- sound & exhaustive

Models: Labelled Transition Systems

Labelled Transition System: $\langle\ S,\ L,\ T,\ s_0\ \rangle$

- S: set of states
- L: set of labels
- T: transitions, $T \subseteq S \times (L \cup \{\tau\}) \times S$
- s_0: initial state, $s_0 \in S$



Models: LTS with Inputs and Outputs

coin, button

from user to machine
initiative with user
machine cannot refuse

? inputs L_I

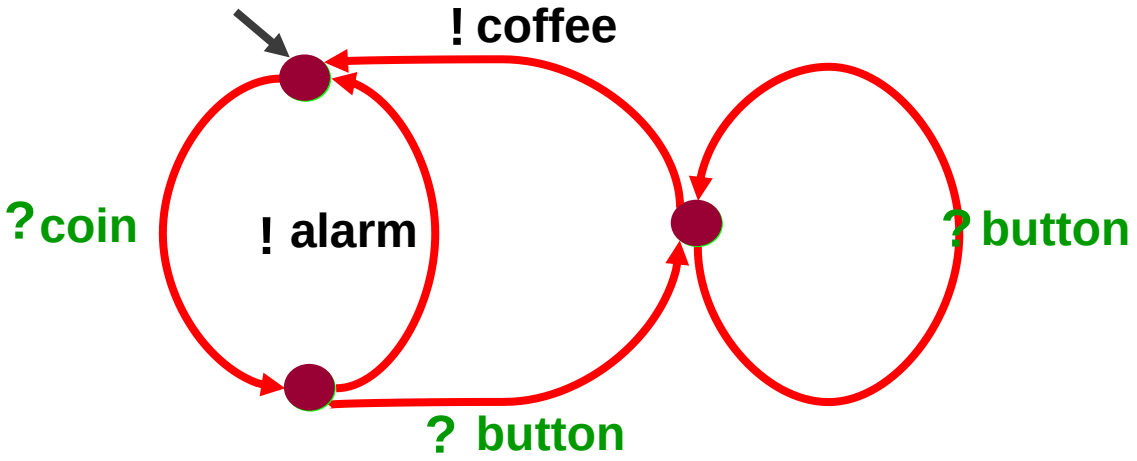
coffee, alarm

from machine to user
initiative with machine
user cannot refuse

! outputs L_U

$L_I \cap L_U = \varnothing$

$L_I \cup L_U = L$



Models: LTS with Inputs and Outputs

Labelled Transition System: $\langle\ S,\ L_I,\ L_U,\ T,\ s_0\ \rangle$

states

input labels

output labels

transitions

$T \subseteq S \times (L \cup \{\tau\}) \times S$

initial state

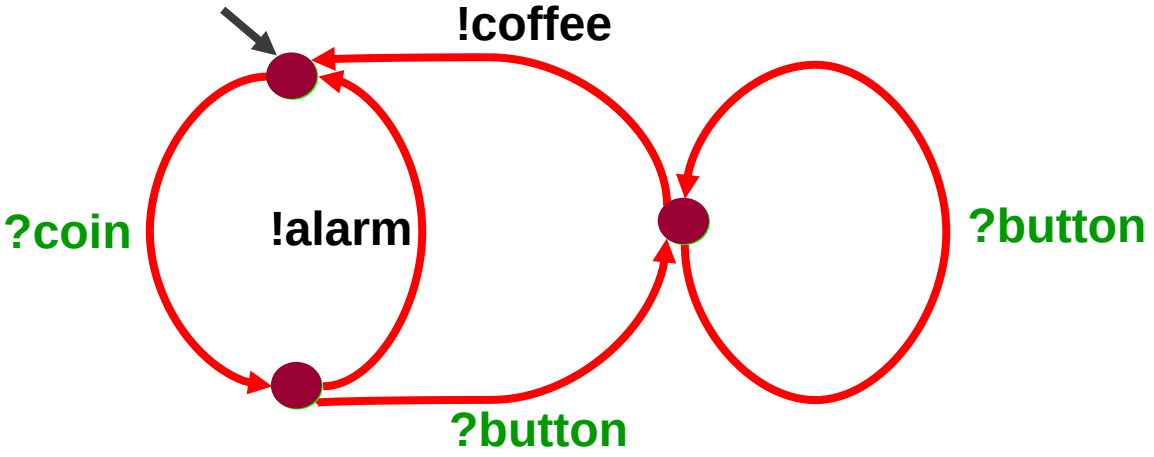
$s_0 \in S$

? = input

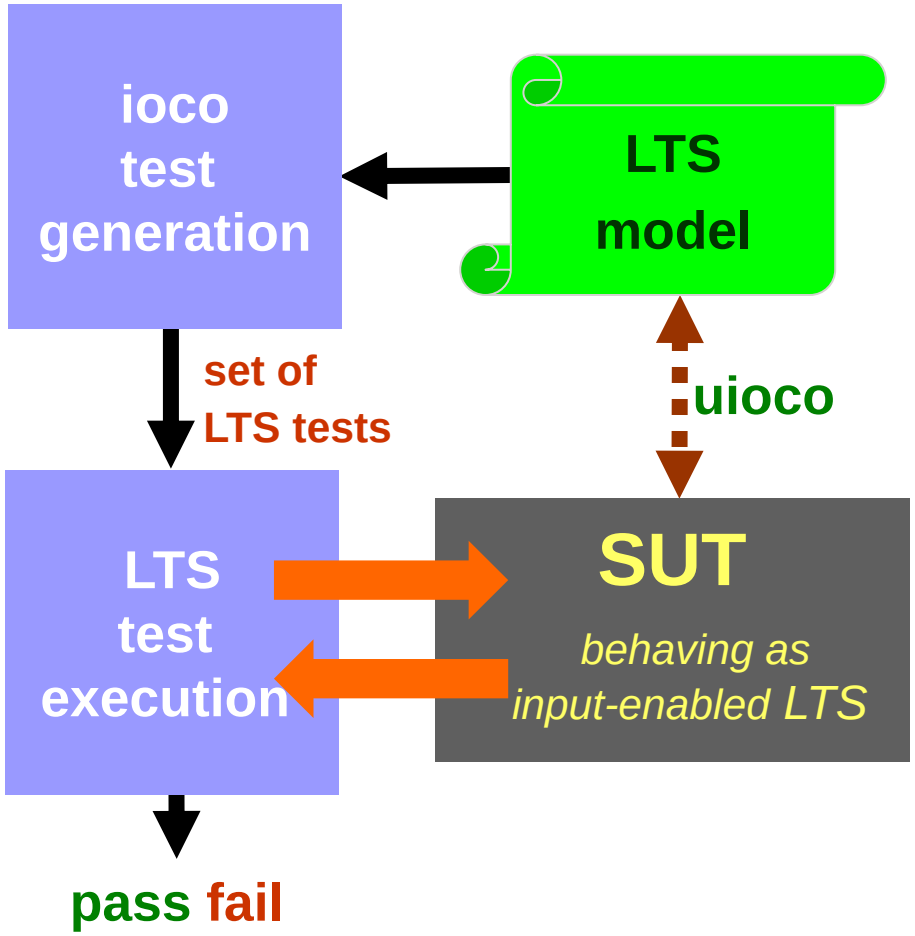
! = output

$L = L_I \cup L_U$

$L_I \cap L_U = \emptyset$



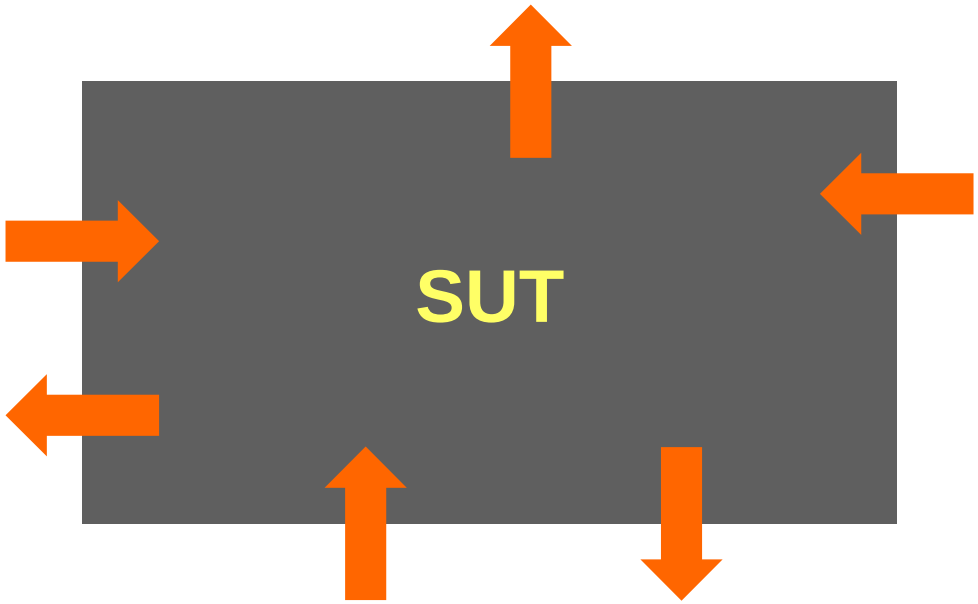
MBT : Labelled Transitions Systems



MBT with LTS topics:

- specification model
- implementation (SUT)
- implementation model
- conformance **uioco**
- test cases
- test generation
- test execution
- test result analysis
- sound & exhaustive

