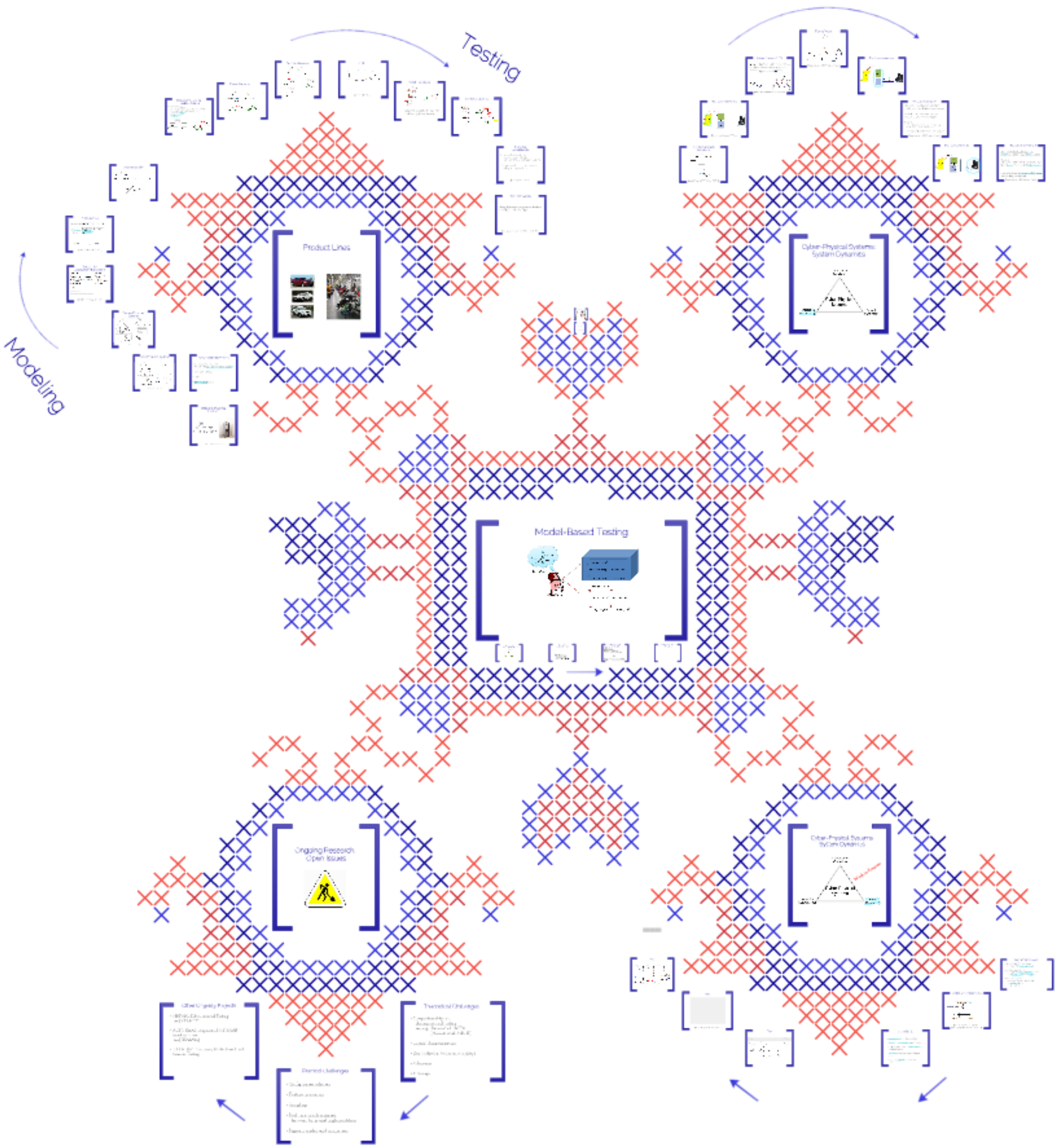


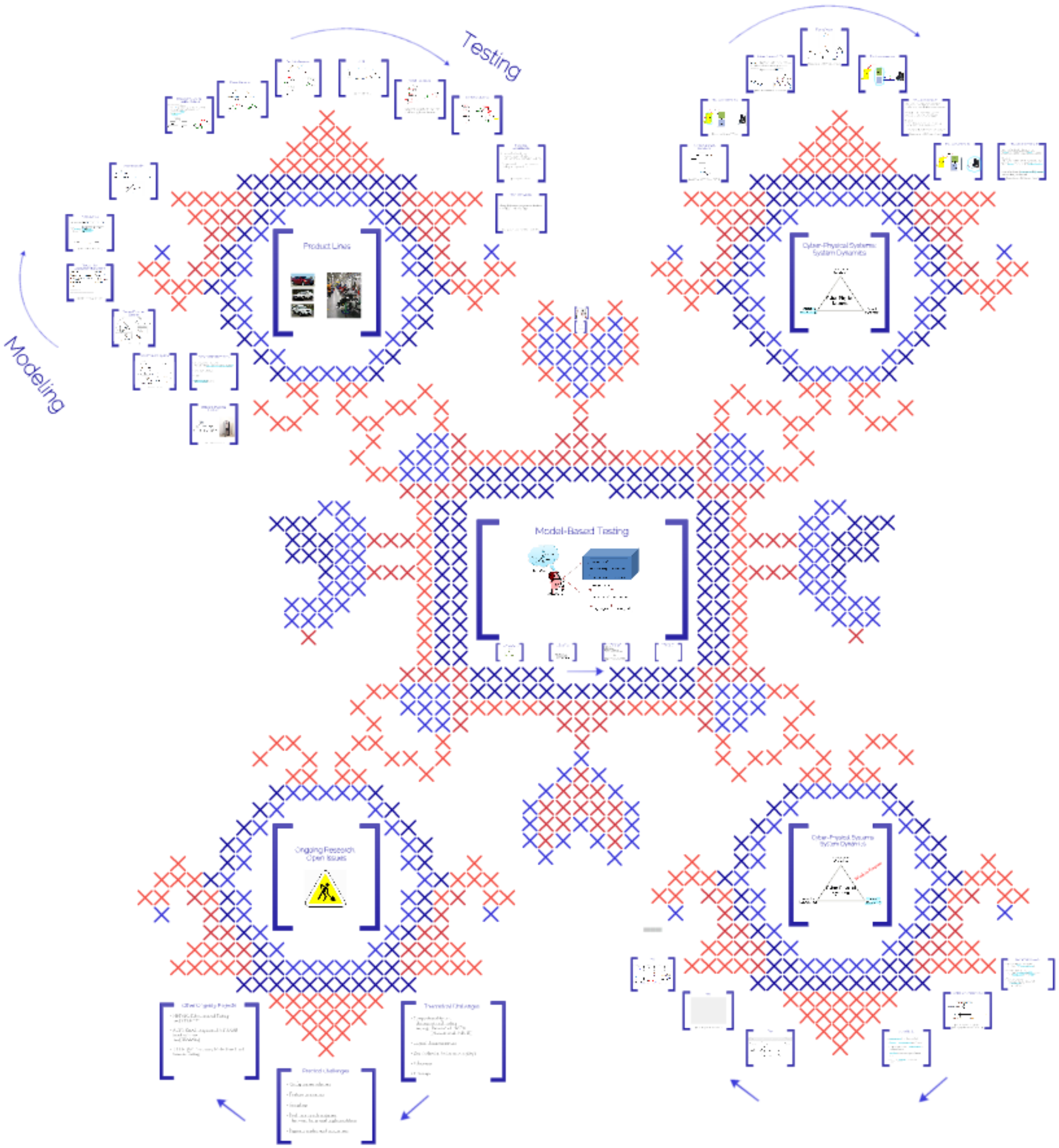
From Concurrency Theory to Model-Based Testing Cyber-Physical Systems

