Software Engineering for Robotics

On the occasion of Jan Peleska's Festschrift

Ana Cavalcanti

robostar.cs.york.ac.uk

February, 2023

Testing in RoboStar - a journey

Model-based testing using CSP and CSP-like notations

Early days

- Jan is a pioneer: theory and tool
- Testing for refinement in CSP: inspired on his tests for negative behaviour
- Developers and testers using the same conformance relation
- Data-rich theory: *Circus*
- Inputs and outputs
- Refusal-traces semantics

Testing in RoboStar - a journey

With Jan, and colleagues

- Kripke structures and the UTP
- Finite and complete test suites

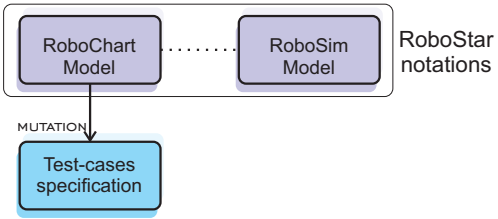
Recently and Now

- Testing theory for *tock*-CSP
- *Using RT-Tester to run simulation and tests*
- RoboStar notations as a front end
- With Alvaro Miyazawa, Uwe Schulze, and Jon Timmis

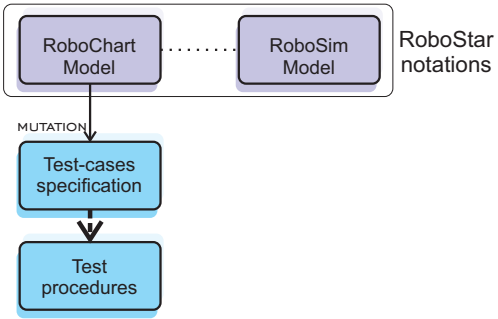
RoboStar and RT-Tester

RoboStar and RT-Tester

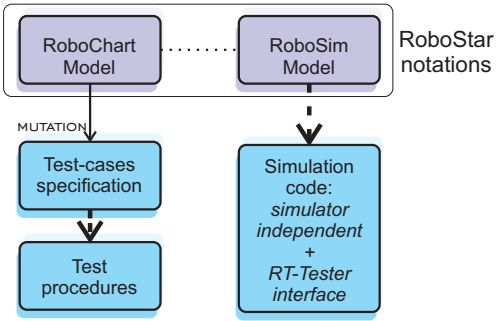
RoboStar and RT-Tester



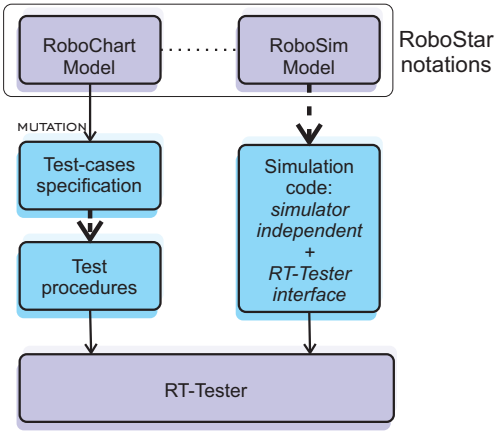
RoboStar and RT-Tester



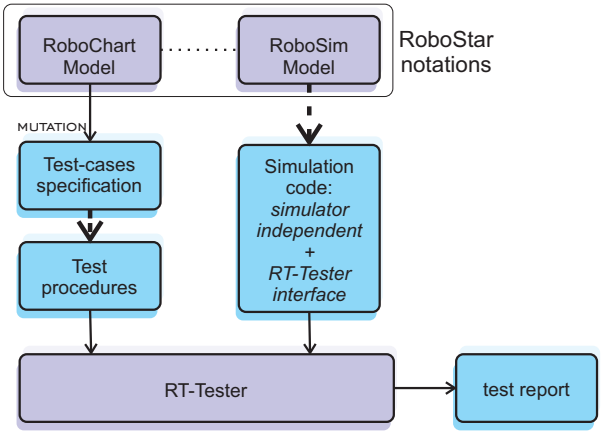
RoboStar and RT-Tester



RoboStar and RT-Tester



RoboStar and RT-Tester



Software Engineering for Robotics: others are interested

Recently published book

`https://www.springer.com/gp/book/9783030664930`

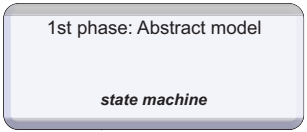
Our distinctive approach

- Complement existing techniques
- Use of domain-specific languages
- Mathematics for validation and verification