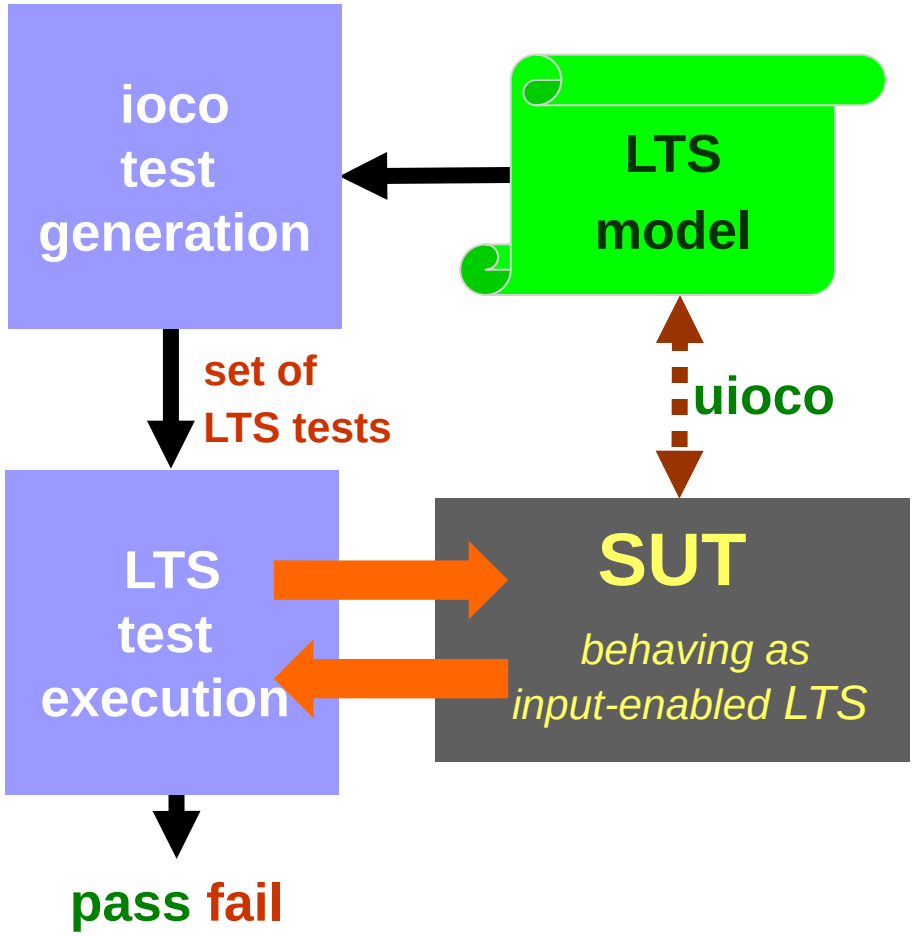
MBT : System Modelling



SUT

- black box
- inputs on interfaces
- outputs on interfaces

MBT : Labelled Transitions Systems



MBT with LTS topics:

- specification model
- implementation (SUT)
- implementation model
- conformance **uioco**
- test cases
- test generation
- test execution
- test result analysis
- sound & exhaustive

Models: Input-Output Transition Systems

In many systems, inputs are always enabled:

input-enabled *transition systems*

= *Input-Output Transition Systems*

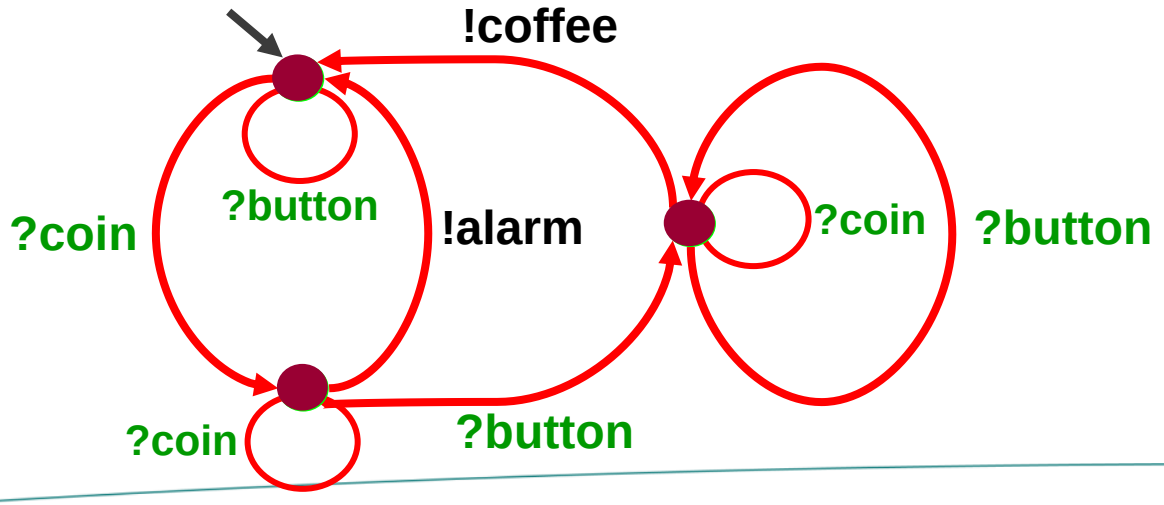
$IOTS(L_I, L_U) \subseteq LTS(L_I \cup L_U)$

input enabled:

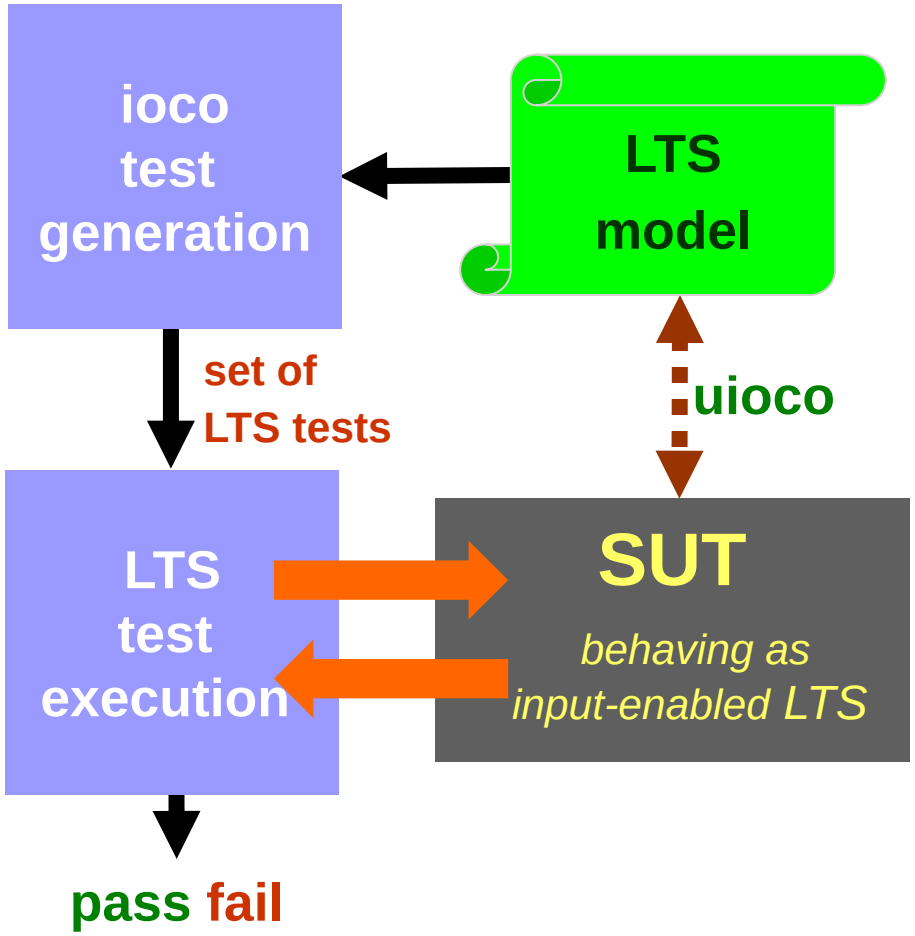
for all states s,

for all inputs $?a \in L_I$:

$s \xRightarrow{?a}$



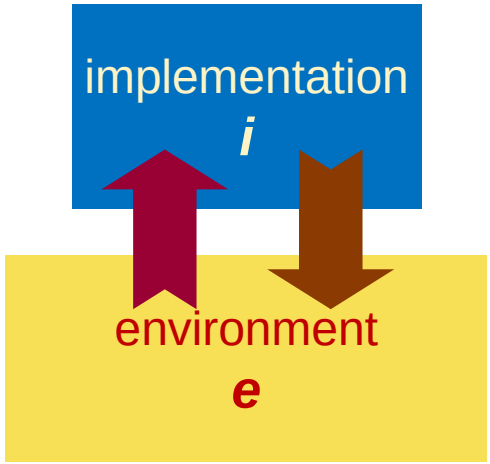
MBT : Labelled Transitions Systems



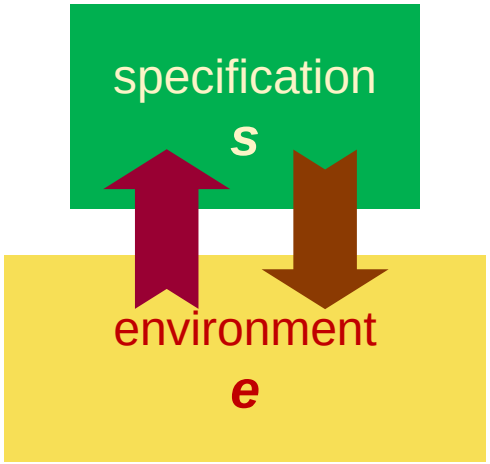
MBT with LTS topics:

- specification model
- implementation (SUT)
- implementation model
- conformance **uioco**
- test cases
- test generation
- test execution
- test result analysis
- sound & exhaustive

Implementation Relations for Input-Output Transition Systems



imp



$i \in IOTS(L_I, L_U)$ $\quad s \in LTS(L_I, L_U)$

$$\mathbf{imp} \subseteq IOTS(L_I, L_U) \times LTS(L_I, L_U)$$

i* imp *s

i is a conforming implementation of **s**

Input/Output Conformance : *uioco*

$$i \text{ uioco } s \quad =_{def} \quad \forall \sigma \in \textit{Utraces}(s) : \textit{out}(i \text{ after } \sigma) \subseteq \textit{out}(s \text{ after } \sigma)$$

s is a Labelled Transition System

i is (assumed to be) an input-enabled LTS

Alternative: **ioco**:

(*see Lecture Notes*)

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

Input/Output Conformance : *uioco*

i **uioco** s $=_{def}$ $\forall \sigma \in$ ***Utraces*** (s) : ***out*** $(i \text{ after } \sigma) \subseteq$ ***out*** $(s \text{ after } \sigma)$

s is a Labelled Transition System

i is (assumed to be) an input-enabled LTS

$p \xrightarrow{\delta} p \iff \forall\ !x \in L_U \cup \{\tau\}\ .\ p \not\xrightarrow{!x} \iff$ **p refuses** L_U

$$Straces(s) = \{\ \sigma \in (L \cup \{\delta\})^* \mid s \stackrel{\sigma}{\Longrightarrow}\ \}$$

$$Utraces(s) = \{\ \sigma \in Straces(s) \mid \forall\ \sigma_1\ ?b\ \sigma_2 = \sigma : not(s \text{ after } \sigma_1 \text{ refuses } \{?b\})\ \}$$

$$out(P) = \{\ !x \in L_U \mid \exists\ p \in P : p \xrightarrow{!x}\ \} \cup \{\ \delta \mid \exists\ p \in P : p \xrightarrow{\delta} p\ \}$$