Mohammad Mousavi

HALMSTAD UNIVERSITY

KK-stiftelsen

Vetenskapsrådet



From Concurrency Theory to Model-Based Testing Cyber-Physical Systems

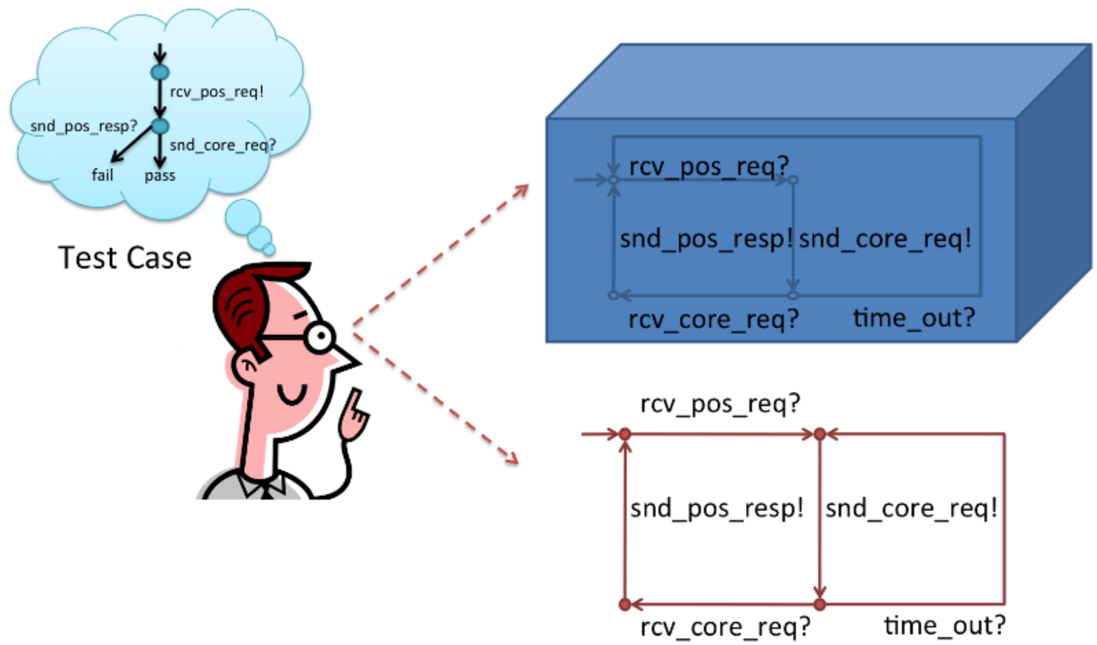
Mohammad Mousavi

HALMSTAD UNIVERSITY

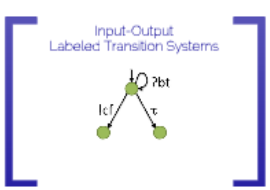
KK-stiftelsen

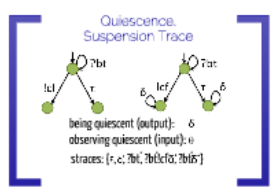
Vetenskapsrådet

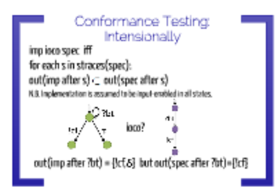
Model-Based Testing



13

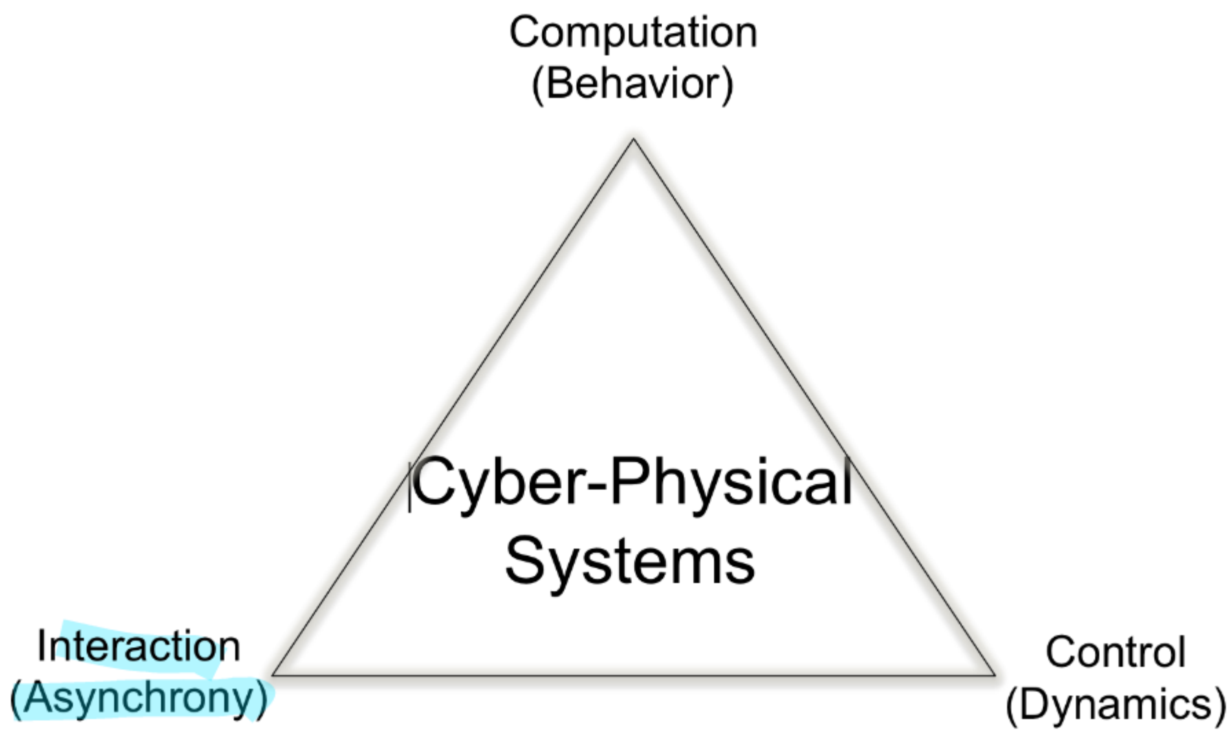




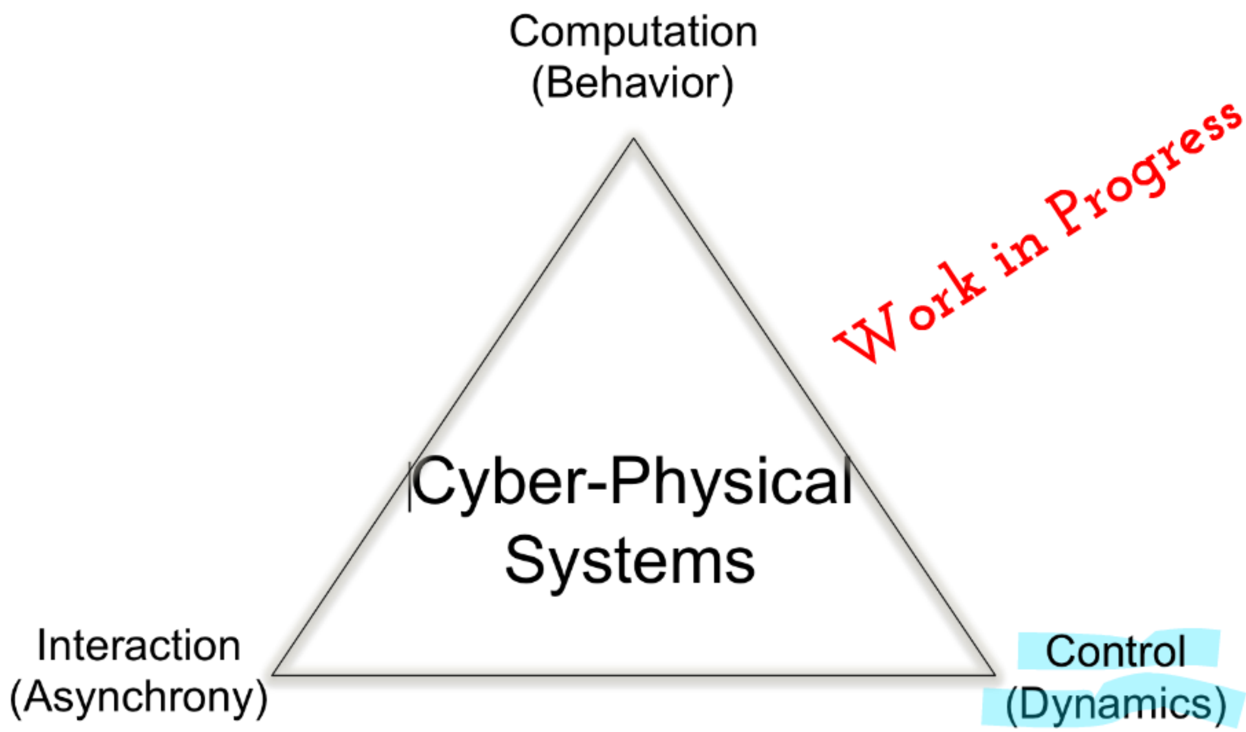


Product Lines

Cyber-Physical Systems: System Dynamics

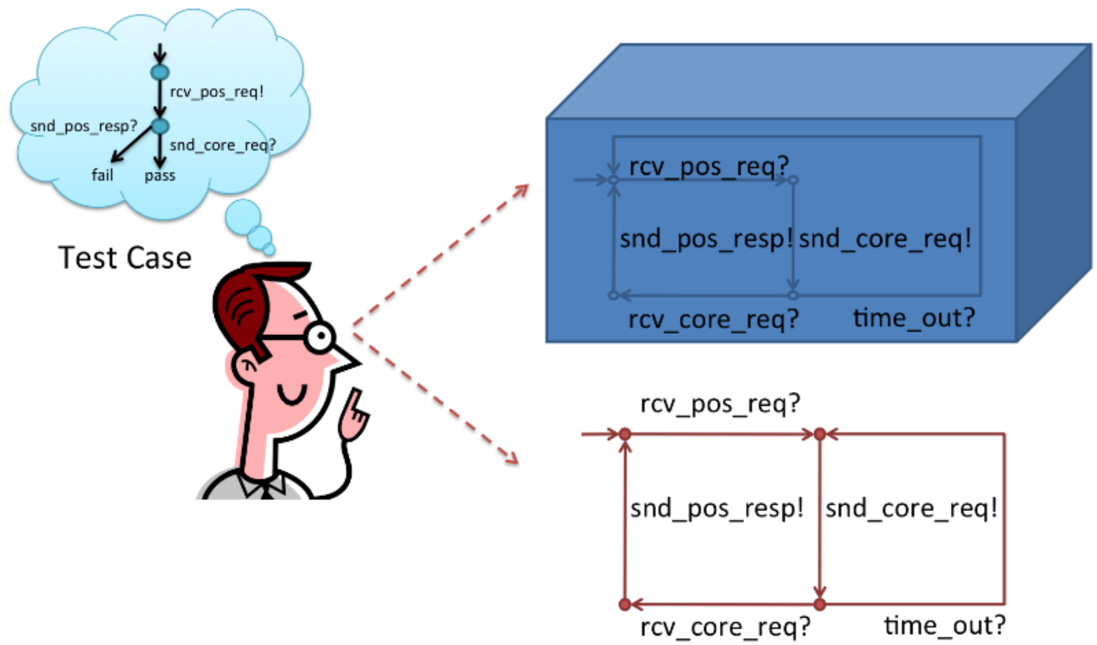


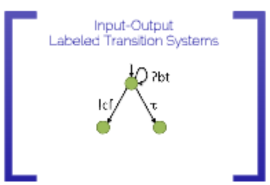
Cyber-Physical Systems: System Dynamics

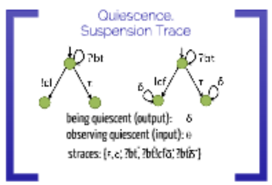


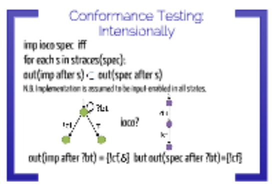
Ongoing Research, Open Issues

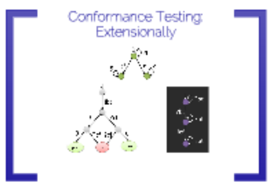
Model-Based Testing



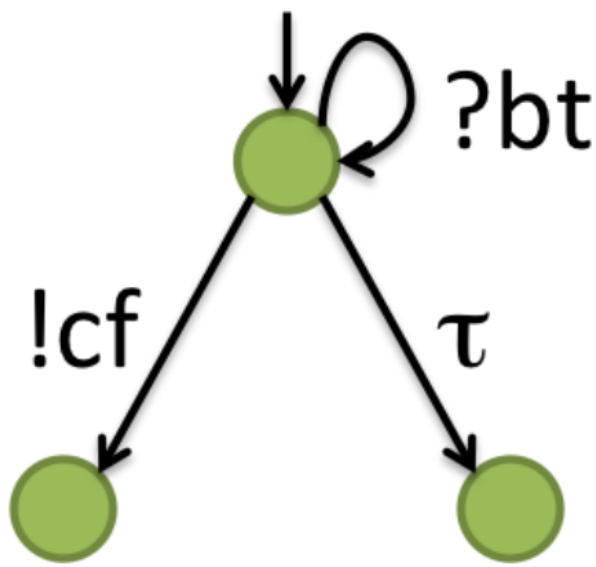




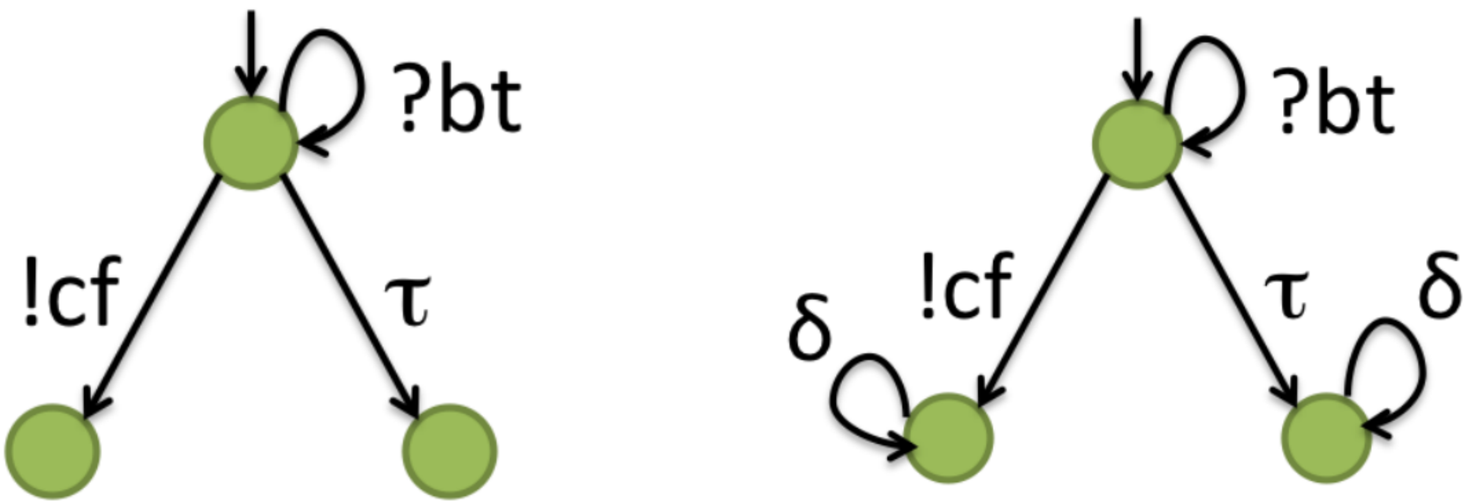




Input-Output Labeled Transition Systems



Quiescence, Suspension Trace



being quiescent (output): δ

observing quiescent (input): θ

straces: $\{\varepsilon, \delta^*, ?bt^*, ?bt^*!cf\delta^*, ?bt^*\delta^*\}$

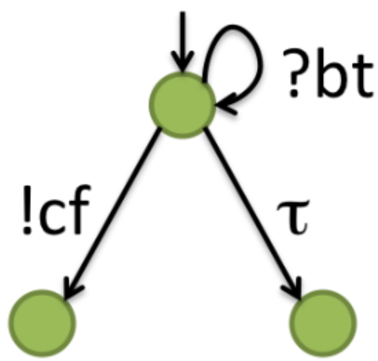
Conformance Testing: Intensionally

imp ioco spec iff

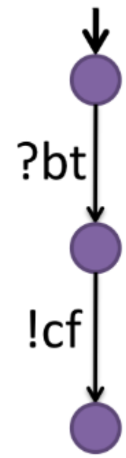
for each s in straces(spec):

out(imp after s) $\subseteq$ out(spec after s)

N.B. Implementation is assumed to be input-enabled in all states.

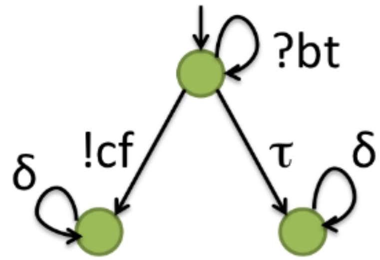


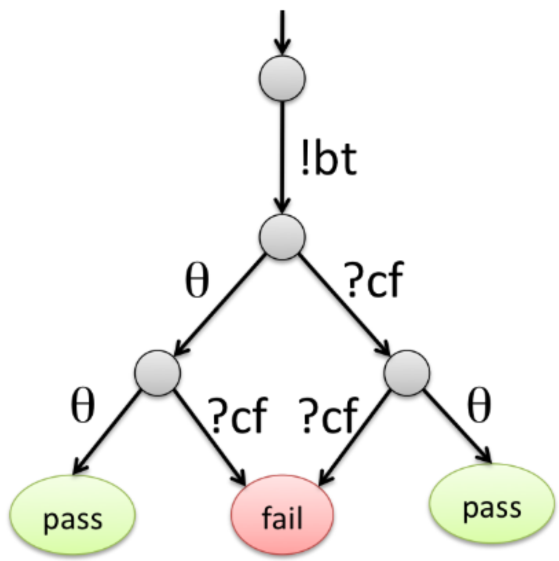
ioco?

