

Some Issues and Challenges when Testing from Symbolic and Timed Specifications

Julien Schmaltz

Institute for Computing and Information Sciences
Radboud University Nijmegen
The Netherlands
julien@cs.ru.nl

Context

- Model-Based Testing
- Black-box
- Based on Labeled Transition Systems (LTS)
- Formal conformance relation (**ioco** and variations)
- Automatic
- European Project Training and Research On Testing (TAROT)
 - Postdoc in Nijmegen since March 2007
 - Joint work with Jan Tretmans
 - Previous experience with theorem proving and hardware

Motivation

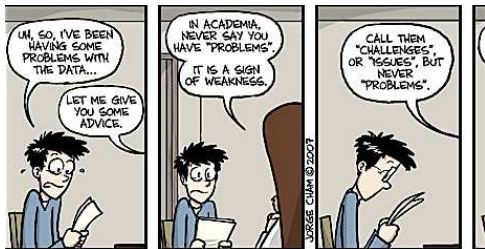
Realistic Safety-Critical Systems deal with both:

- Symbolic Models
 - Several data types
 - Operations on constants and variables
- Timed Models
 - Timed events and actions
 - Timing requirements

There are formal testing techniques for each one of them, but none for both.

Objective

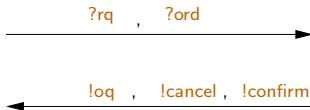
- Global objective: combine testing techniques for timed and symbolic systems
- Today: discuss “problems” ... Issues and Challenges !



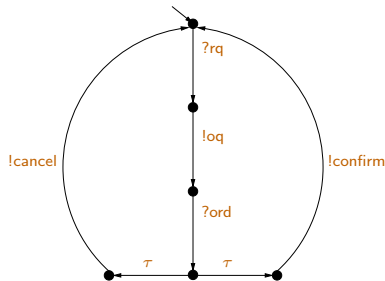
Outline

- 1 The **ioco** theory and its variations
 - The original **ioco** theory
 - Symbolic **ioco**
 - Timed **ioco**
- 2 Timed and Symbolic Models
 - Flow between symbolic and timed models
 - Symbolic Timed Automata: Syntax and Semantics
 - Quiescence

Customer and Supplier



Input Output Labeled Transition System (IOLTS) model of the supplier.



Input Output Conformance: the **ioco** theory

- Specification models are IOLTS
- Implementation models are **input-enabled** IOLTS
- Implementation Imp is **ioco**-conforming to Specification $Spec$ if
 - Every output produced by Imp can also be produced by $Spec$
 - If Imp does not produce any output, so does $Spec$ (quiescence)

Definition (The **ioco** conformance relation)

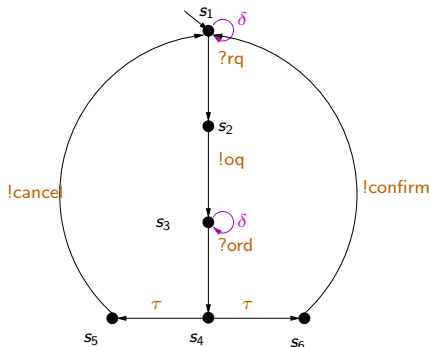
Let $Spec$ be an IOLTS, Imp be an input enabled IOLTS, and let consider a set of traces \mathcal{F} :

$$Imp \mathbf{ioco}_{\mathcal{F}} Spec \equiv \forall \sigma \in \mathcal{F}, \mathbf{out}(i_0 \mathbf{after} \sigma) \subseteq \mathbf{out}(s_0 \mathbf{after} \sigma)$$

Quiescence

Definition (Quiescence)

A state is quiescent if there is **no output or no τ -transition**.



Symbolic Extensions

Different definitions have been proposed:

- Legall (Transition Systems)
- Frantzen (Transition Systems)
- Gottlieb (Constraint Solving)
- Hierons (Extended FSM)
- ...

They all face similar issues:

- Calculations about guards
- Satisfiability of guards
- Symbolic execution/reachability
- Choosing pertinent data in test cases
- Quiescence
- ...