Input/Output Conformance : *uioco*

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**

s is a Labelled Transition System

i is (assumed to be) an input-enabled LTS

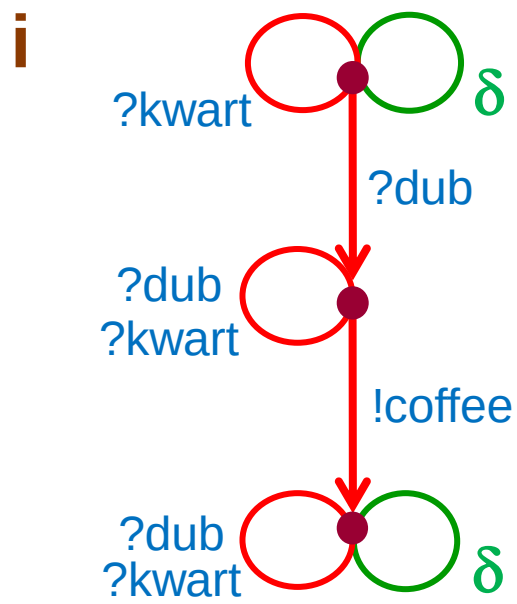
Intuition:

i uioco-conforms to **s**, iff

- if **i** produces output **x** after *U*-trace σ,
  then **s** can produce **x** after σ
- if **i** cannot produce any output after *U*-trace σ,
  then **s** cannot produce any output after σ (called *quiescence* δ)

Implementation Relation uioco

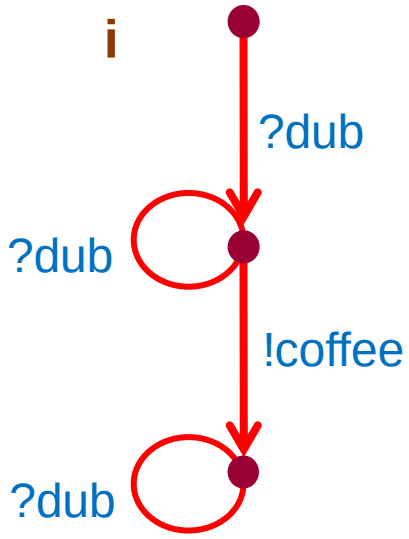
$$i \text{ uioco } s =_{def} \forall \sigma \in Utraces(s) : out(i \text{ after } \sigma) \subseteq out(s \text{ after } \sigma)$$



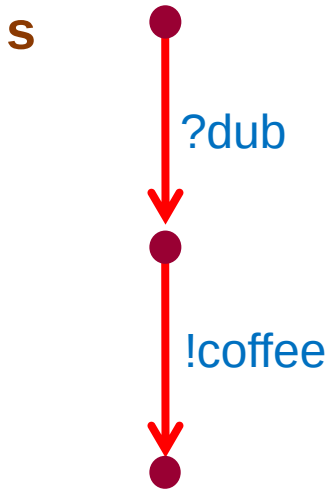
out (i after ε)	=	{ δ }
out (i after ?dub)	=	{ !coffee }
out (i after ?dub ?dub)	=	{ !coffee }
out (i after ?dub !coffee)	=	{ δ }
out (i after ?kwart)	=	{ δ }
out (i after !coffee)	=	∅
out (i after ?dub !tea)	=	∅
out (i after δ)	=	{ δ }
out (i after δ δ ?dub)	=	{ !coffee }

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



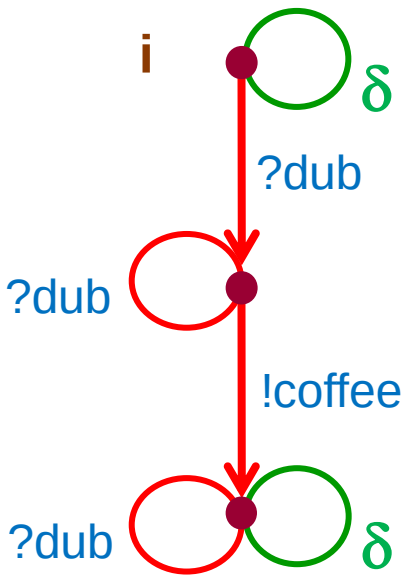
uioco



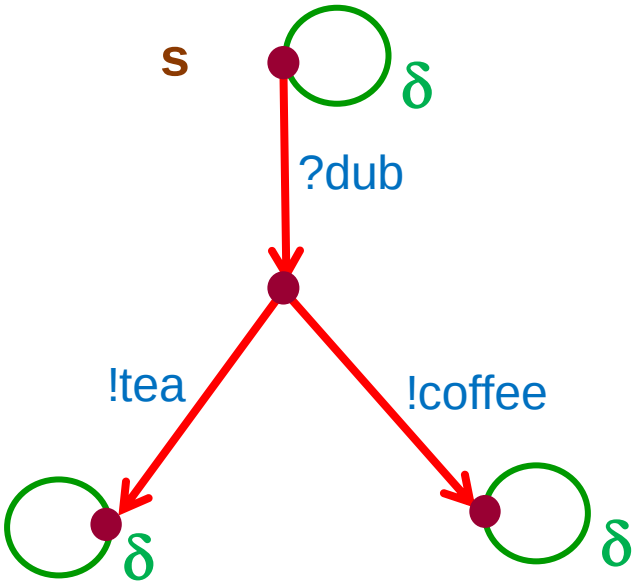
out **(i after** ε **)**	= { δ }	***out*** **(s after** ε **)**	= { δ }
out **(i after** ?dub **)**	= { !coffee }	***out*** **(s after** ?dub **)**	= { !coffee }
out **(i after** ?dub !coffee**)**	= { δ }	***out*** **(s after** ?dub !coffee**)**	= { δ }

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



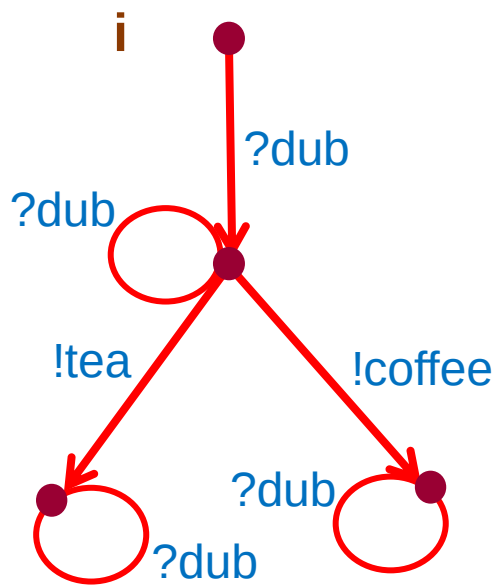
uioco



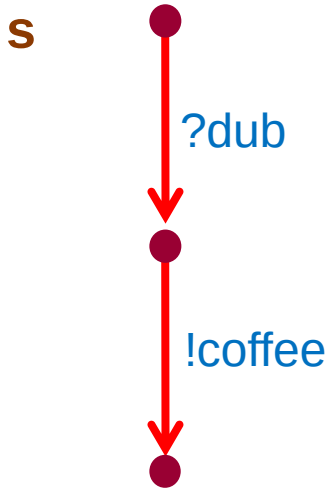
out **(i after** ?dub) = { !coffee } $\subseteq$ ***out*** **(s after** ?dub) = { !coffee. !tea }

Implementation Relation uioco

$$i \text{ uioco } s =_{def} \forall \sigma \in Utraces(s) : out(i \text{ after } \sigma) \subseteq out(s \text{ after } \sigma)$$



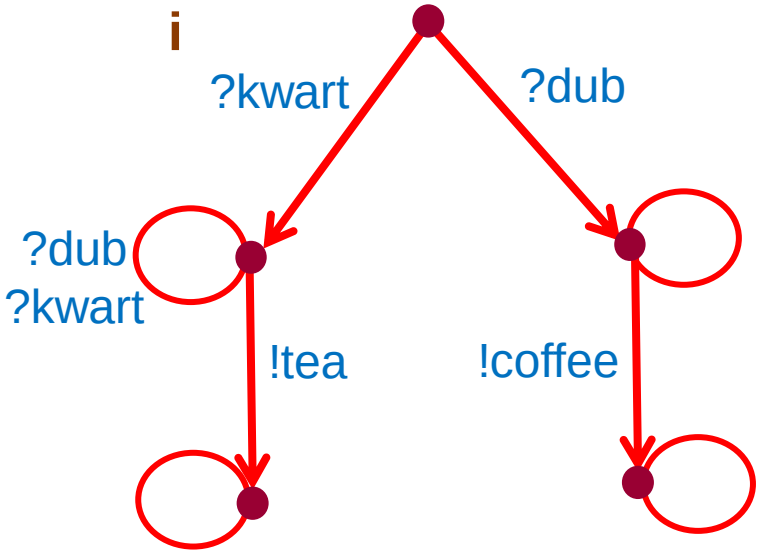
uioco



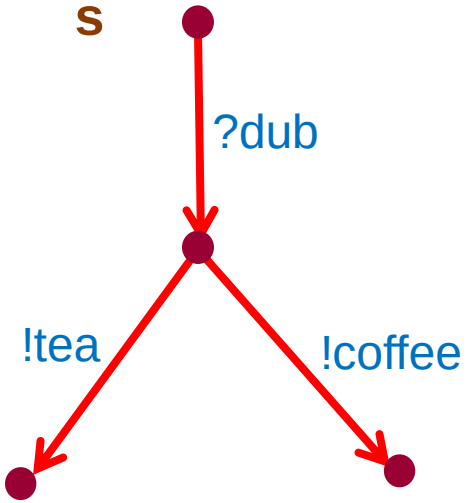
$$out(i \text{ after } ?dub) = \{ !coffee, !tea \} \not\subseteq out(s \text{ after } ?dub) = \{ !coffee \}$$