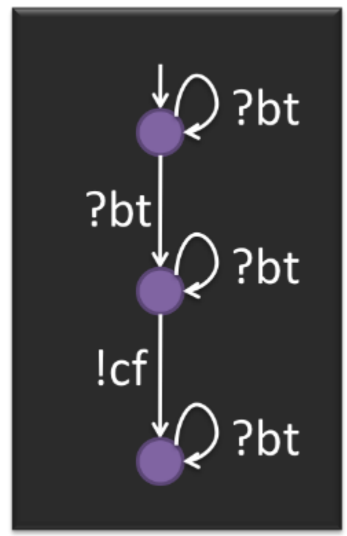


out(imp after ?bt) = {!cf,δ} but out(spec after ?bt)={!cf}

Conformance Testing: Extensionally

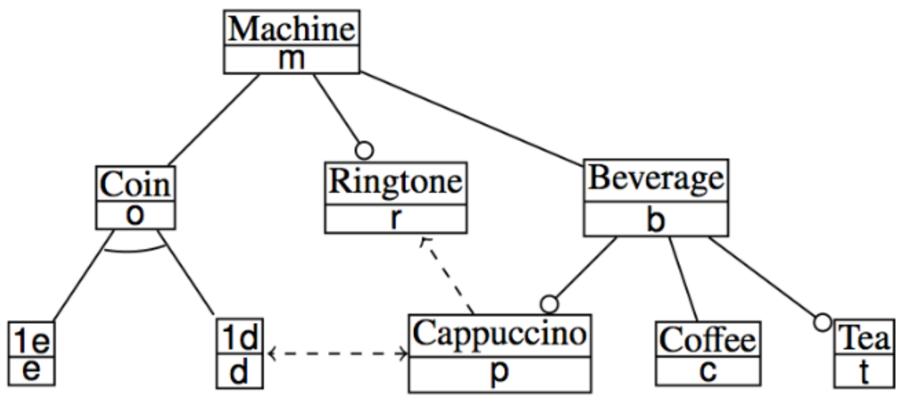






Product Lines

Structural Modeling: Features



[Schobbens, Heymans, and Trigaux, RE'06]

Behavioral Models for SPLs

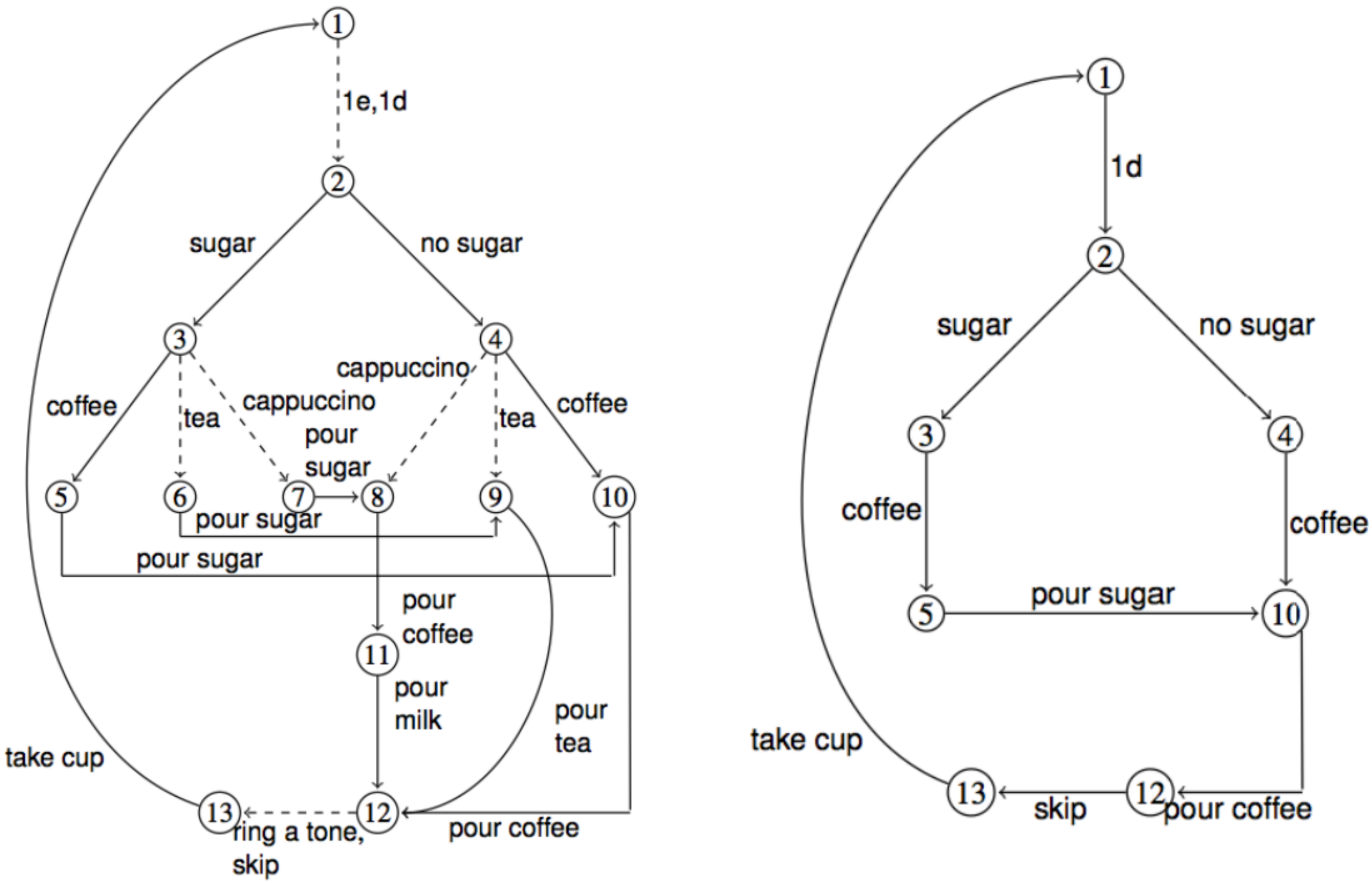
- Various extensions of LTSs:
  Modal- and **Featured Transitions Systems**

- Feature(d) Petri Nets

- PL-CCS

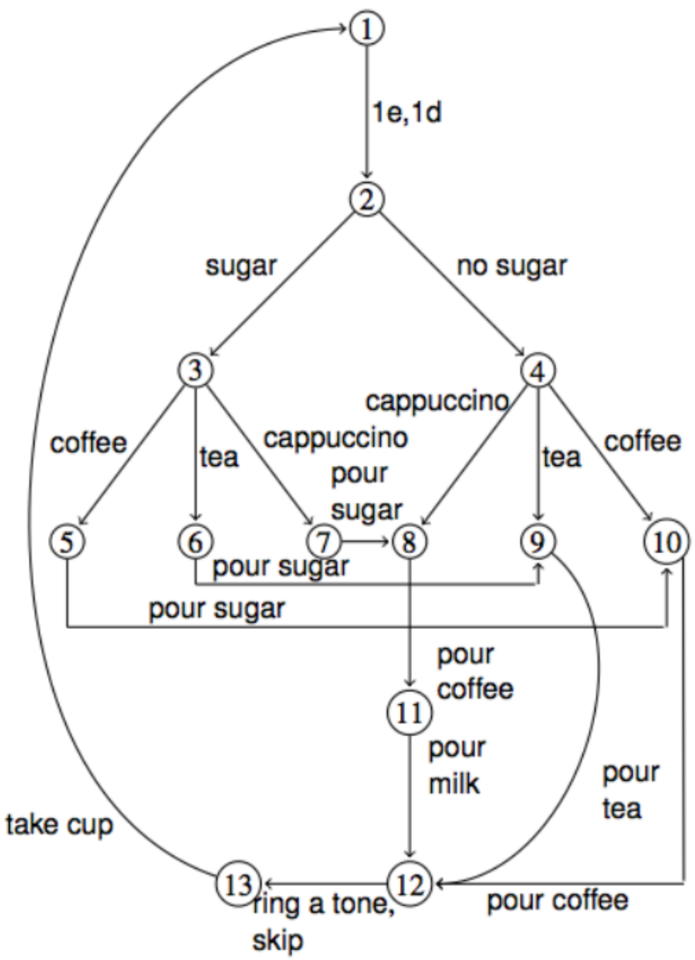
- **Delta-Oriented** FSMs

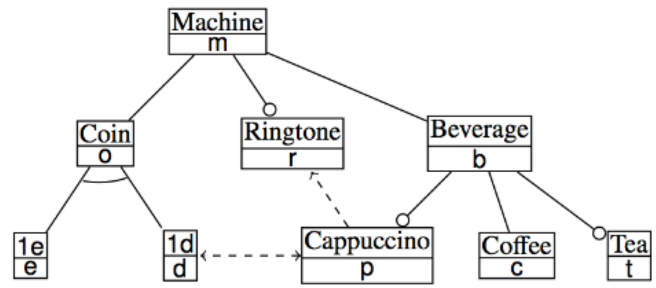
Modal Transition Systems



[Larsen and Thomsen. LICS'98]

Featured Transition Systems





Transitions	*Features*
$s_1 \xrightarrow{1e} s_2$	e
$s_1 \xrightarrow{1d} s_2$	d
$s_2 \xrightarrow{coffee} s_5$	c
$s_2 \xrightarrow{tea} s_6$	t

Transitions	*Features*
$s_2 \xrightarrow{cappuccino} s_7$	p
$s_{12} \xrightarrow{ring\ a\ tone} s_{13}$	$p \Rightarrow r$
remaining transitions	m

[Classen et al., ICSE 2010]

Product Line Labeled Transition Systems

$s_1 = \textit{1e}.s_2 \oplus \textit{1d}.s_2$

$s_2 = \textit{sugar}.s_3 + \textit{no sugar}.s_4$

$s_3 = \textit{coffee}.s_5 + \langle \textit{tea}.s_6 + \textit{cappuccino}.s_7 \rangle$

$s_4 = \textit{coffee}.s_{10} + \langle \textit{tea}.s_9 + \textit{cappuccino}.s_8 \rangle$

$s_5 = \textit{pour sugar}.s_{10}$

$s_6 = \textit{pour sugar}.s_9$

$s_7 = \textit{pour sugar}.s_8$

$s_8 = \textit{pour coffee}.s_{11}$

$s_9 = \textit{pour tea}.s_{12}$

$s_{10} = \textit{pour coffee}.s_{12}$

$s_{12} = \textit{ring tone}.s_{13} + \textit{skip}.s_{13}$

$s_{13} = \textit{take cup}.s_1 \quad .$

$\langle P \rangle = P \oplus \mathbf{Nil}.$

$(\mathbb{P} \times \{L, R, ?\}^I, A, I, \rightarrow)$

$\rightarrow \subseteq (\mathbb{P} \times \{L, R, ?\}^I) \times (A \times \{L, R, ?\}^I) \times (\mathbb{P} \times \{L, R, ?\}^I)$

[Gruler, Leucker, and Scheidemann., FMOODS'08]

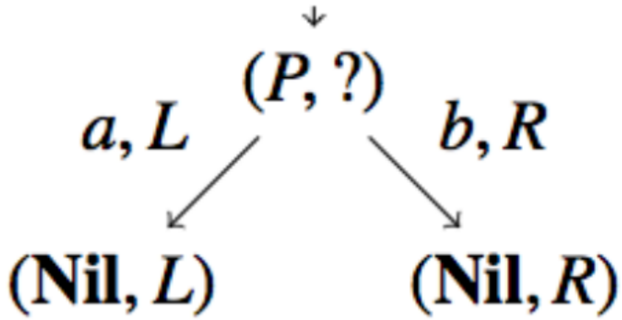
Expressiveness

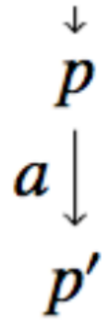
- SPL Formalism: $\mathbf{M} = (\mathbb{M}, [\![\ ]\!])$, $[\![\ ]\!] : \mathbb{M} \rightarrow 2^{\mathbf{P}}$
- Given $\mathbf{M} = (\mathbb{M}, [\![\ ]\!])$, $\mathbf{M}' = (\mathbb{M}', [\![\ ]\!]')$
  an encoding is $E : \mathbb{M} \rightarrow \mathbb{M}'$
  such that $[\![\ ]\!] = E \circ [\![\ ]\!]'$

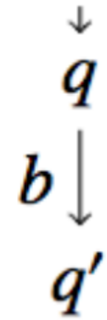
$$\textbf{MTSs} \longrightarrow \textbf{PL-LTSs} \longrightarrow \textbf{FTSs}$$