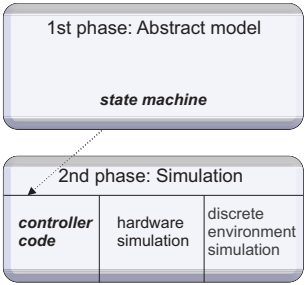
Largest research group in the world in the area

Current approach

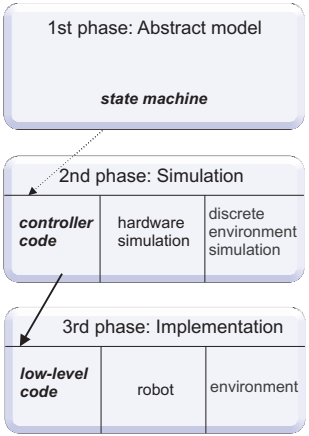
Current approach



Current approach

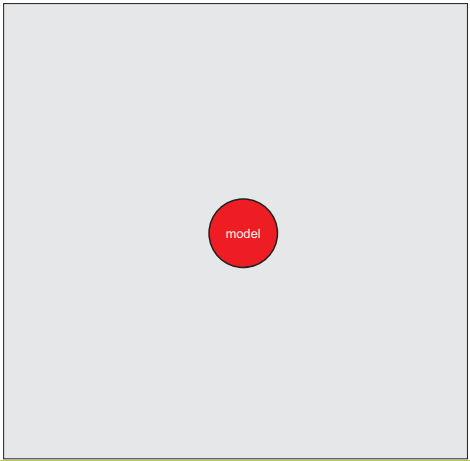


Current approach

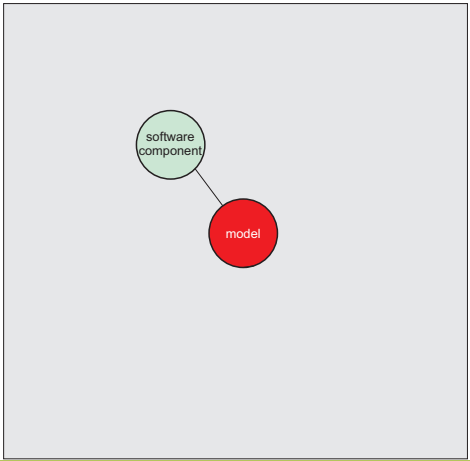


RoboStar vision

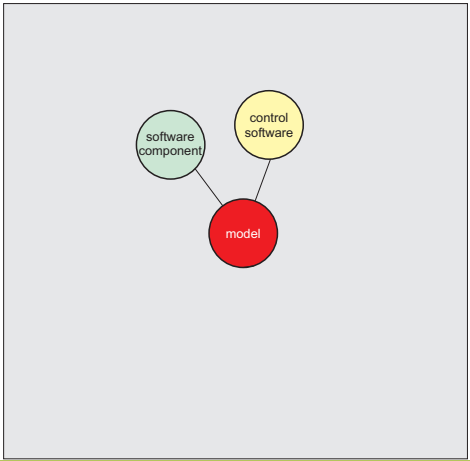
RoboStar vision