Implementation Relation uioco

$$i \text{ uioco } s =_{def} \forall \sigma \in Utraces(s) : out(i \text{ after } \sigma) \subseteq out(s \text{ after } \sigma)$$



uioco



out (**i** **after** ?dub) = { !coffee }

out (**i** **after** ?kwart) = { !tea } $\notin$

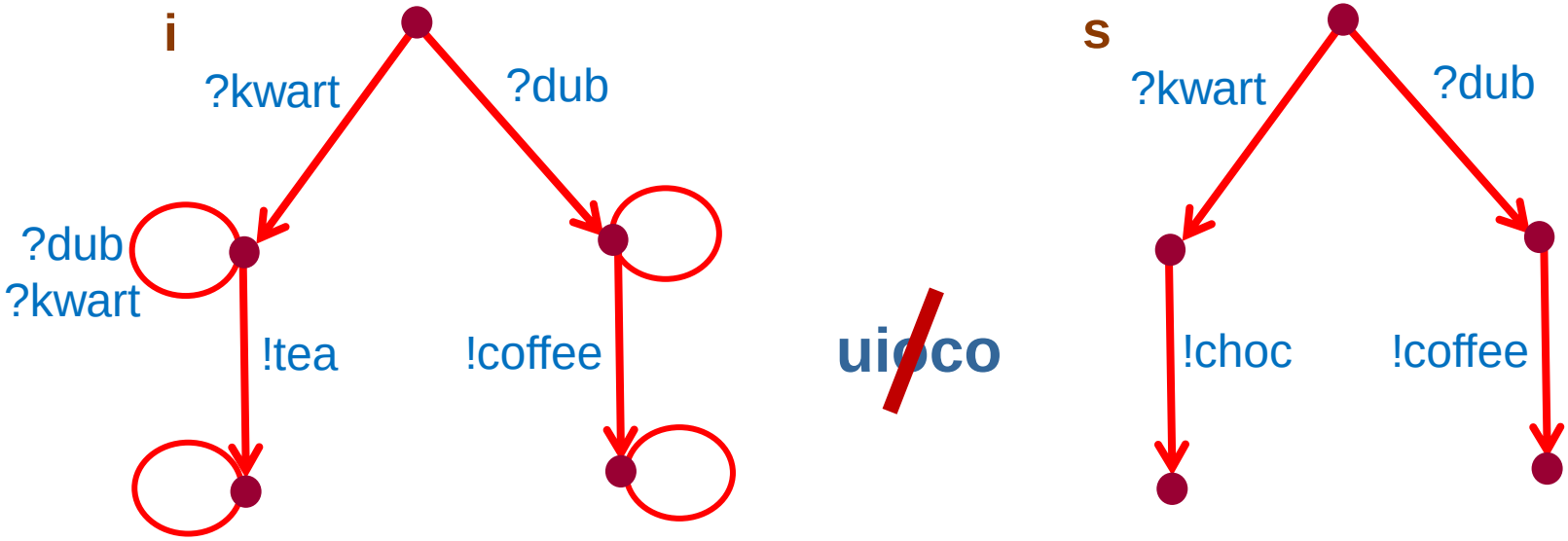
out (**s** **after** ?dub) = { !coffee, !tea }

out (**s** **after** ?kwart) = $\varnothing$

but ?kwart $\notin$ ***Utraces*** **(s)**

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ *Utraces* (s) : *out* (i after σ) $\subseteq$ *out* (s after σ)



out (**i after** ?dub) = { !coffee }

out (**i after** ?kwart) = { !tea }

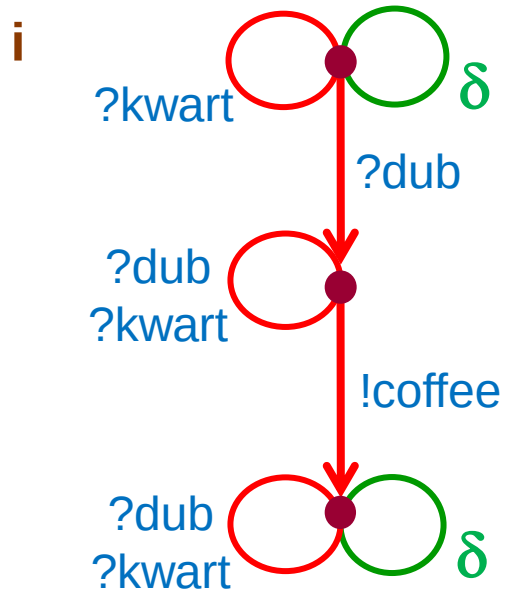
$\not\subseteq$

out (**s after** ?dub) = { !coffee }

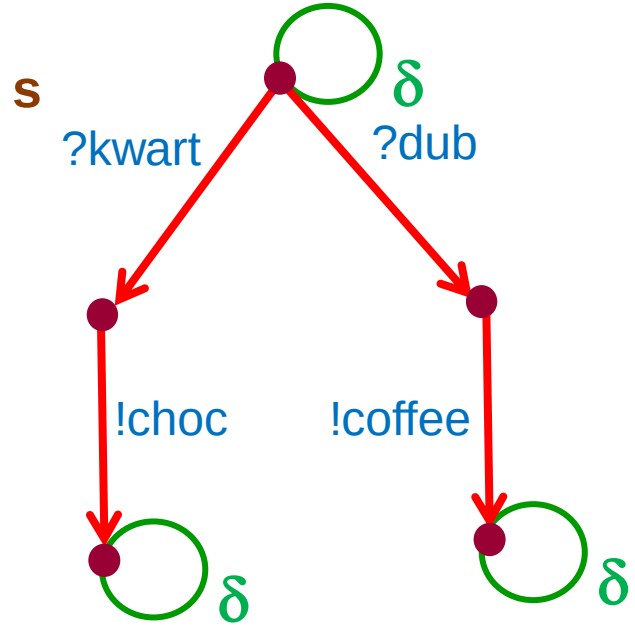
out (**s after** ?kwart) = { !choc }

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



uioco



out **(i after** ?dub **)** = { !coffee }

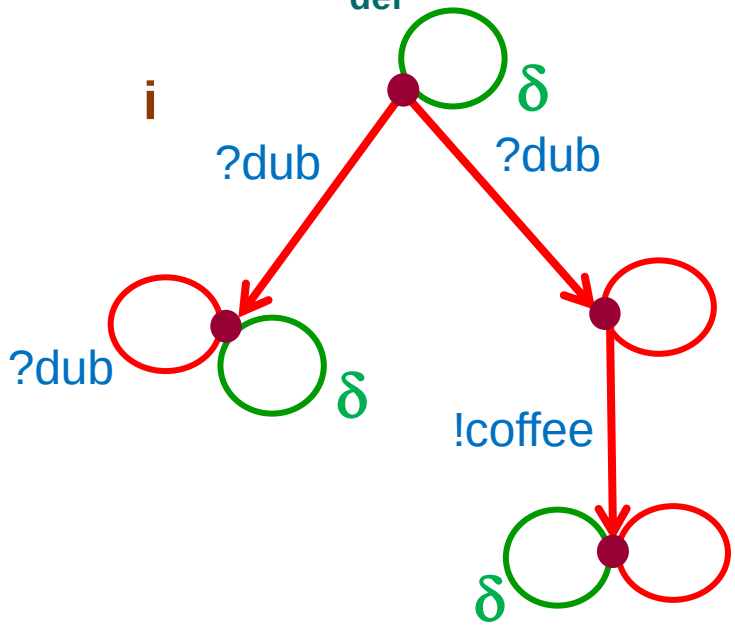
out **(i after** ?kwart **)** = { δ } $\notin$

out **(s after** ?dub **)** = { !coffee }

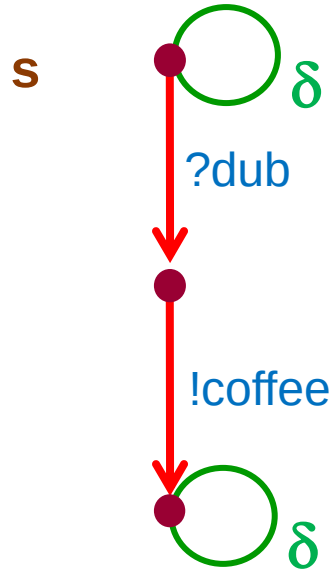
out **(s after** ?kwart **)** = { !choc }

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



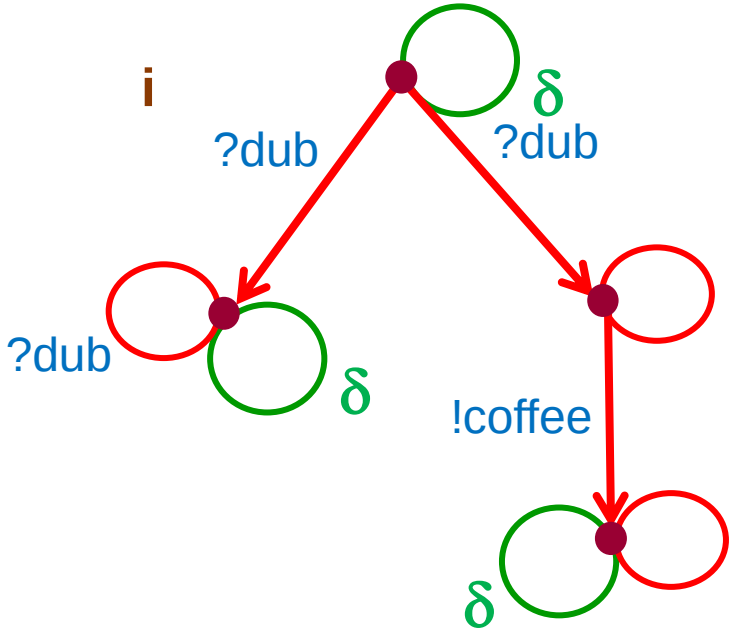
uioco



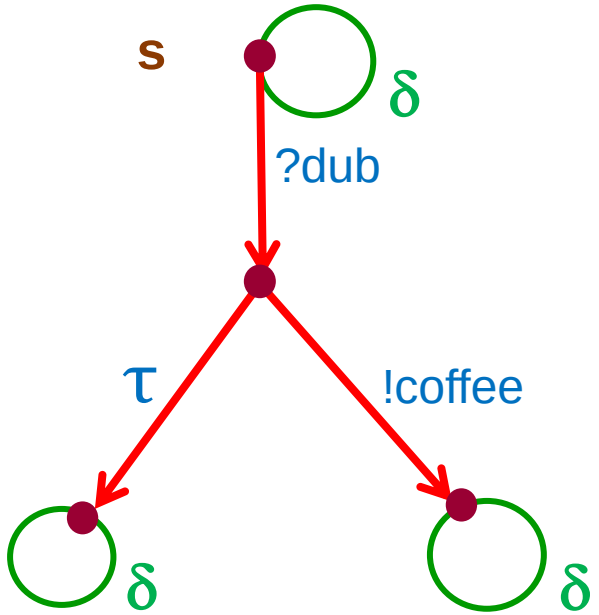
out **(i after** ?dub) = { !coffee, δ } $\not\subseteq$ ***out*** **(s after** ?dub) = { !coffee }