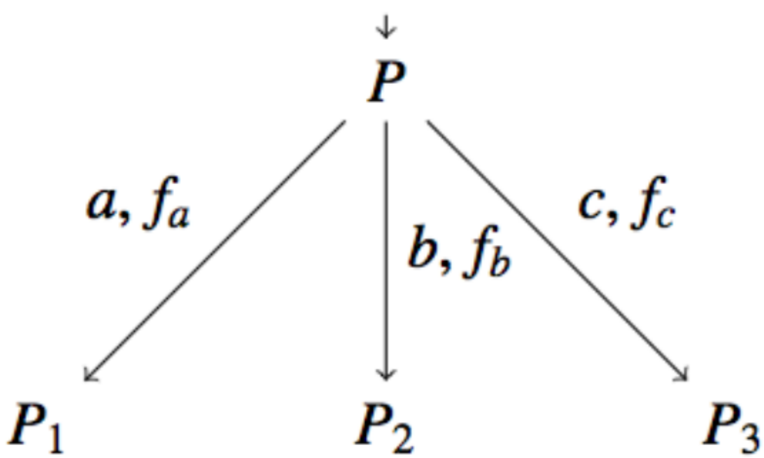
[Beohar, Varshosaz, MRM, SCP'15]

Counterexamples





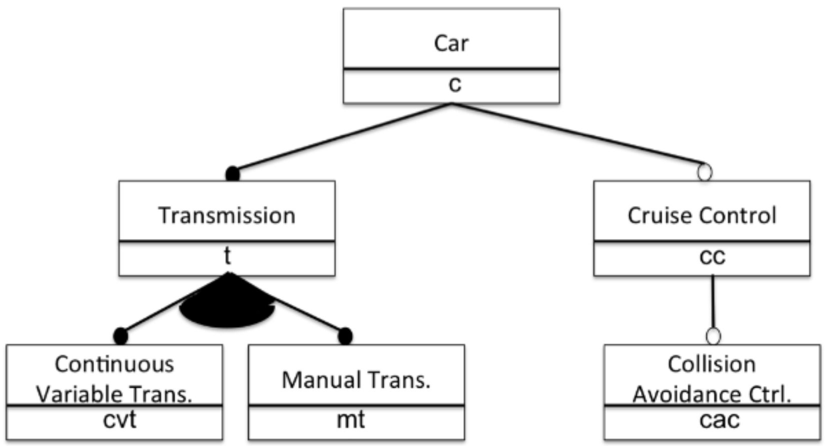


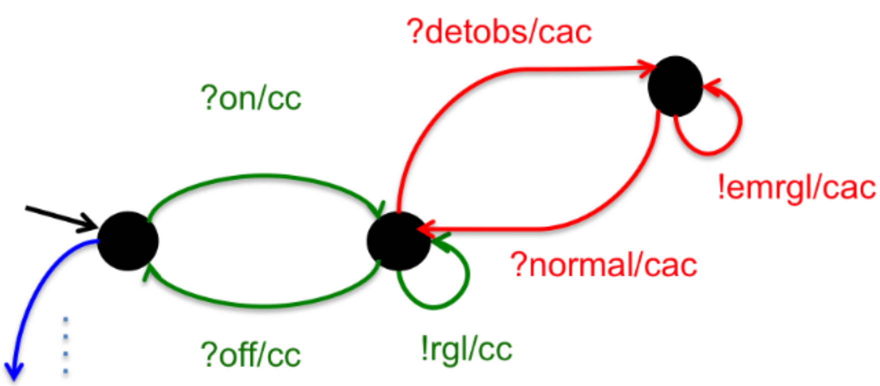


Featured Input-Output Transition Systems

$(S, s, A_\tau, F, T, \Lambda)$, where:

- S, s, A_τ have the same intuition as in IOLTS,
- F is a set of propositions (features),
- $T \subseteq S \times A_\tau \times \mathbb{B}(F) \times S$
- $\Lambda \subseteq \{\lambda : F \to \mathbb{B}\}$

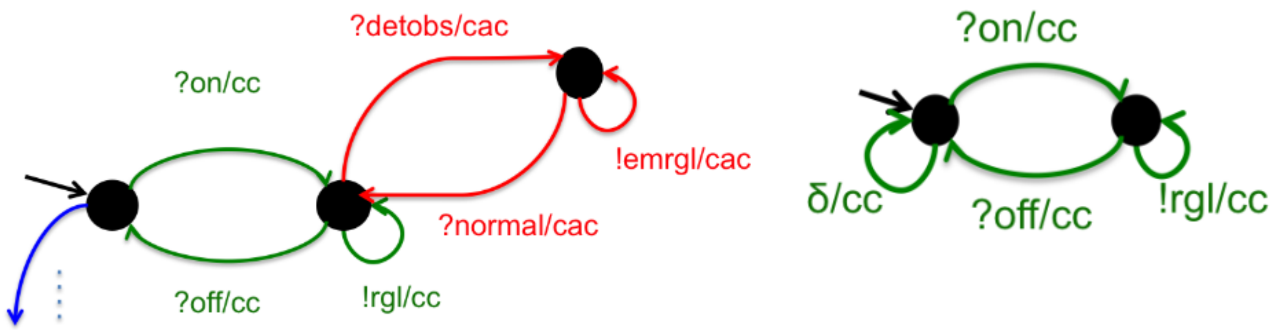




Product Derivation

$$\frac{\exists_{\lambda}\ \lambda \models (\gamma(s,a,s') \wedge \varphi)}{\Delta_{\varphi}(s) \xrightarrow{a}_{\gamma(s,a,s') \wedge \varphi} \Delta_{\varphi}(s')} \quad (1)$$

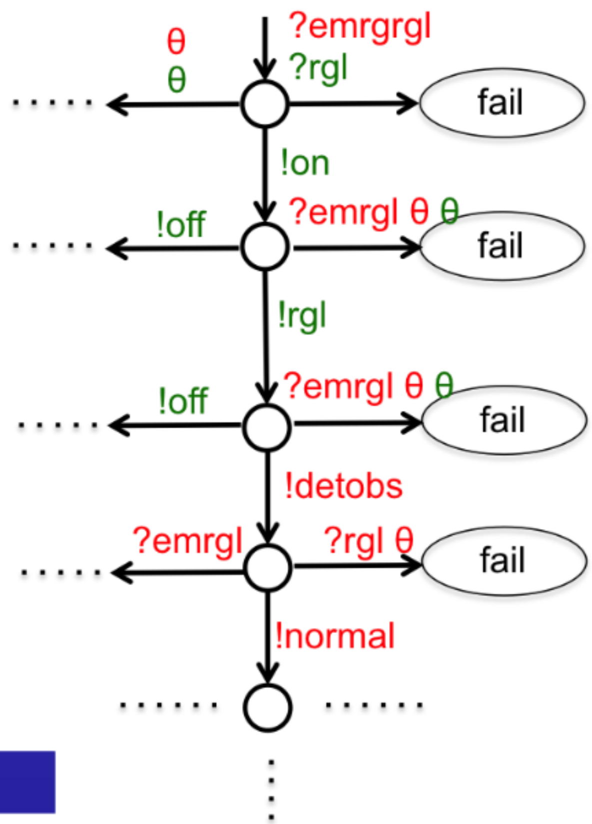
$$\frac{\nexists_{\lambda,s',a}\ \lambda \models (\gamma(s,a,s') \wedge \varphi) \wedge a \in A_{\mathcal{O}} \cup \{\tau\}}{\Delta_{\varphi}(s) \xrightarrow{\delta}_{\varphi} \Delta_{\varphi}(s)} \quad (2)$$

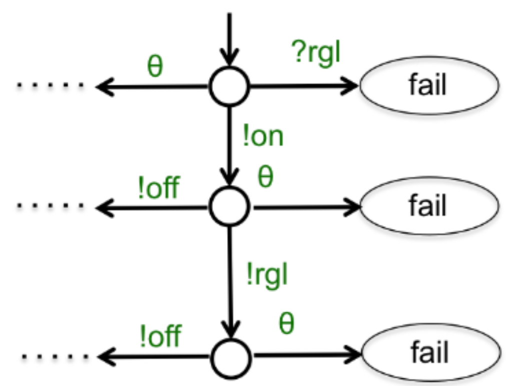


Test-Suite Derivation

$$\frac{X, Y \neq \emptyset \quad (X,\sigma), (Y, \sigma a) \in \mathbf{X}_s^{\varphi}}{(X,\sigma) \xrightarrow{f(a)}_{\varphi} (Y, \sigma a)} \quad (3) \qquad \frac{a \in A_{\mathcal{O}} \cup \{\theta\} \quad (X,\sigma) \xrightarrow{a}_{\varphi} (Y, \sigma')}{(X,\sigma) \xrightarrow{a}_{\varphi} \mathbf{pass}} \quad (4)$$

$$\frac{(X,\sigma) \not\xrightarrow{a}_{\varphi} \mathbf{pass}}{(X,\sigma) \xrightarrow{a}_{\varphi} \mathbf{fail}} \quad (5) \qquad \frac{a \in A_{\mathcal{O}} \cup \{\theta\}}{\mathbf{pass} \xrightarrow{a}_{\varphi} \mathbf{pass} \quad \mathbf{fail} \xrightarrow{a}_{\varphi} \mathbf{fail}} \quad (6)$$



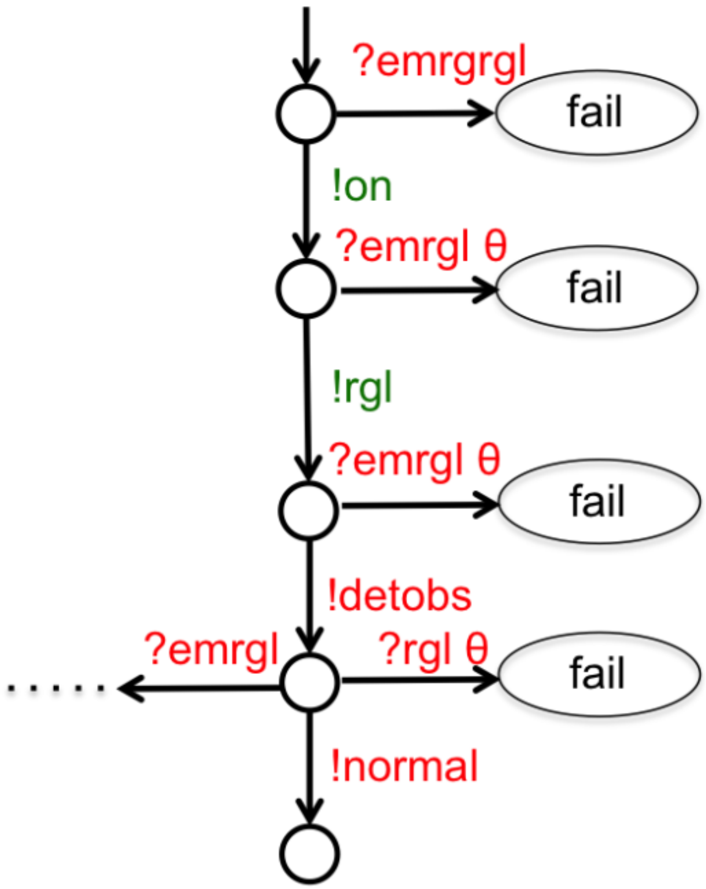


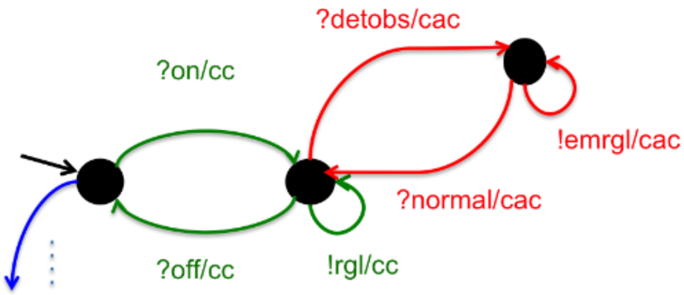
Click!

$$
\begin{array}{ccc}
\Delta_{\varphi}(s) & \xrightarrow{\textbf{test generation}} & (\hat{\mathbf{X}}_s, \varepsilon) \\
\Delta_{\varphi'}(\_) \downarrow & & \downarrow \Delta^{\mathbf{t}}_{\varphi'}(\_) \\
\Delta_{\varphi \wedge \varphi'}(s) & \xrightarrow[\textbf{test generation}]{} & (\hat{\mathbf{X}}_s^{\varphi \wedge \varphi'}, \varepsilon) \cong \Delta^{\mathbf{t}}_{\varphi'}(\hat{\mathbf{X}}_s, \varepsilon)
\end{array}
$$

[Beohar and MRM. SVT'14]