Symbolic Extensions: **sio** by Frantzen

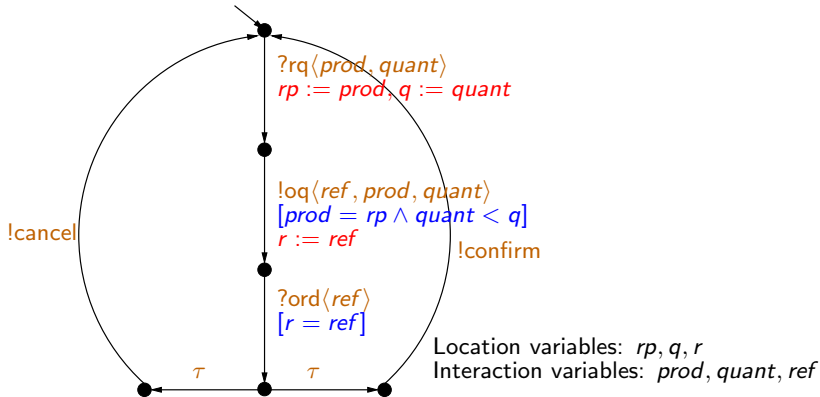
Different definitions have been proposed:

- Legall (Transition Systems)
- **Frantzen (Transition Systems)**
- Gottlieb (Constraint Solving)
- Hierons (Extended FSM)
- ...

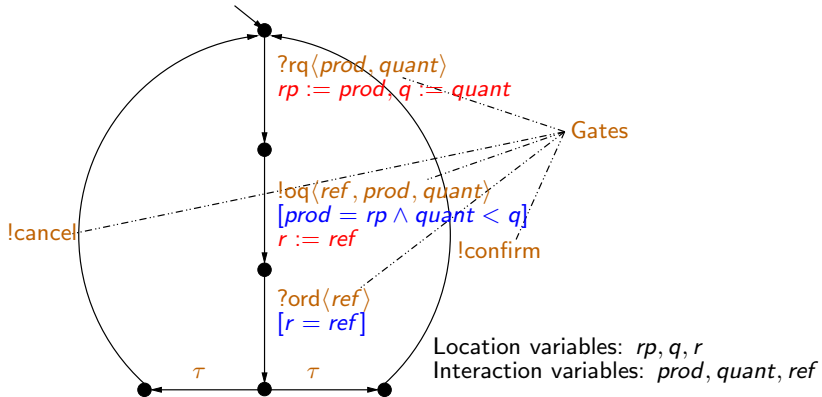
They all face similar issues:

- Calculations about guards
- Satisfiability of guards
- Symbolic execution/reachability
- Choosing pertinent data in test cases
- Quiescence
- ...

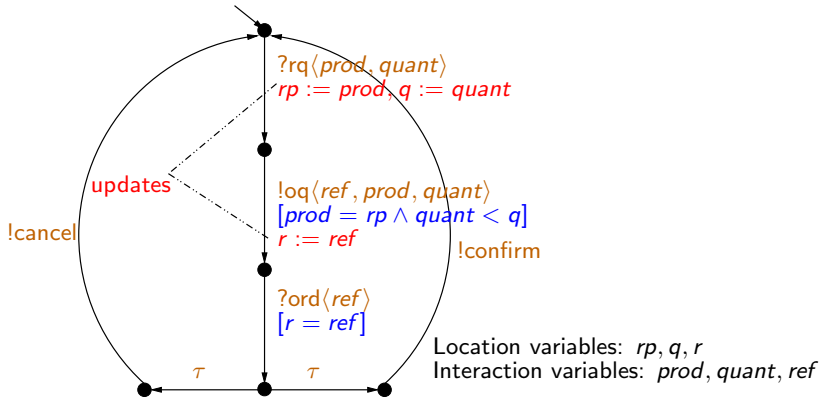
Example: Symbolic specification of our supplier



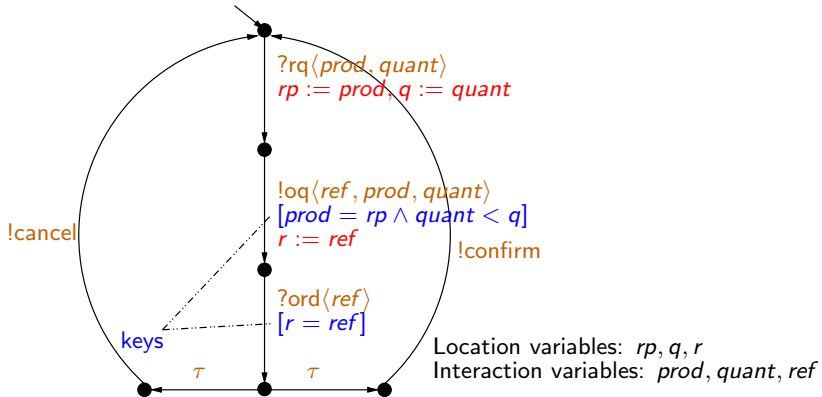
Example: Symbolic specification of our supplier