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



uioco

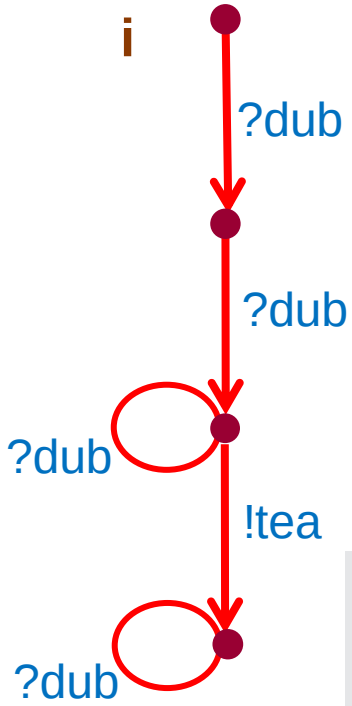


out **(i after** ?dub) = { !coffee, δ }

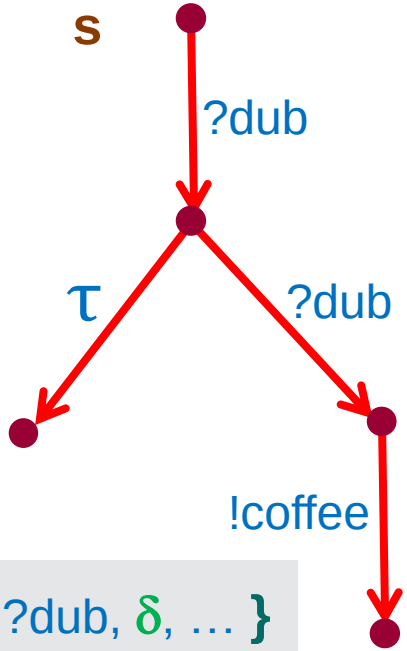
out **(s after** ?dub) = { !coffee, δ }

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



uioco



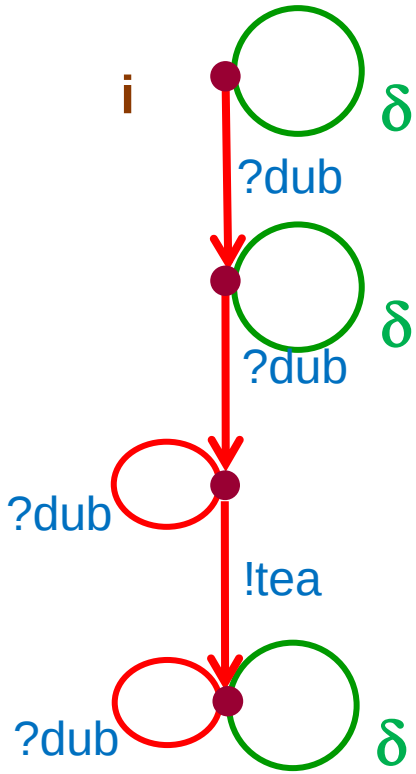
?dub ?dub $\notin$ ***Utraces*** **(s)** = **{** ε, ?dub, δ, … **}**

because **s after** ?dub **refuses {** ?dub **}**

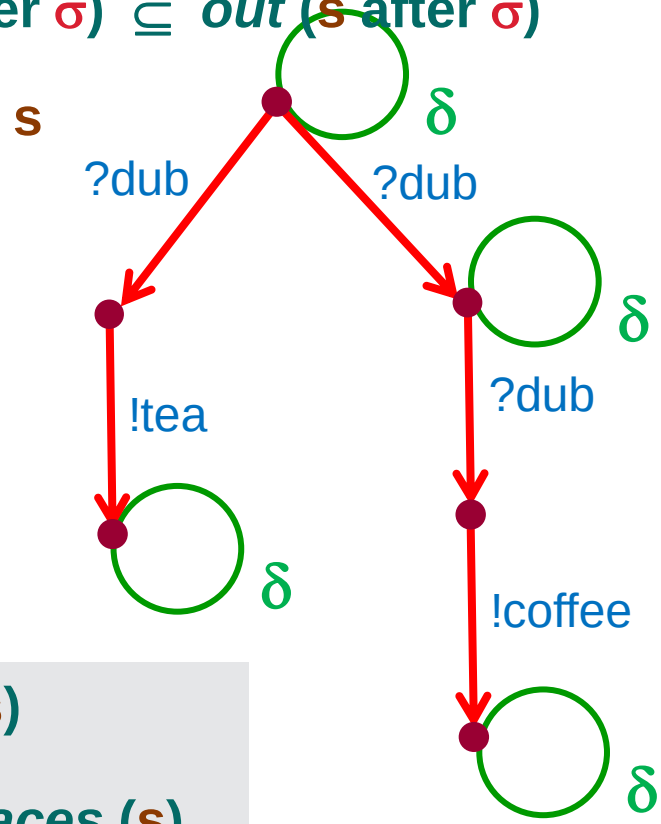
out **(i after** ?dub ?dub **)** = { !tea } $\notin$ ***out*** **(s after** ?dub ?dub **)** = { !coffee }

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



uioco



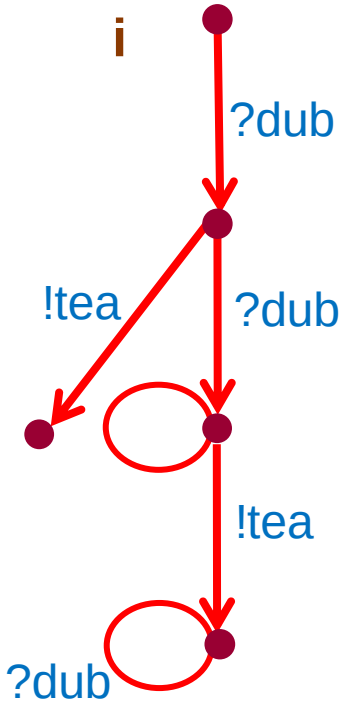
?dub ?dub $\notin$ ***Utraces*** **(s)**

but ?dub δ ?dub $\in$ ***Utraces*** **(s)**

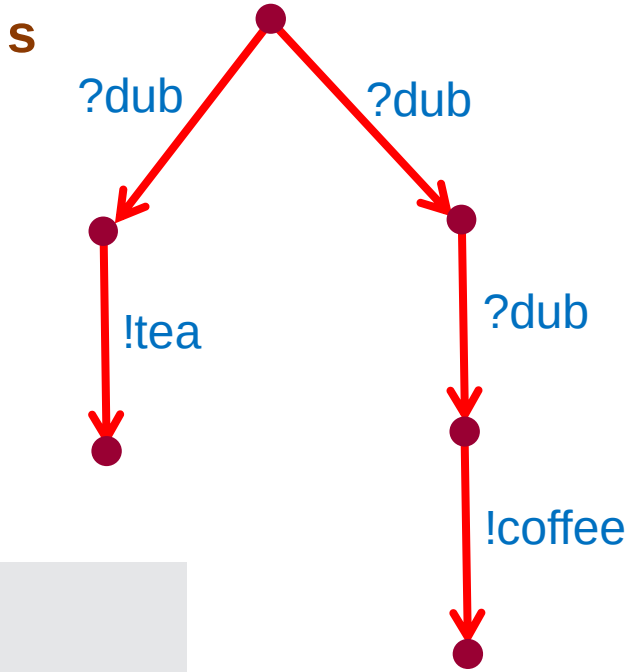
out **(i after** ?dub δ ?dub **)** = { !tea } $\not\subseteq$ ***out*** **(s after** ?dub δ ?dub **)** = { !coffee }

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



uioco



?dub ?dub $\notin$ ***Utraces*** **(s)**

but ?dub δ ?dub $\in$ ***Utraces*** **(s)**

out **(i after** ?dub δ ?dub **)** $= \varnothing$ $\subseteq$ ***out*** **(s after** ?dub δ ?dub **)** $=$ { !coffee }

Input/Output Conformance : *uioco*

i **uioco** s $=_{def}$ $\forall \sigma \in$ ***Utraces*** (s) : ***out*** $(i \text{ after } \sigma) \subseteq$ ***out*** $(s \text{ after } \sigma)$

s is a Labelled Transition System

i is (assumed to be) an input-enabled LTS

$p \xrightarrow{\delta} p \Leftrightarrow \forall\ !x \in L_U \cup \{\tau\}\ .\ p \not\xrightarrow{!x} \Leftrightarrow$ **p refuses** L_U

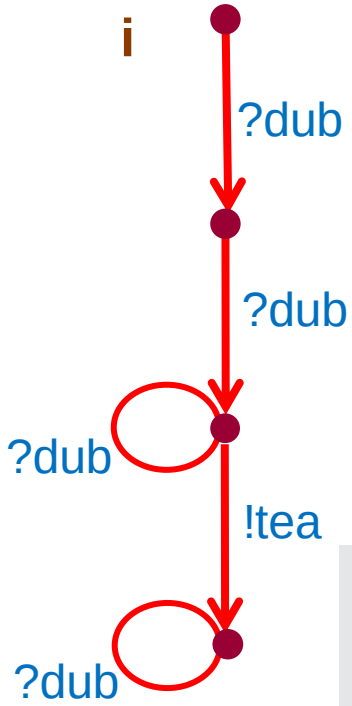
$$Straces(s) = \{\ \sigma \in (L \cup \{\delta\})^* \mid s \stackrel{\sigma}{\Longrightarrow}\ \}$$

$$Utraces(s) = \{\ \sigma \in Straces(s) \mid \forall\ \sigma_1\ ?b\ \sigma_2 = \sigma : not(s \text{ after } \sigma_1 \text{ refuses } \{?b\})\ \}$$