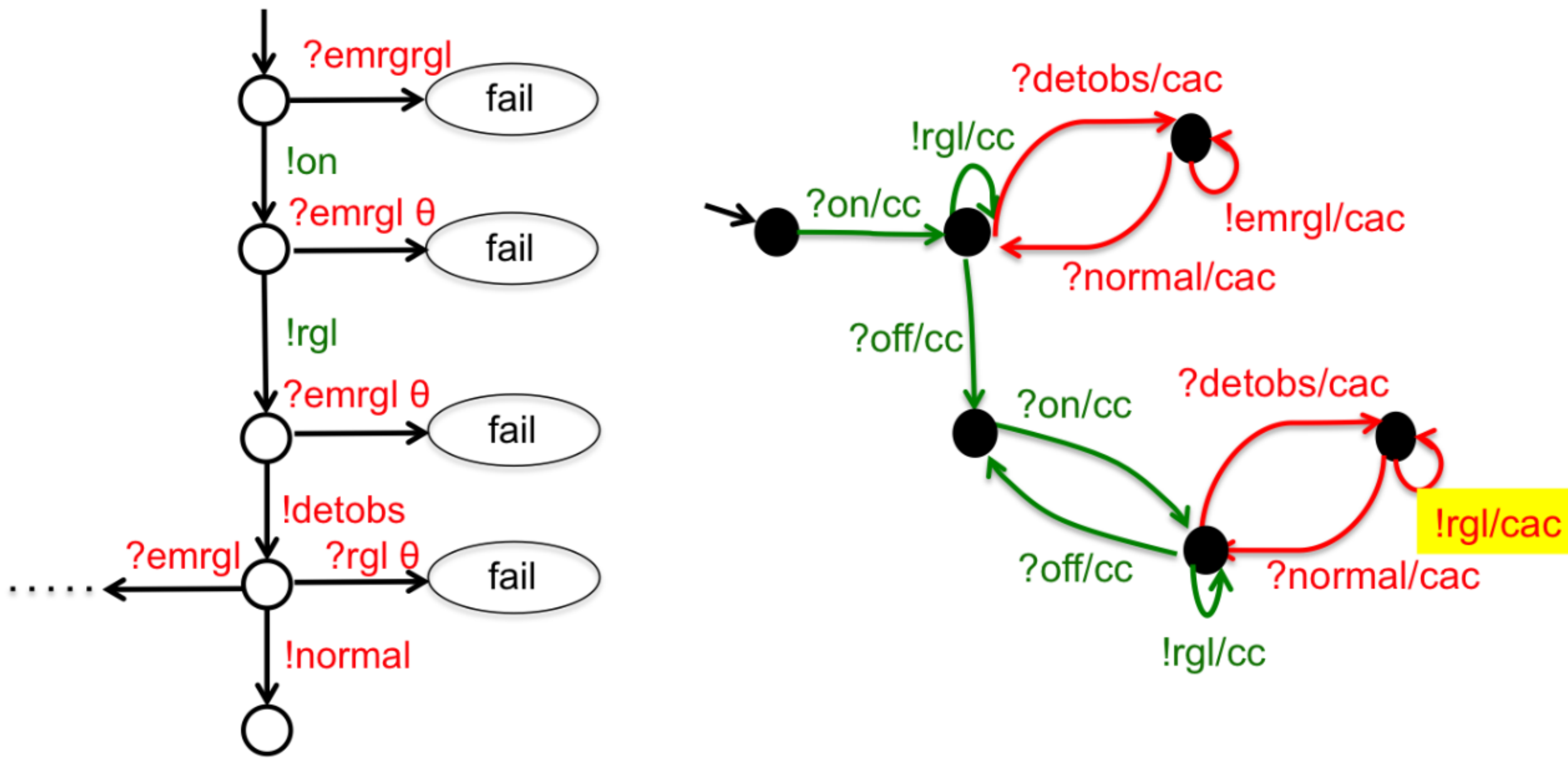
Spinal Test Suites





Spines: Simple paths through test suite leading to new behavior

Non-Exhaustiveness



Recovering Exhaustiveness

Orthogonal implementations:

interacting with old features
does not influence new features' behavior

Theorem: Spinal test suites are exhaustive for orthogonal implementations

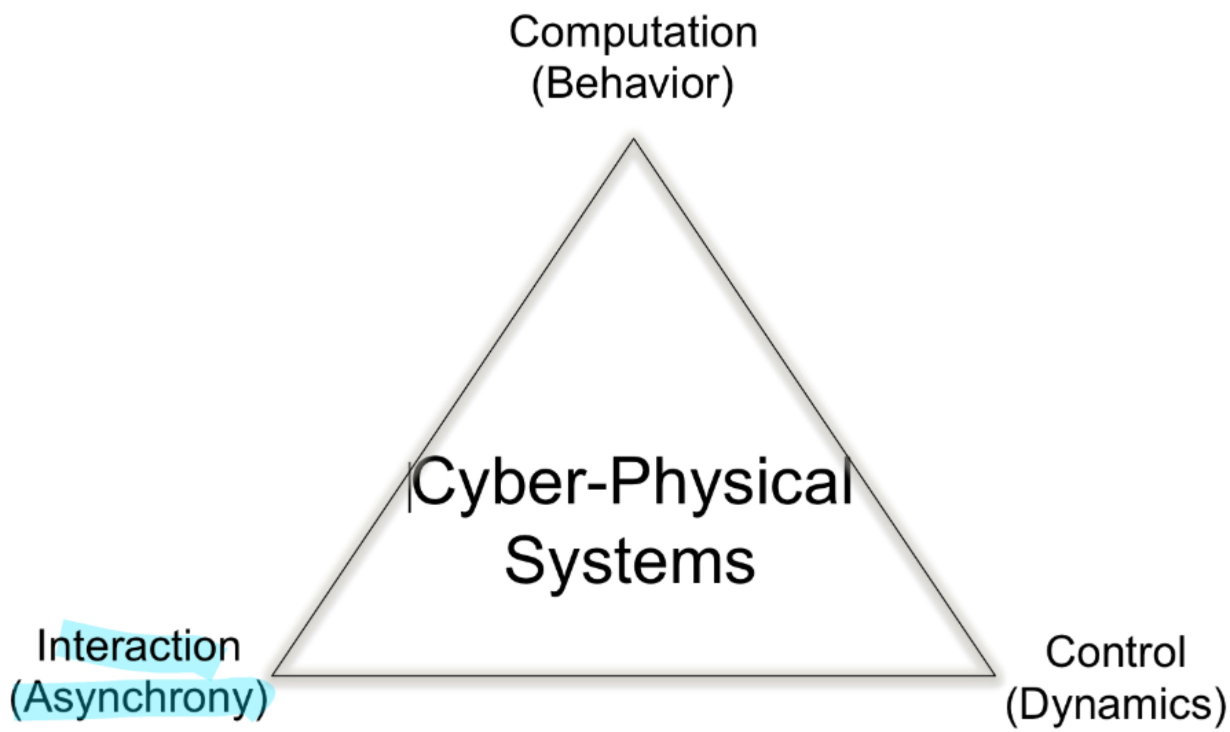
[Beohar and MRM. MBT'14]

Work in Progress

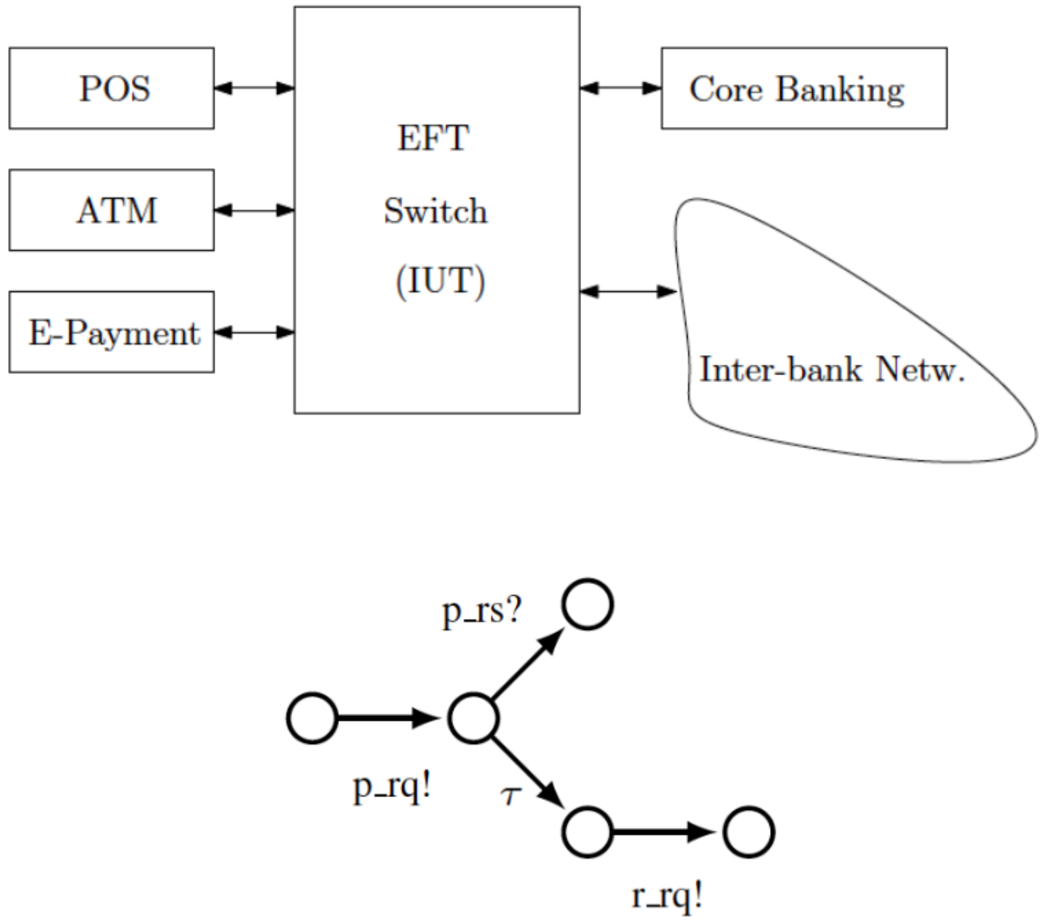
Using differential program verification to establish orthogonality

[Logozzo, Lahiri, Fahndrich, and Blackshear. PLDI'14]

Cyber-Physical Systems: System Dynamics

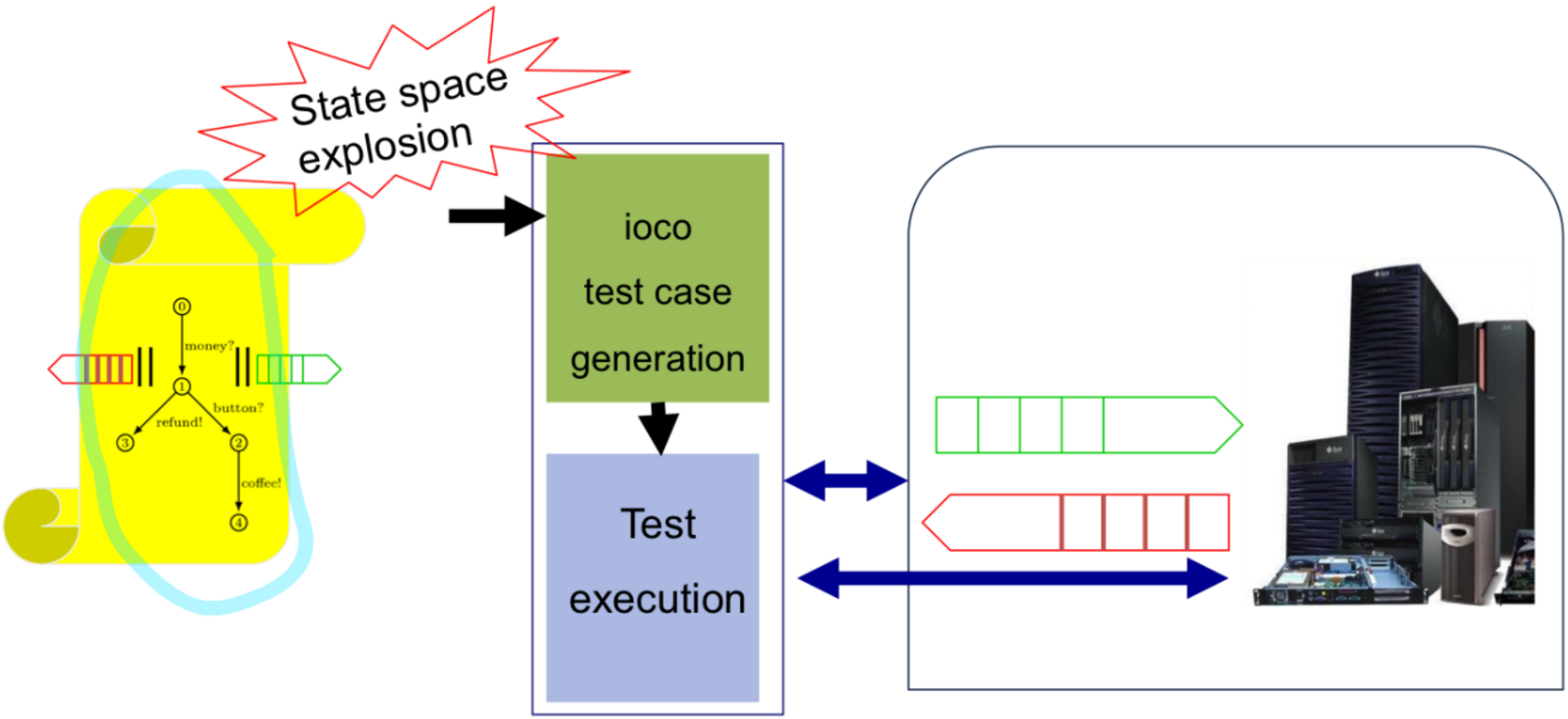


Our Encounter with Asynchrony



[Asadi, Khosravi, MRM, Noroozi. FSEN'11]