Example: Symbolic specification of our supplier

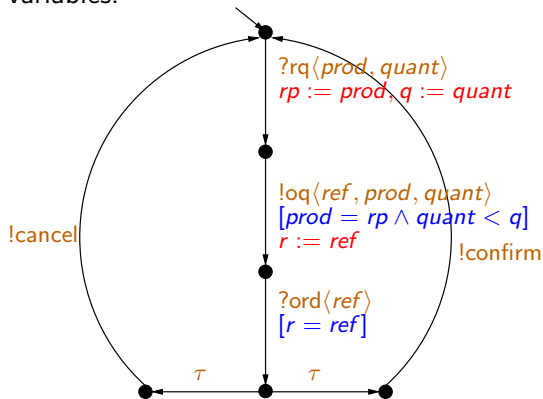


Example: Symbolic specification of our supplier



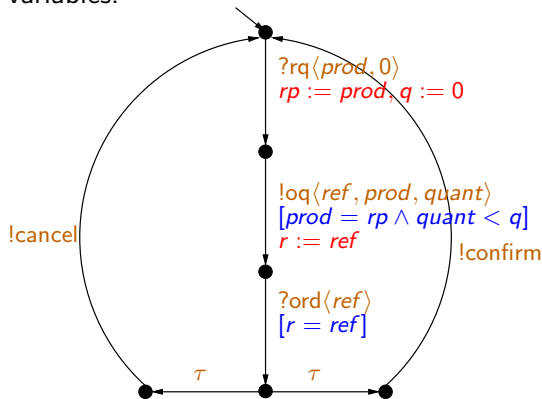
Symbolic Quiescence

Depends on the existence of proper interaction variables to enable transition, which depends on the values of previous interaction variables.



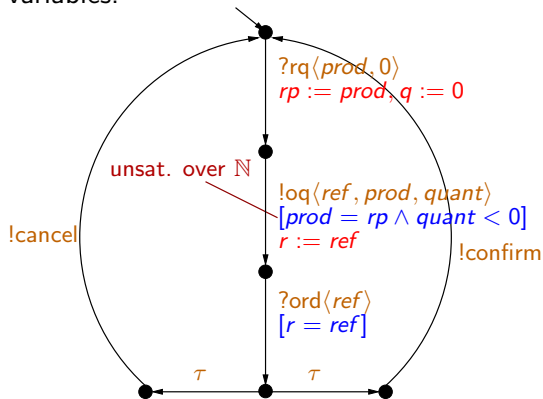
Symbolic Quiescence

Depends on the existence of proper interaction variables to enable transition, which depends on the values of previous interaction variables.



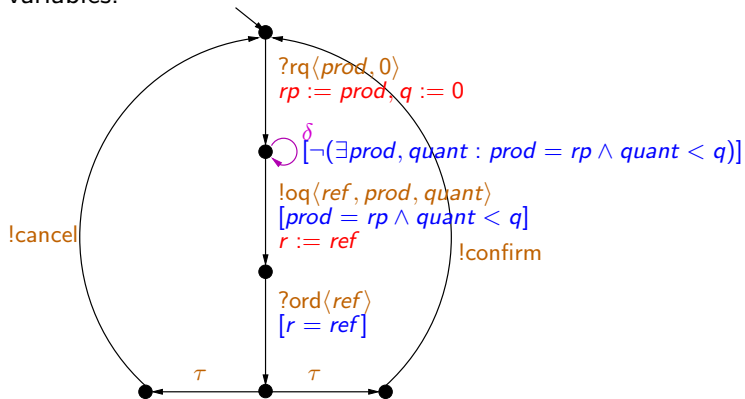
Symbolic Quiescence

Depends on the existence of proper interaction variables to enable transition, which depends on the values of previous interaction variables.



Symbolic Quiescence

Depends on the existence of proper interaction variables to enable transition, which depends on the values of previous interaction variables.



Variations of **tio**co

Active research domain:

- Bohnenkamp *et al.*, T-TorX (TA)
- Larsen *et al.*, TRON (based on UPPAAL) (TA)
- Krichen *et al.*, TTG (TA)
- Brandon Briones *et al.* (TTS)
- ...

Common challenges:

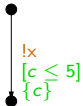
- Quiescence
- Practical Implementations
- Timed Automata (e.g. urgency)
- ...