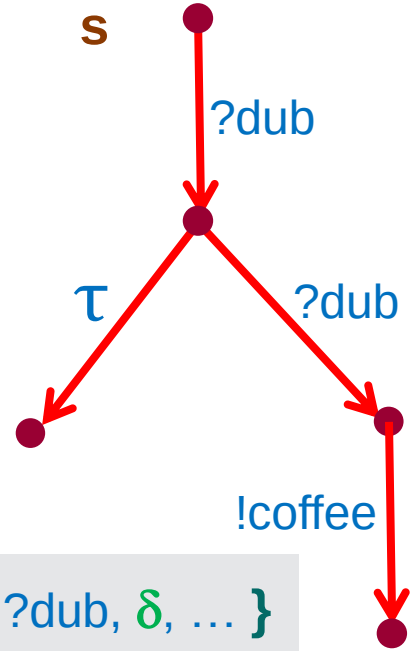
$$out(P) = \{\ !x \in L_U \mid \exists\ p \in P : p \xrightarrow{!x}\ \} \cup \{\ \delta \mid \exists\ p \in P : p \xrightarrow{\delta} p\ \}$$

Implementation Relation uioco

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



uioco



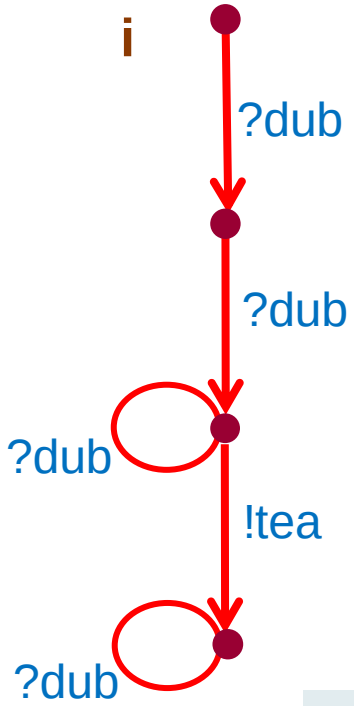
?dub ?dub $\notin$ ***Utraces*** **(s)** = **{** ε, ?dub, δ, … **}**

because **s after** ?dub **refuses {** ?dub **}**

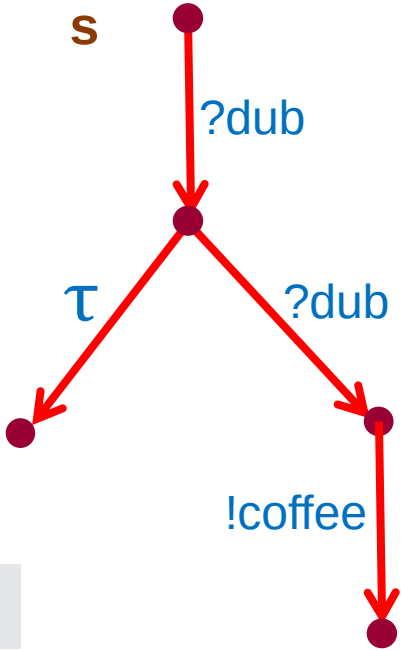
out **(i after** ?dub ?dub **)** = { !tea } $\notin$ ***out*** **(s after** ?dub ?dub **)** = { !coffee }

Alternative Implementation Relation ioco

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



ioco



?dub ?dub $\in$ ***Straces*** **(s)**

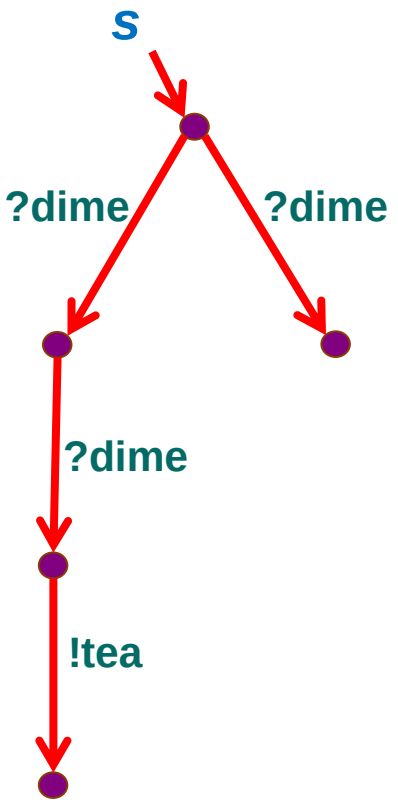
Straces (**s**) = { $\sigma \in (L \cup \{\delta\})^*$ | s $\stackrel{\sigma}{}$ }

out **(i after** ?dub ?dub) = { !tea } $\notin$ ***out*** **(s after** ?dub ?dub) = { !coffee }

Input/Output Conformance : (*u*)*ioco*

i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**

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



$$Straces(s) = \{ \sigma \in (L \cup \{\delta\})^* \mid s \stackrel{\sigma}{\Longrightarrow} \}$$

$$Utraces(s) = \{ \sigma \in Straces(s) \mid \forall \sigma_1\ ?b\ \sigma_2 = \sigma : not(s \text{ after } \sigma_1 \text{ refuses } \{?b\}) \}$$

?dime ?dime $\in$ ***Straces*** **(s)**

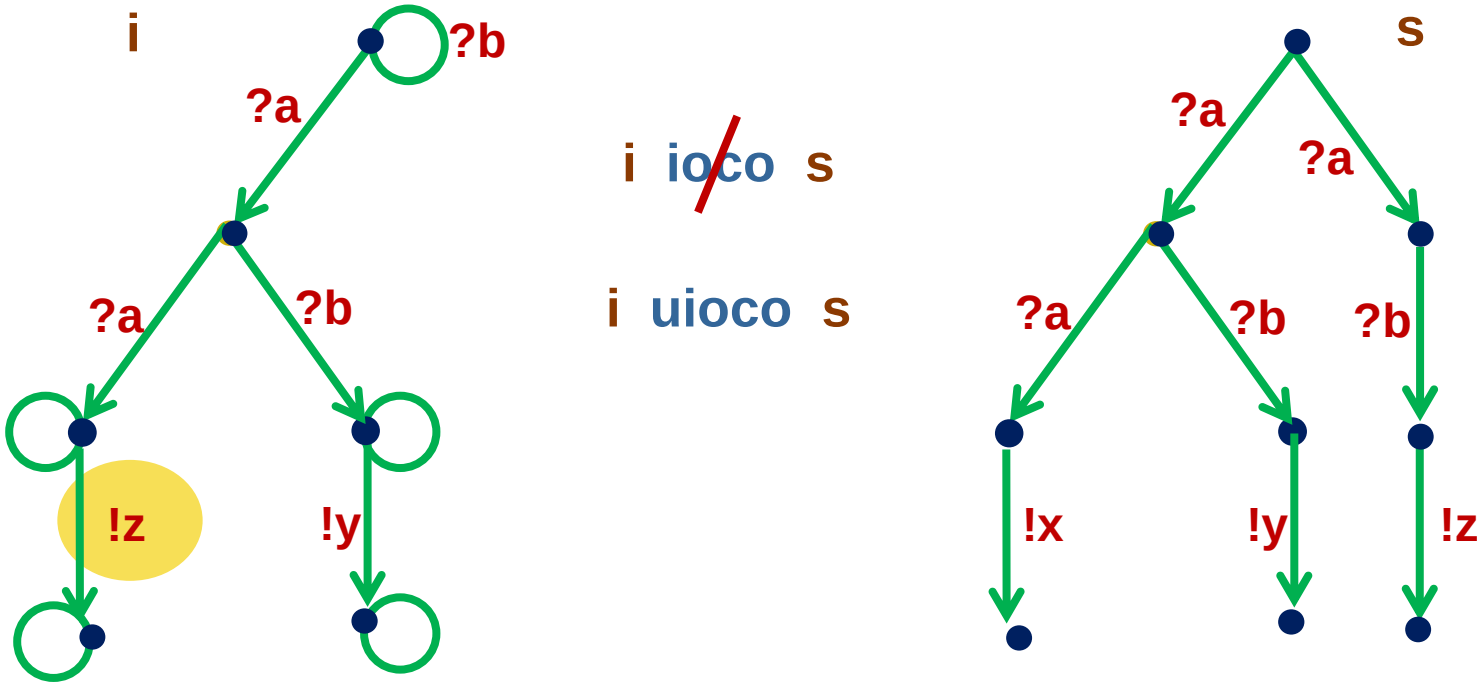
?dime ?dime $\notin$ ***Utraces*** **(s)**

ioco $\subset$ **uioco**

Input/Output Conformance : (*u*)*ioco*

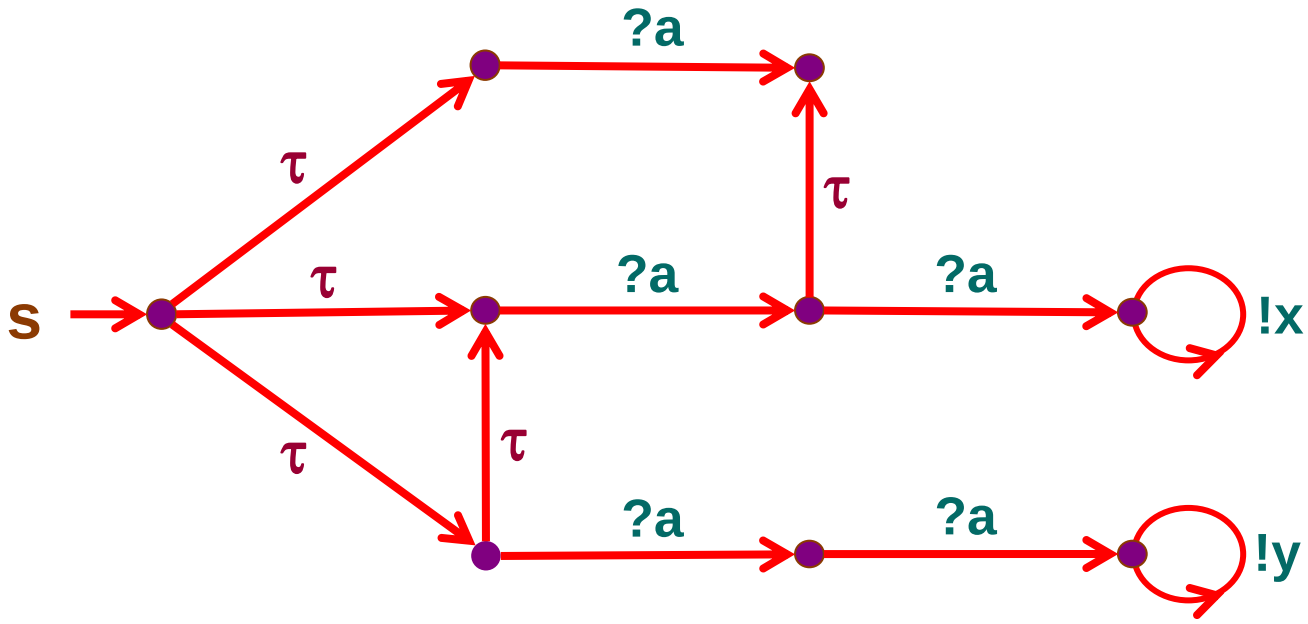
i uioco s $=_{def}$ $\forall \sigma \in$ ***Utraces*** **(s) :** ***out*** **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**