Test-Case Generation

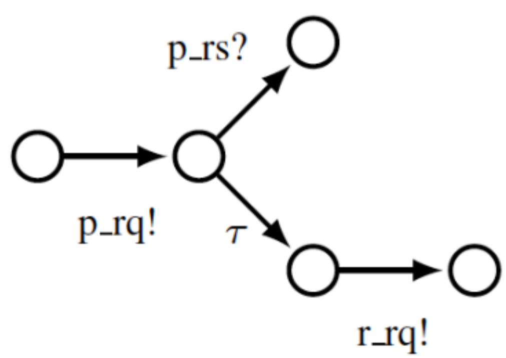


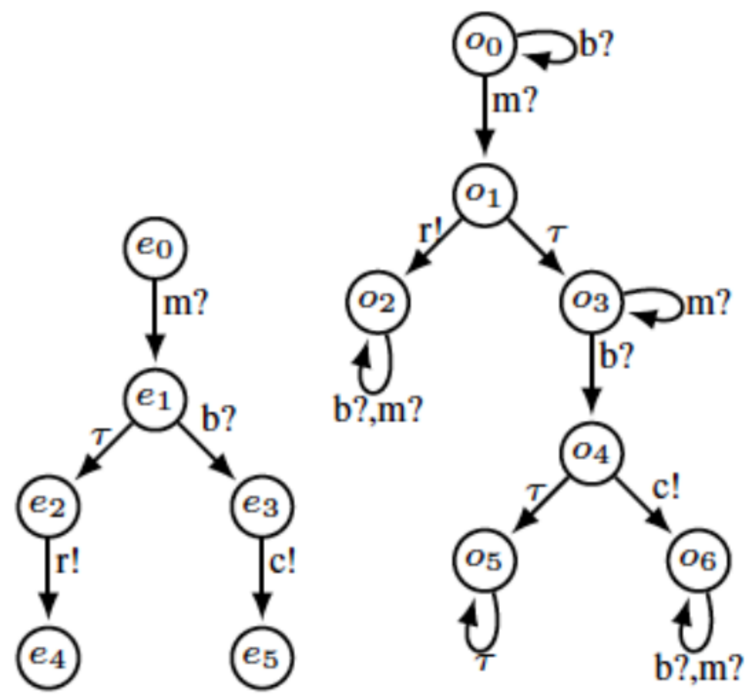
[Tretmans and Verhaard. PSTV'92]

Internal Choice IO(L)TS

IOLTS$^{\sqcap}$: Inputs are only enabled when no output action is weakly enabled

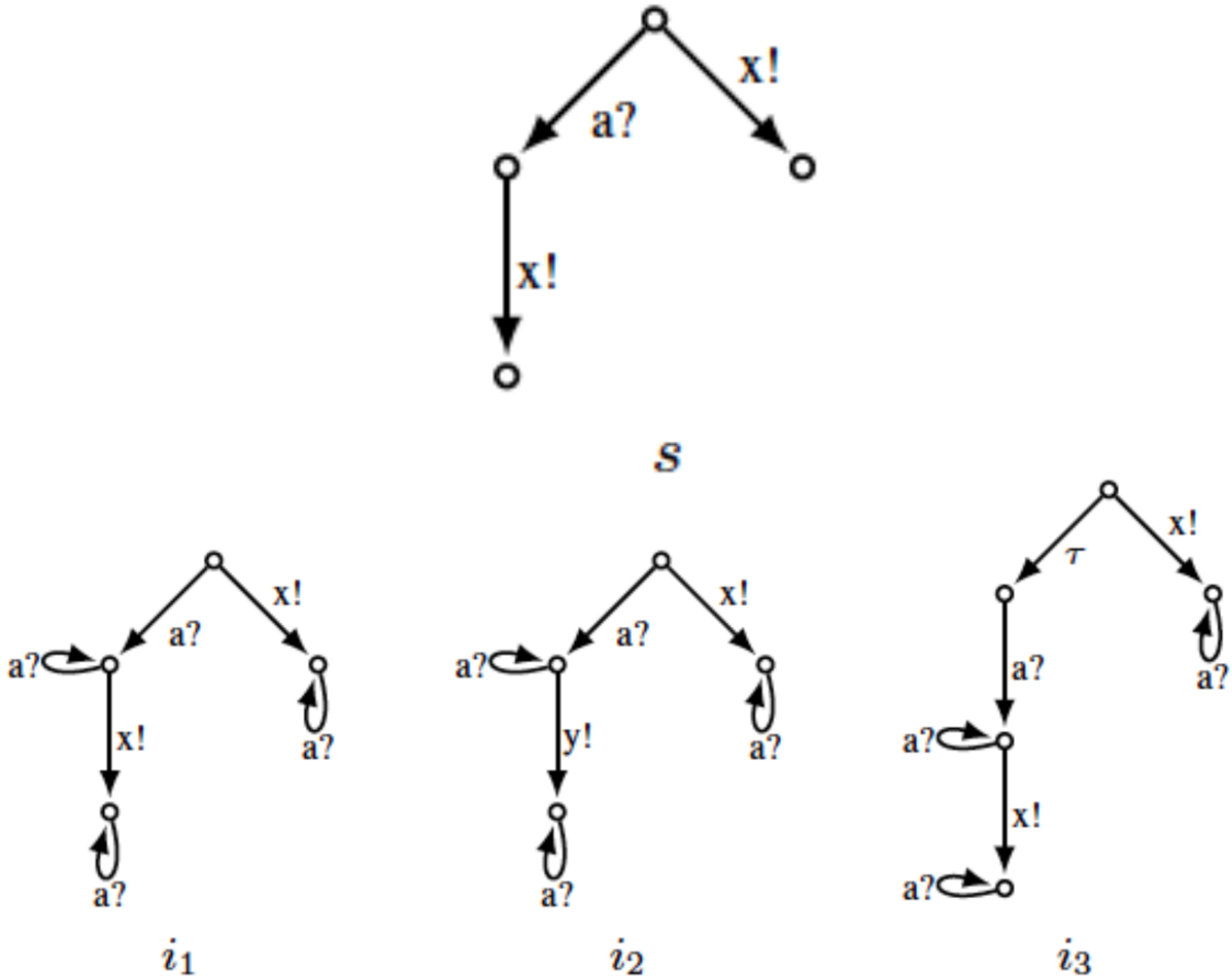
IOTS$^{\sqcap}$: Input-enabled **IOLTS**$^{\sqcap}$





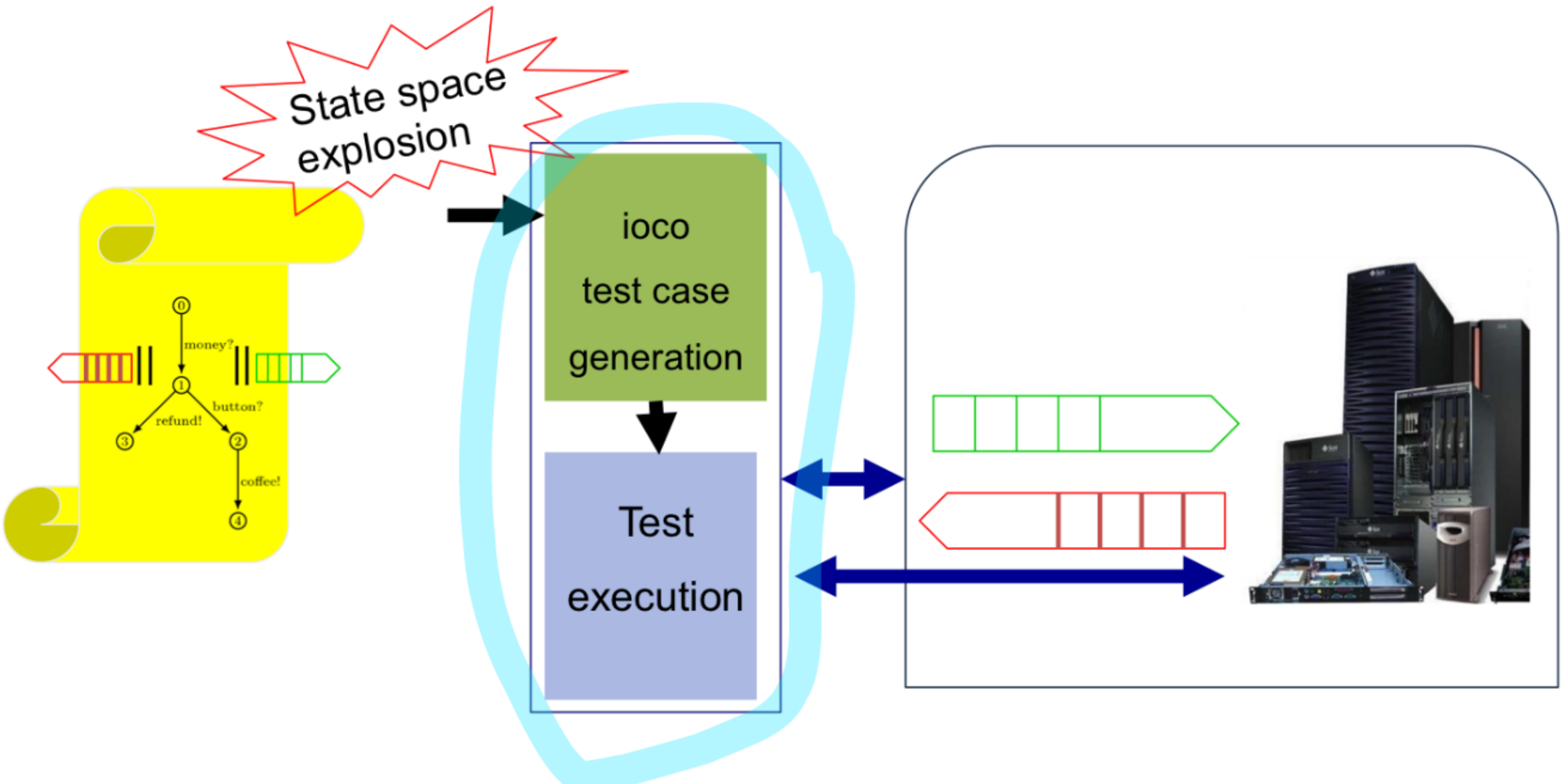
[Weiglhofer and Wotawa. COMPSAC'09]

Testing Power



[Noroozi, Khosravi, MRM, Willemse. SoSym'15]

Test-Case Generation



Test Case Generation

Challenge: selecting test cases that lead to sound verdicts in queue context

Internal choice test cases: observe quiescence before feeding input

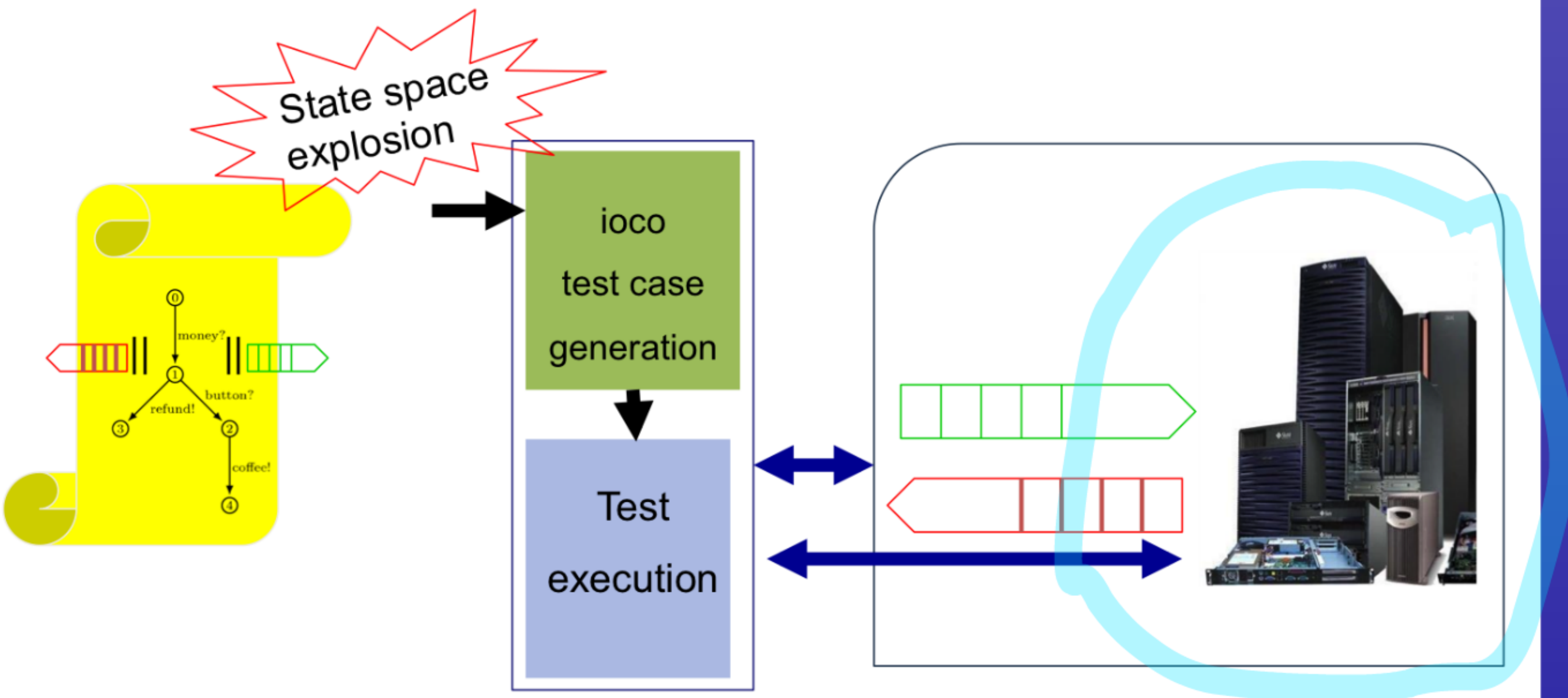
Theorem:

for a non-divergent spec s, internal choice imp. i,
for each internal choice test case t of s,

i passes t if and only if Q(i) passes t

[Noroozi, Khosravi, MRM, Willemse. SoSym'15]