Urgency in Timed Automata

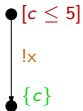
According to TA semantics, transitions *must* or *may* be taken.
There are two main possibilities to express urgency:

- Deadlines/Urgency Predicates
 - Conceptually nicer
 - No timed deadlock
 - Non-convex zones
- Location invariants
 - Efficient implementations
 - Less restrictive
 - Potential timed deadlock
- Issue
 - Urgency and Quiescence

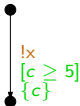
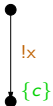
Urgency in Timed Automata: Examples



“the system **may** output an x within 5 time units, or no output is ever produced”

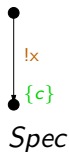


“the system **must** output an x within 5 time units”



“the system **may** output an x (after 5 time units) **at any time**, or **never produce an output**”

Urgency and Quiescence



Spec: possible to delay !x forever
Imp: never produce an output

- Case 1: *Imp* is conformed to *Spec*
 - Theory **rtioco** from Krichen *et al.*
(extended with quiescence)
 - Issue: compatibility with the original **ioco** theory
- Case 2: *Imp* is not conformed to *Spec*
 - *Spec* and *Imp* as LTS, agreement with **ioco**
 - Theory **tioco** of Brandan-Briones *et al.*
 - Issue: output **must** be produced

Quiescence: Definitions

Definition (Timed Quiescence *a la ioco*)

A state is quiescent iff there is no enabled output or τ -transition, **now and in the future**.

Definition (Timed Quiescence (T-TorX))

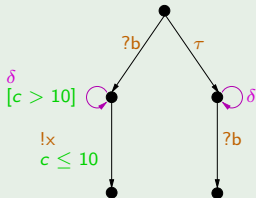
A state is quiescent iff there is no state reachable by τ -steps or by delaying, where a transition with an output label is enabled.

Quiescence: Definitions and their Consequences

Definition (Timed Quiescence *a la* **io**co)

A state is quiescent iff there is no enabled output or τ -transition, **now and in the future**.

Example

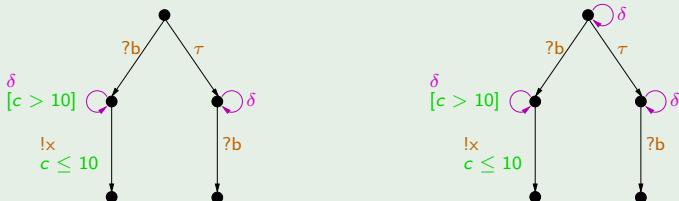


Quiescence: Definitions and their Consequences

Definition (Timed Quiescence (T-TorX))

A state is quiescent iff there is no state reachable by τ -steps or by delaying, where a transition with an output label is enabled.

Example



Outline

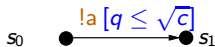
- 1 The **ioco** theory and its variations
 - The original **ioco** theory
 - Symbolic **ioco**
 - Timed **ioco**
- 2 Timed and Symbolic Models
 - Flow between symbolic and timed models
 - Symbolic Timed Automata: Syntax and Semantics
 - Quiescence

Combining Data and Time

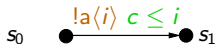
- Timed Automata
 - Clocks: **continuous** variables
 - Clock constraints
 - Clock invariants
- Symbolic Transition Systems
 - **Discrete** variables
 - Interaction variables associated with labels
 - Location variables
 - Symbolic guards

Interaction between time and data: restrictions

We should restrict the following cases:

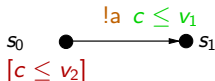


Interaction between clock and location variables

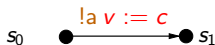


Clocks guarded by interaction variables

But, we should allow:

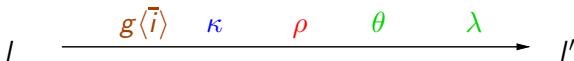


Using integer loc. var. in clock guards and invariants

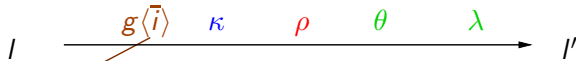


Storing/Sending time stamps of events

Symbolic Timed Automata: Syntax



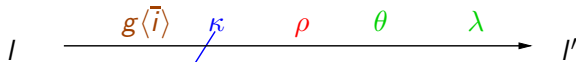
Symbolic Timed Automata: Syntax



Gate g
Interaction Variables \bar{i}
Interaction with environment

Ex: $?msg\langle m, n \rangle$

Symbolic Timed Automata: Syntax



Key κ
First order formula over
interaction and location variables

Ex: $[m == \text{start} \wedge n == 1]$

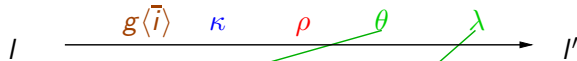
Symbolic Timed Automata: Syntax



Update ρ
Assign first order terms
over inter., loc. and clock variables
to loc. variables