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



Difference **ioco-uioco** is in *non-deterministic under-specification*

A non-*ioco*-implementable specification

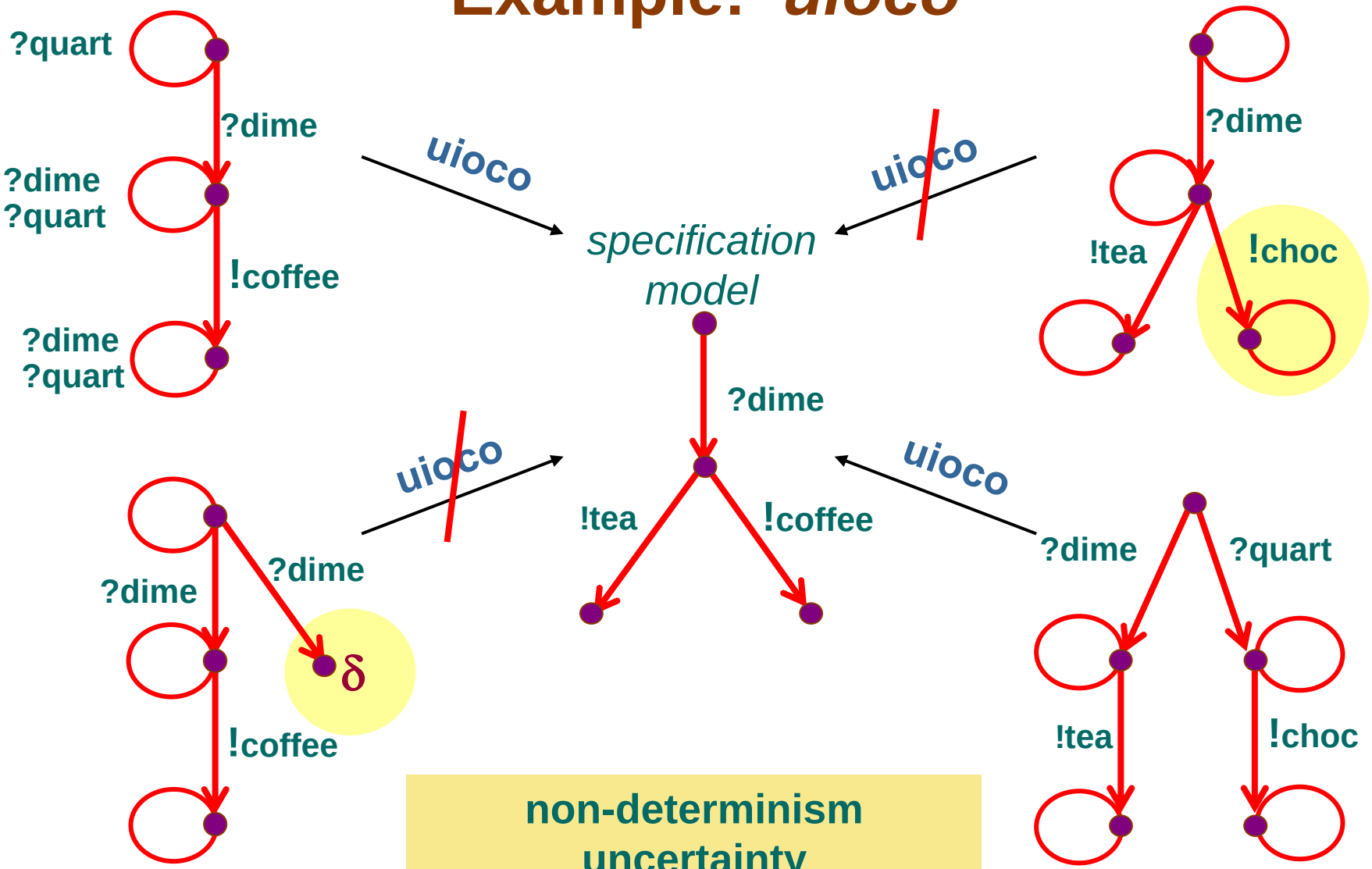


There is ***no*** implementation ***i*** ; what would be ***out*** **(*i* after δ ?a δ ?a)** ?

let *out* (***i*** after δ?aδ?a) = **X** then

- **X** ≠ ∅
- **X** ⊆ *out*(***i*** after δ?a?a) ⊆ *out*(***s*** after δ?a?a) = { !x }
- **X** ⊆ *out*(***i*** after ?aδ?a) ⊆ *out*(***s*** after ?aδ?a) = { !y }

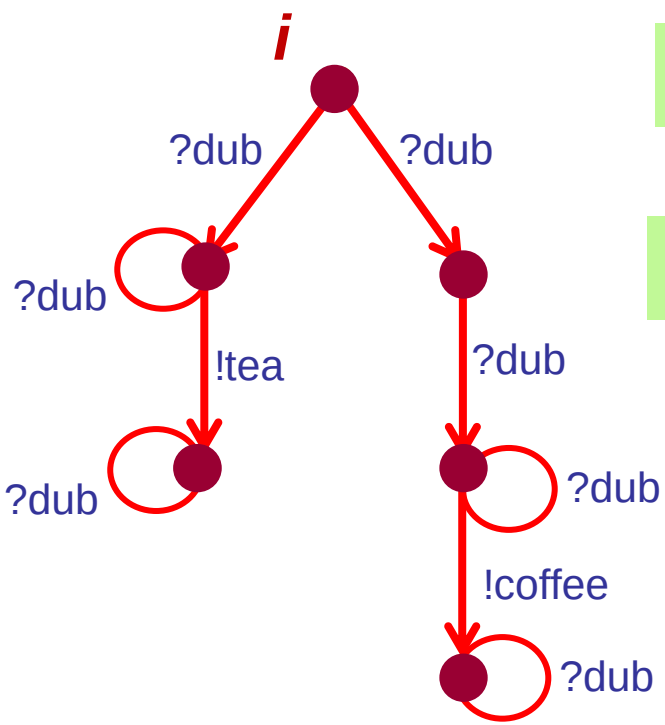
Example: *uioco*



non-determinism
uncertainty
under-specification

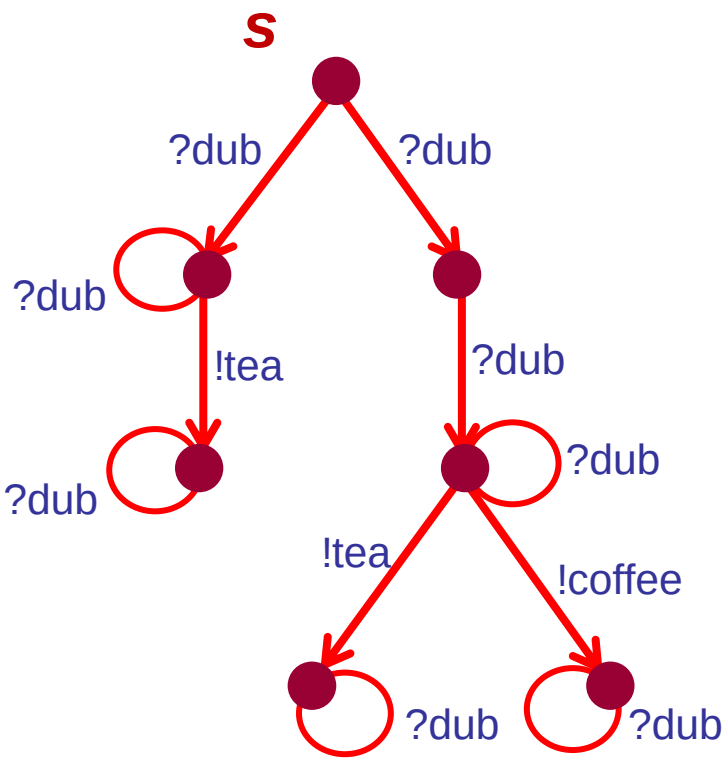
Example: *ioco*

i u/ioco s $=_{def}$
$\forall \sigma \in$ ***U/Straces*** **(s) :**
out **(i after** σ**)** $\subseteq$ ***out*** **(s after** σ**)**