Test-Case Generation



Testable Implementations

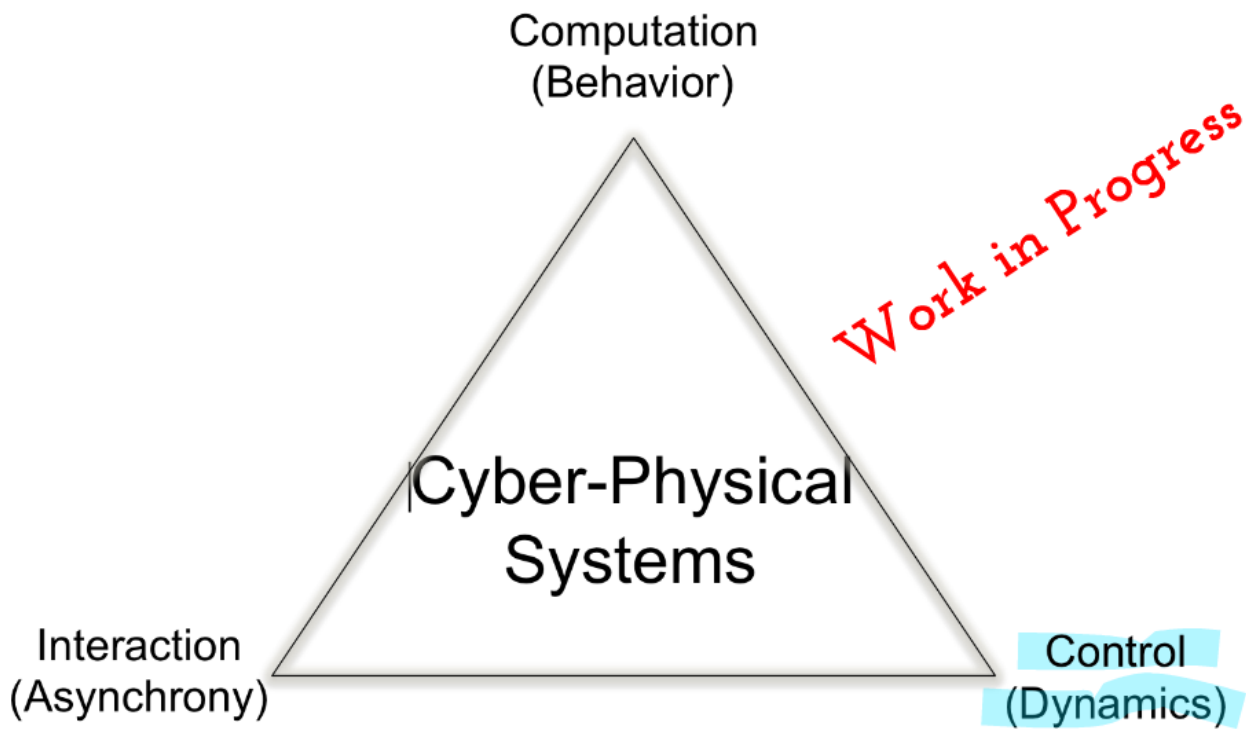
Delay right-closed implementations:
if t.x!.a? in straces(i) then t.a?.x! in straces(i)

Theorem:
If implementation i is delay right-closed,
then i ioco s if and only if Q(i) ioco s

If for all test cases t, i passes t iff Q(i) passes t
then i is delay right-closed.

[Noroozi, Khosravi, MRM, Willemse. SoSym'15]

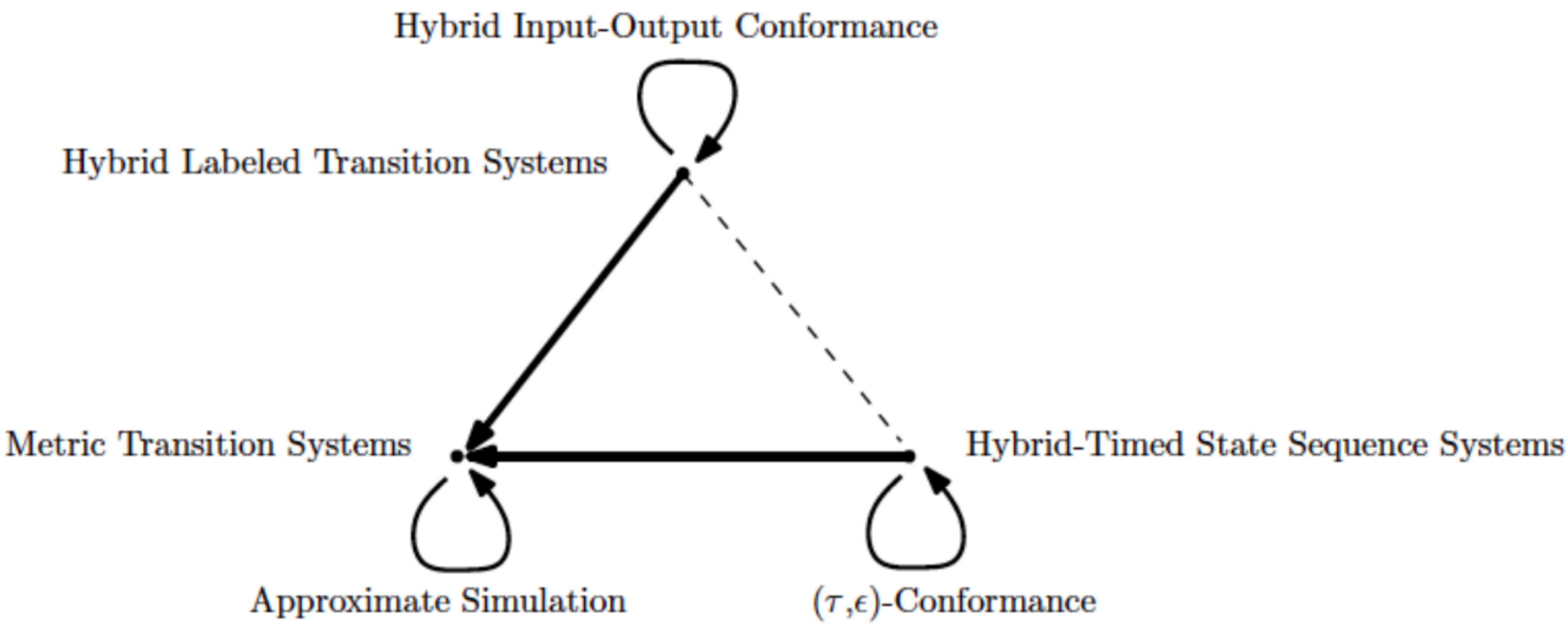
Cyber-Physical Systems: System Dynamics



Semantics Models

- Hybrid Labeled Transition System:
  - LTS with continuous trajectory labels
- Metric Transition Systems:
  - LTS with observation functions on states and metrics on states and transition
- Timed State Sequences:
  - Mappings of initial conditions and discretized inputs to discretized outputs

Models and Conformance



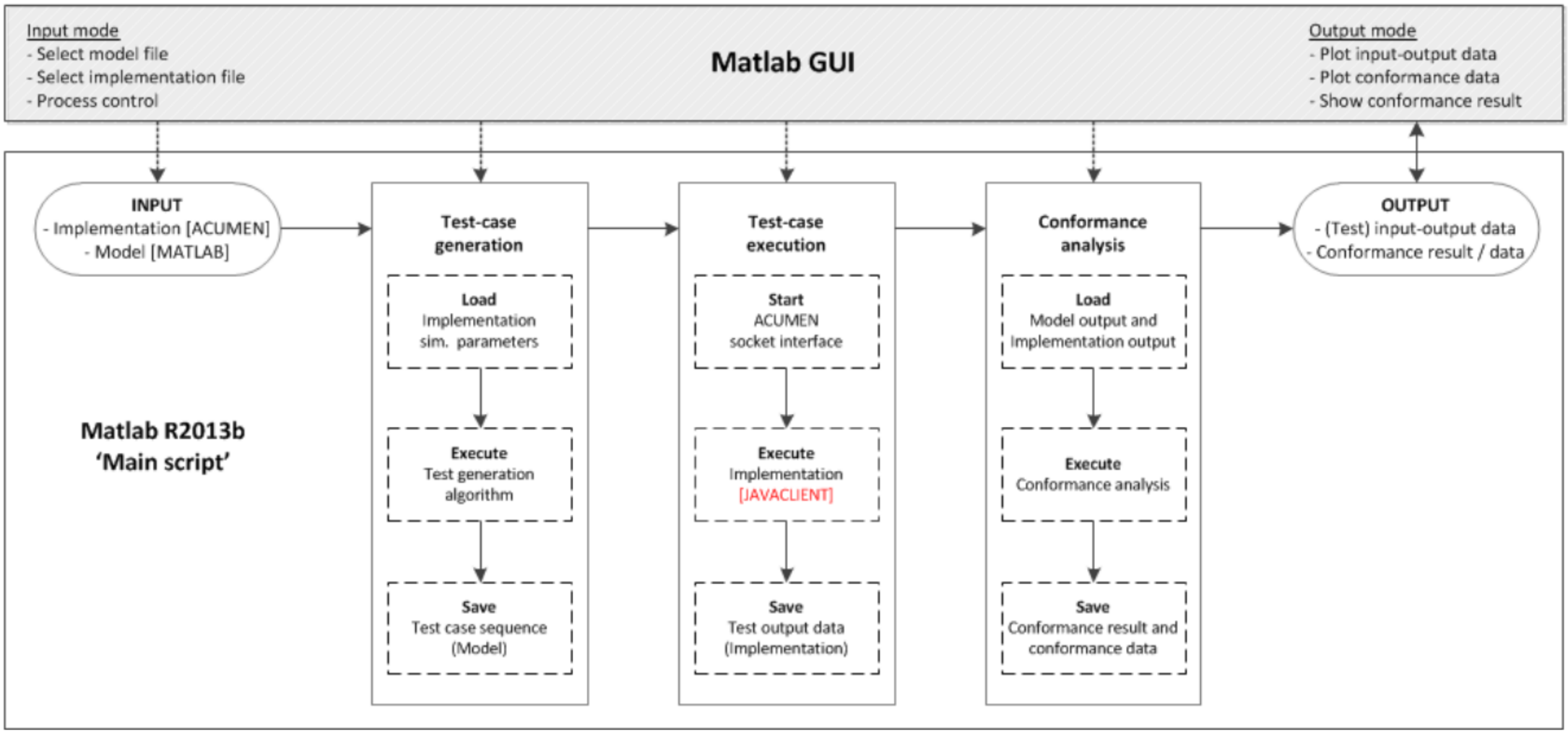
[Khakpour and MRM. CONCUR'15]
[Mohaqeqi, MRM, and Taha. AVoCS'14]

Our Wish List

A pre-congruence for a hybrid model with:

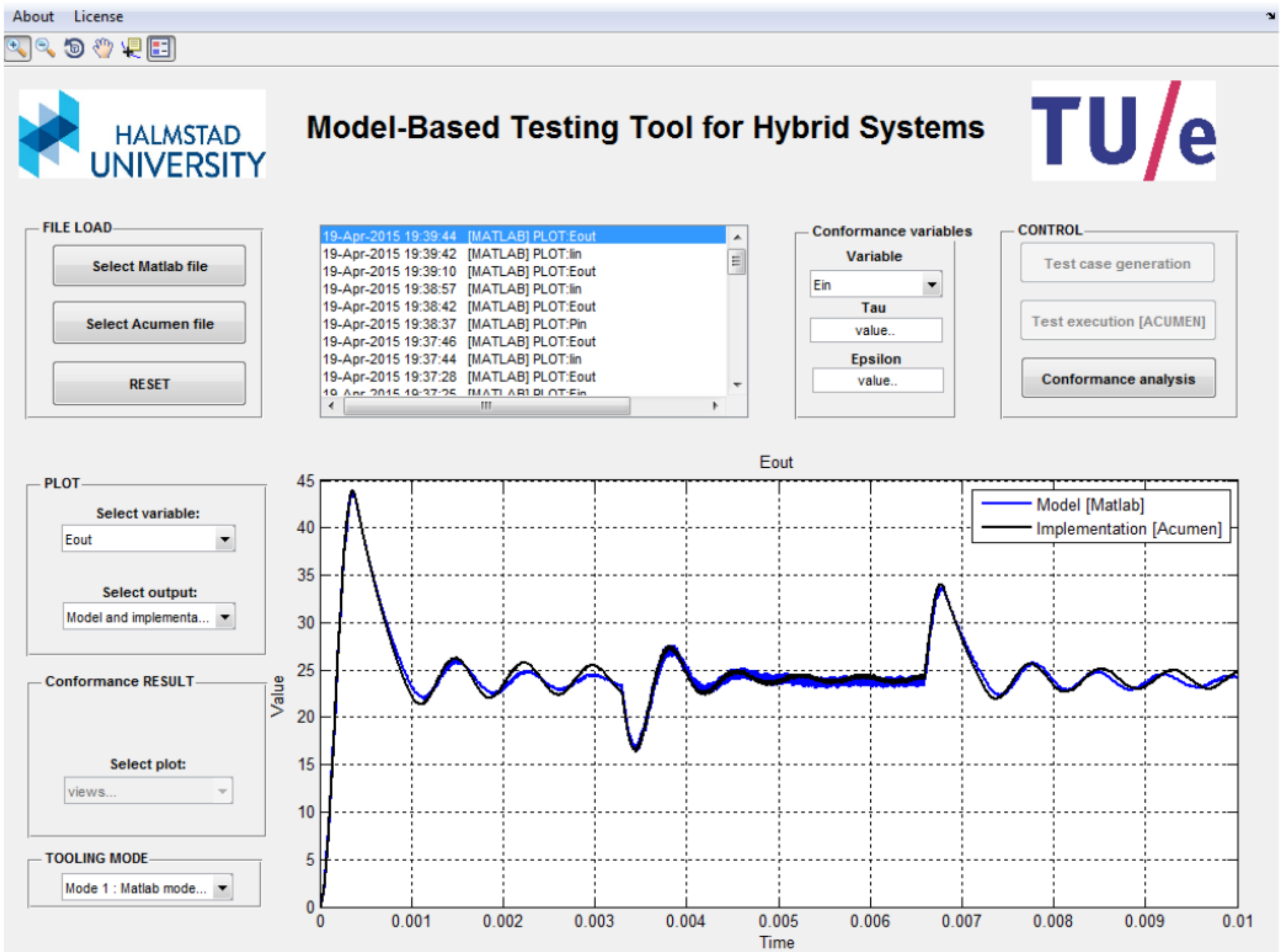
- Partial and non-deterministic specifications,
- Explicit input and output discrete actions and continuous signals,
- Sampling of continuous input and output, and
- A notion of proximity for comparing sampled outputs.

Tool



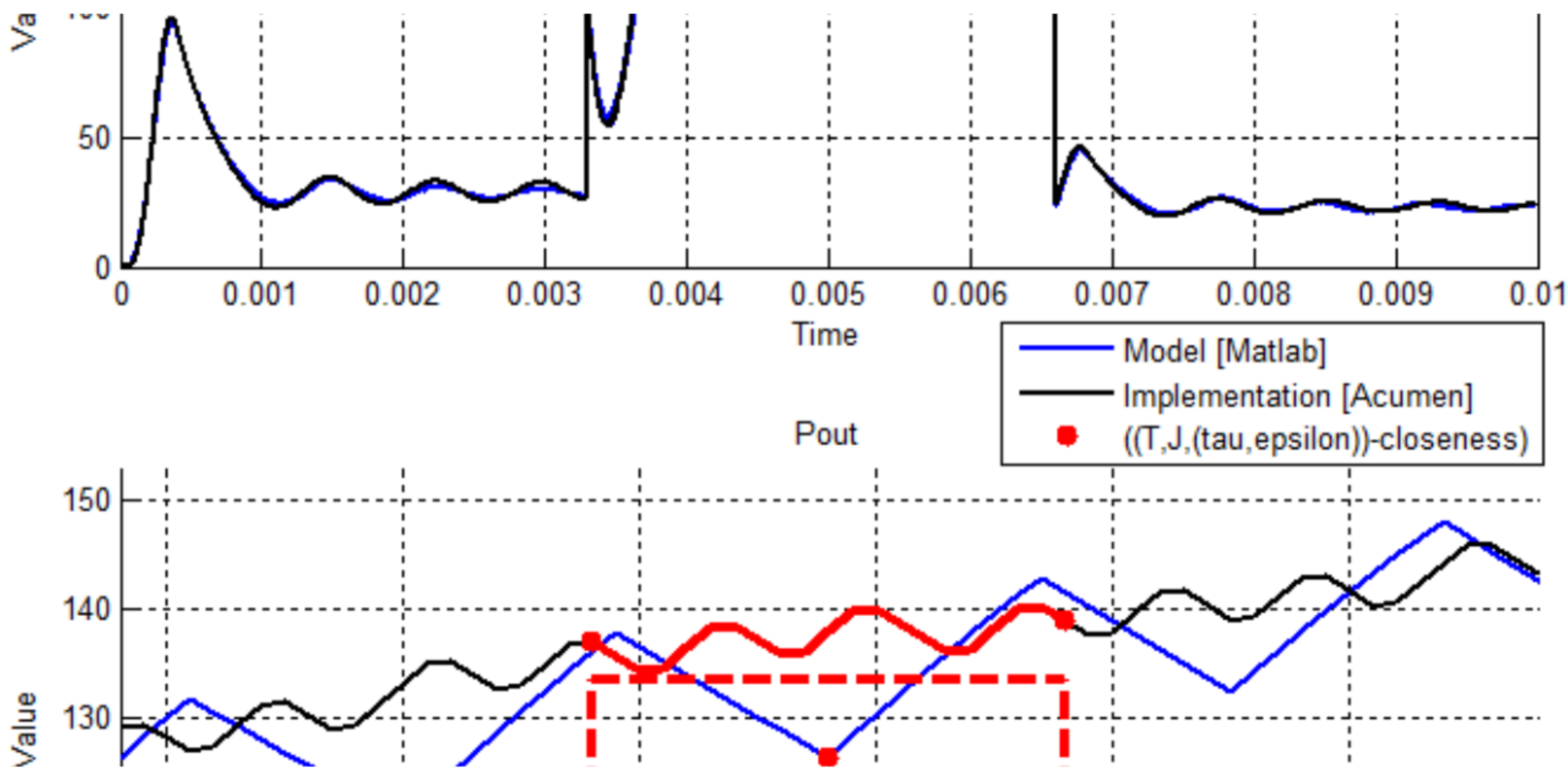
Input mode
- Select model file
- Select implementation file
- Process control
Matlab GUI
Output mode
- Plot input-output data
- Plot conformance data
- Show conformance result
INPUT
- Implementation [ACUMEN]
- Model [MATLAB]
Test-case generation
Load
Implementation sim. parameters
Execute
Test generation algorithm
Save
Test case sequence (Model)
Test-case execution
Start
ACUMEN socket interface
Execute
Implementation
[JAVACLIENT]
Save
Test output data (Implementation)
Conformance analysis
Load
Model output and Implementation output
Execute
Conformance analysis
Save
Conformance result and conformance data
OUTPUT
- (Test) input-output data
- Conformance result / data
Matlab R2013b
'Main script'

Tool



[Aerts, MRM, and Reniers. ICTAC'15]

Tool



Va
50
0
0
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.009
0.01
Time
Model [Matlab]
Implementation [Acumen]
((T,J,(tau,epsilon))-closeness)
Pout
150
140
130
Value

Ongoing Research, Open Issues

Theoretical Challenges

- Compositionality and decompositional testing
  see, e.g., [Benes et al. CBSE'15]
  [Noroozi et al. MBT'13]
- Logical characterizations
- Zeno behavior (quiescence, agility)
- Robustness
- Coverage

Practical Challenges

- Configuration selection
- Feature interaction
- Sampling
- Real-time synchronization between tester and implementation
- Rigorous models and simulations

Other Ongoing Projects

- MBT4SPL: Delta-Oriented Testing
  see [ICFEM '15]
- AUTO-CAAS: Diagnosis of AUTOSAR-based software
  see [WASA'15]
- EFFEMBAC: Combining Model-Based- and Concolic Testing

Collaborators

Narges Khakpour

Wojciech Mostowski

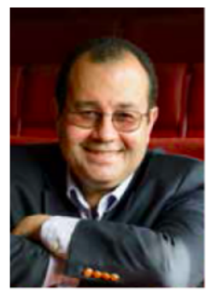
Harsh Beohar

Tim Willemse

Neda Noroozi

Morteza Mohaqeqi

Mahsa Varshosaz



Walid Taha

Ramtin Khosravi

Michel Reniers

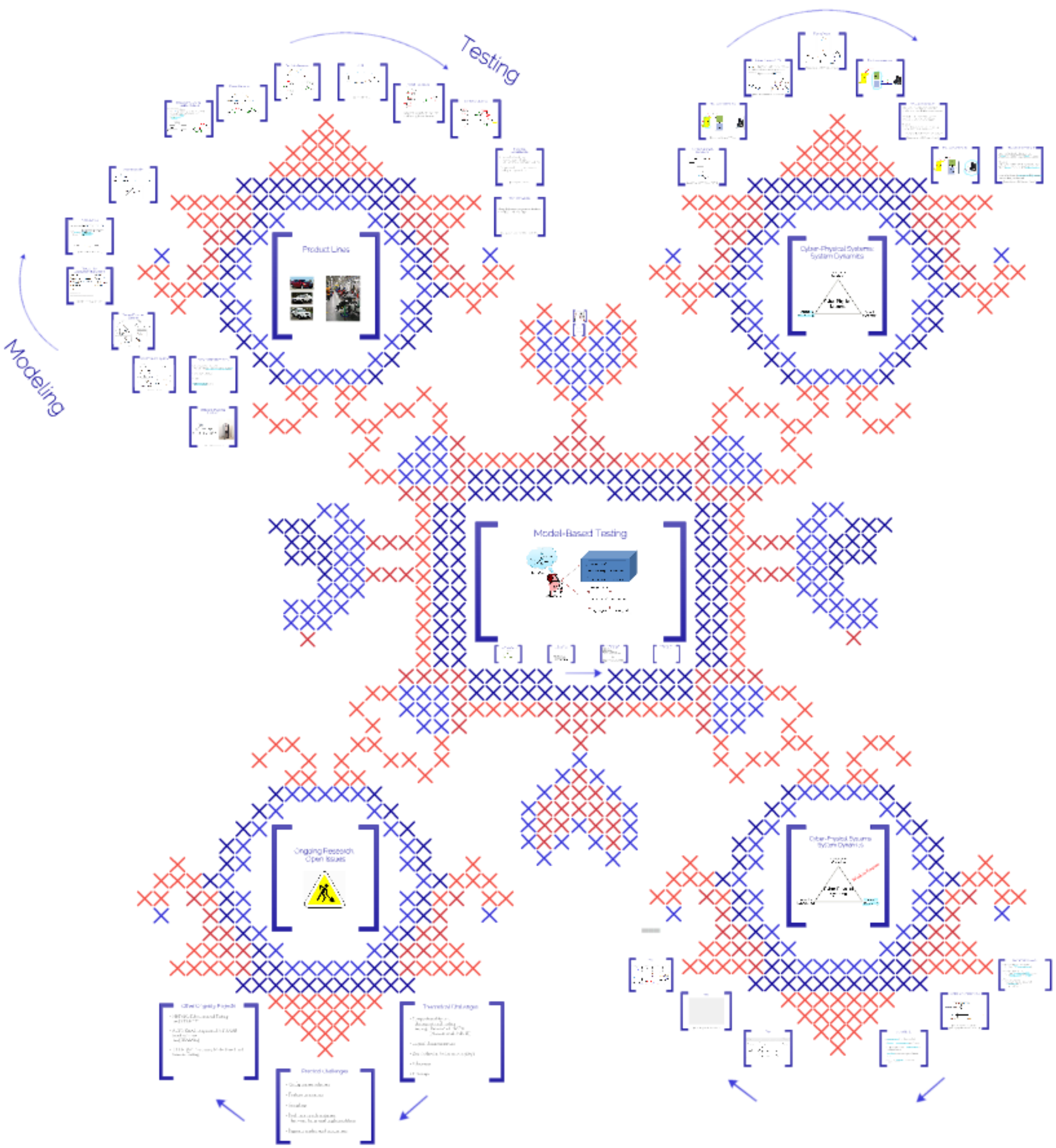
Sebastian Kunze

Masoumeh Taromirad

Arend Aerts

Thank You!

m.r.mousavi@hh.se



From Concurrency Theory to Model-Based Testing Cyber-Physical Systems

Mohammad Mousavi

HALMSTAD UNIVERSITY

KKstiftelsen

Vetenskapsrådet