Ex: $v_1 := n + v_2$ or $v_1 := n + c$

Symbolic Timed Automata: Syntax

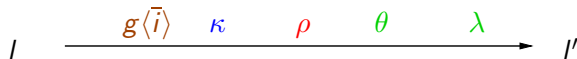


Clock guards θ
Boolean expression btw.
Formula over loc. variables
and clock constraints

Clock reset λ
Set of clocks to be reset

Ex: $v_1 \leq v_2 \rightarrow c \leq v_3, \{c\}$

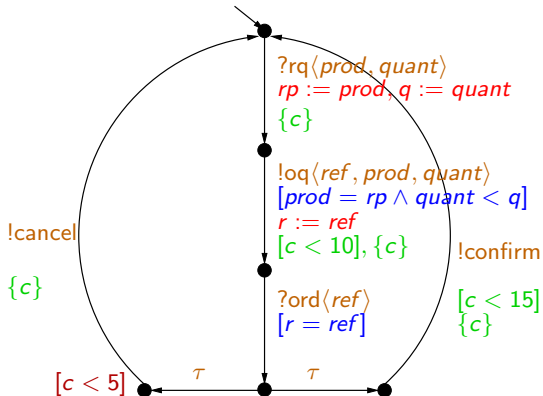
Symbolic Timed Automata: Semantics



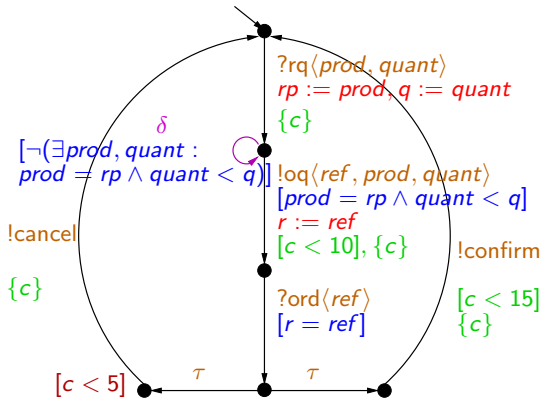
Semantics is given by a Timed Transition System:

- States are triple composed of a location, valuations for the location and clock variables
- Transitions enabled if the clock guard, the key, and the destination invariant are satisfied
- A location must be left if its invariant is not satisfied, time can pass otherwise
- Input/Output: $\mathcal{G} = \mathcal{G}_I \cup \mathcal{G}_U$
 - \mathcal{G}_I are input gates,
 - \mathcal{G}_U are output gates

stioco supplier

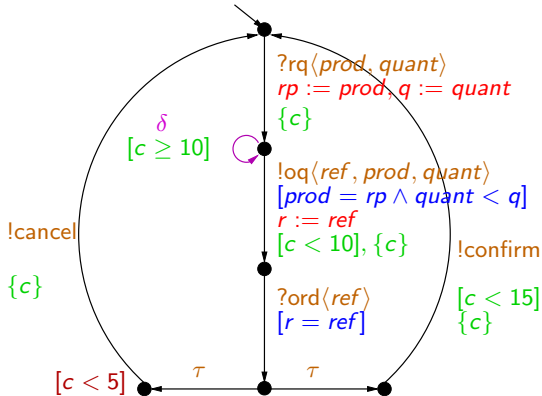


Quiescence for STIOA: Symbolic Quiescence



Quiescence if keys are unsatisfiable

Quiescence for STIOA: Timed Quiescence



Quiescence if keys are unsatisfiable or if clock guards are false

Conclusion

- Identified challenges/issues
 - Urgency
 - Quiescence
 - ...
- Definition for Timed and Symbolic Models
 - Symbolic Timed Input Output Automata
- Future Work
 - Define a testing theory for STIOA
 - Algorithms/Implementation

There are many issues and challenges ... find solutions !



Further Reading

-  L. Frantzen, J. Tretmans and T.A.C. Willemse “A Symbolic Framework for Model-Based Testing”, FATES/RV 2006
-  L. Brandan Briones, “Theories for Model-Based Testing: Real-Time and Coverage”, PhD Thesis, University of Twente, The Netherlands, 2007
-  B. Gebremichael and F. Vaandrager, “Specifying Urgency in Timed I/O Automata”, SEFM 2005