i (u)ioco *s*

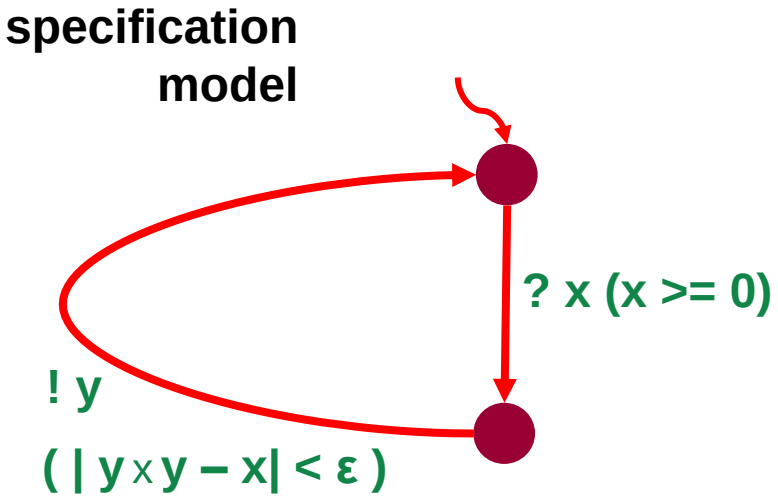
s (u)ioco *i*

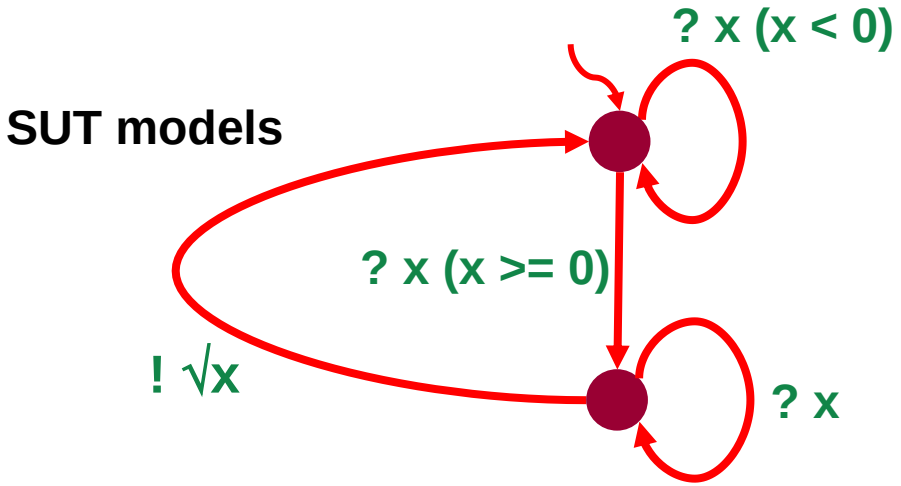


out (***i*** **after** ?dub.?dub) = *out* (***s*** **after** ?dub.?dub) = { !tea, !coffee }

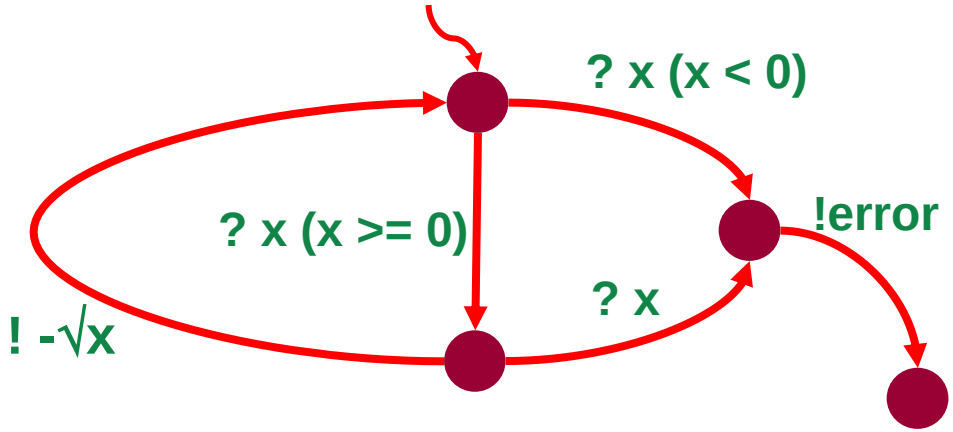
out (***i*** **after** ?dub.δ.?dub) = { !coffee } $\neq$ *out* (***s*** **after** ?dub.δ.?dub) = { !tea, !coffee }

MBT : Nondeterminism, Underspecification

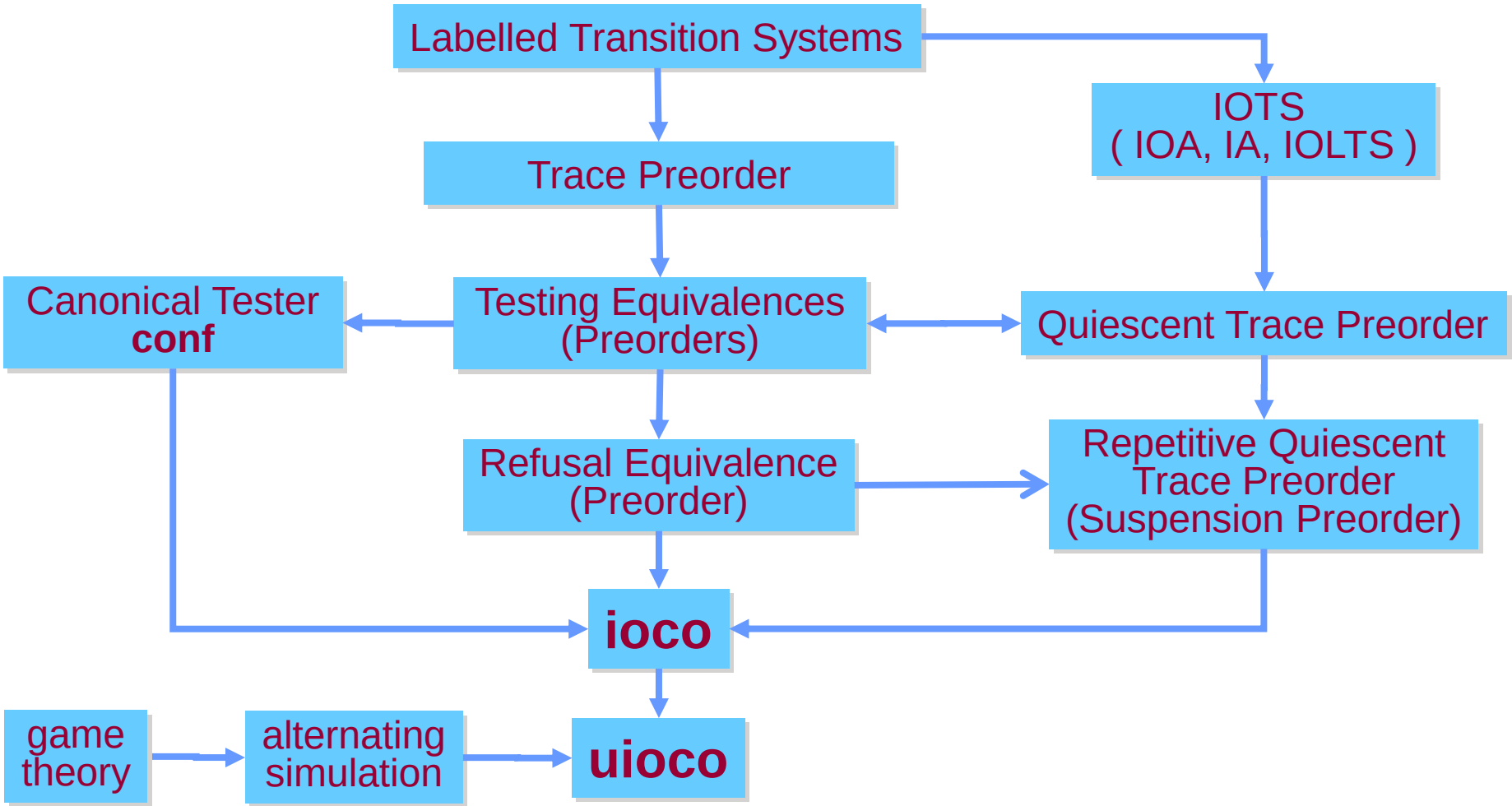




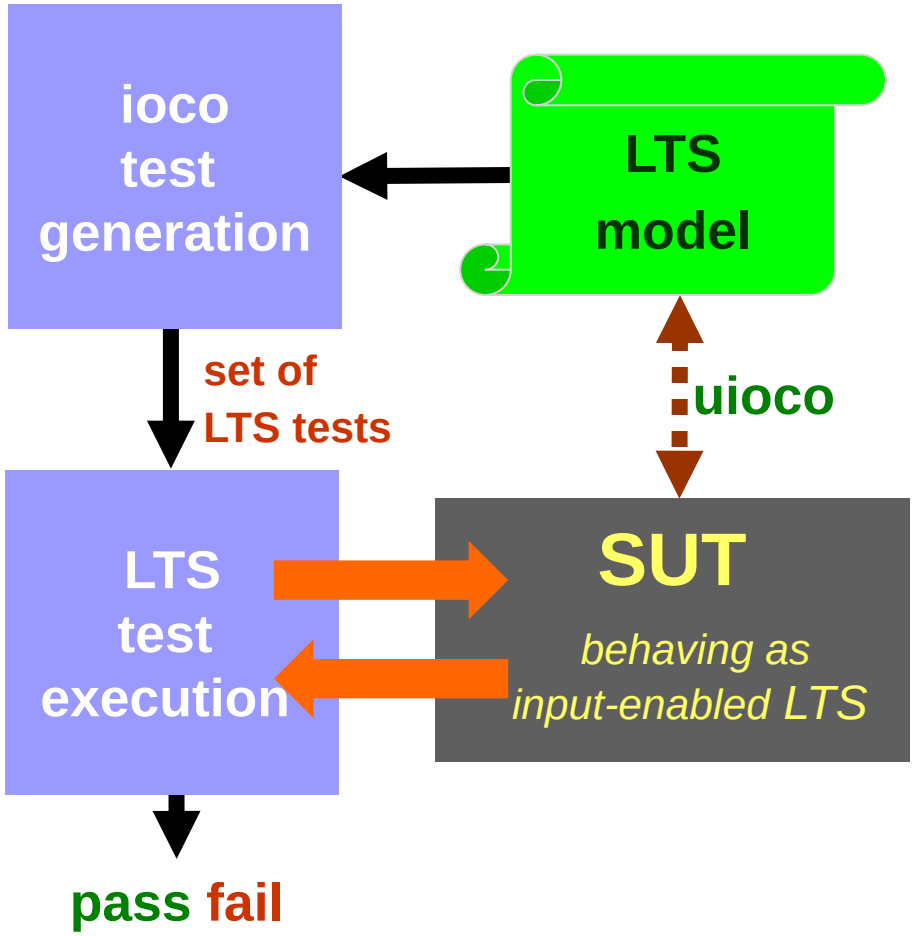
- non-determinism
- under-specification
- specification of properties rather than construction



Genealogy of u/ioco



MBT : Labelled Transitions Systems



MBT with LTS topics:

- specification model
- implementation (SUT)
- implementation model
- conformance **uioco**
- test cases
- test generation
- test execution
- test result analysis
- sound & exhaustive