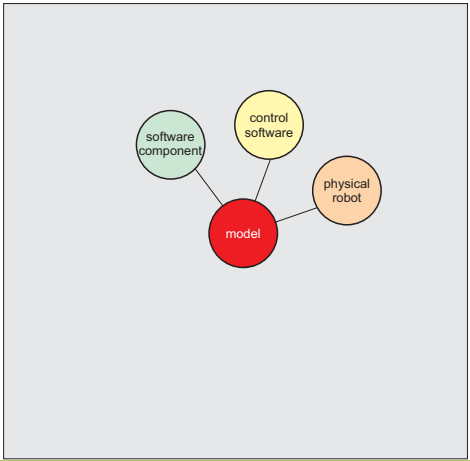
RoboStar vision



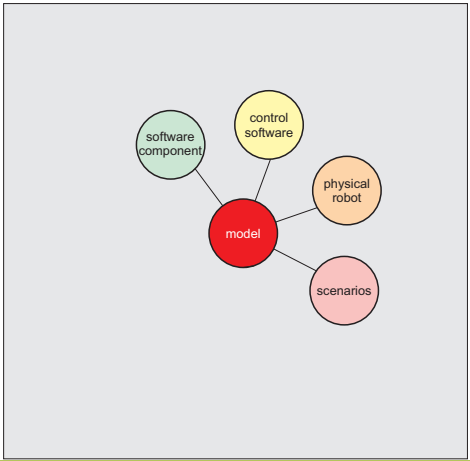
RoboStar vision



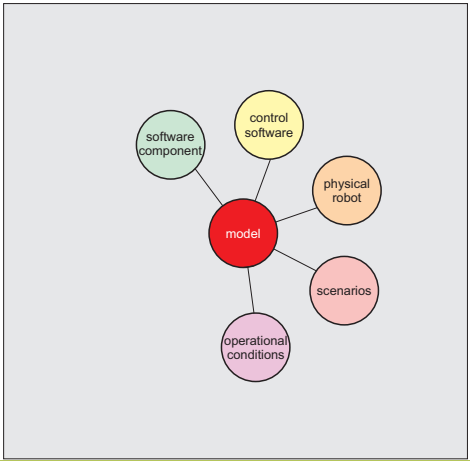
RoboStar vision



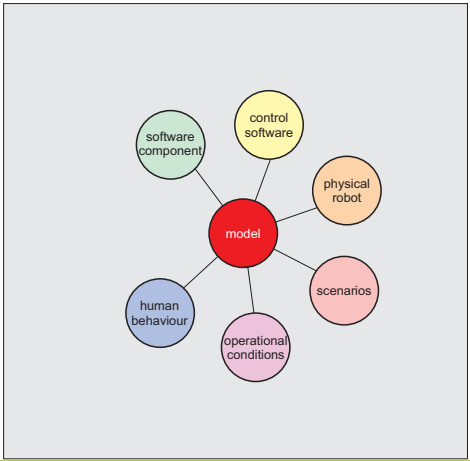
RoboStar vision



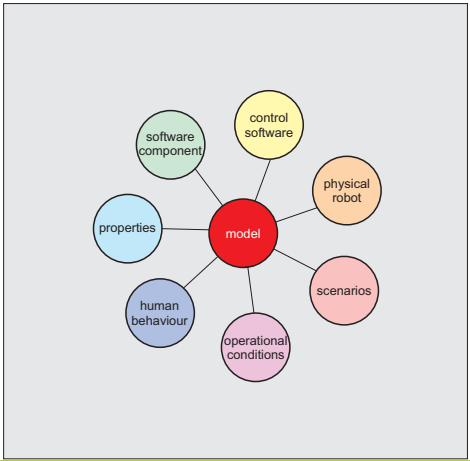
RoboStar vision



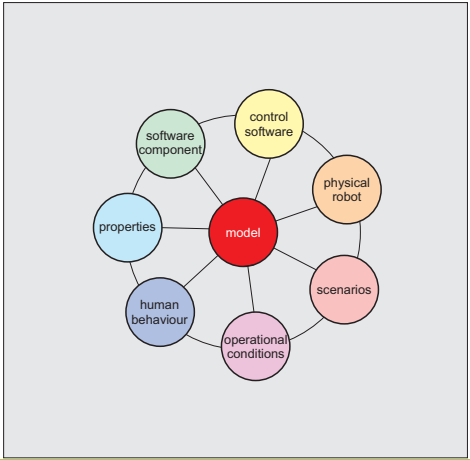
RoboStar vision



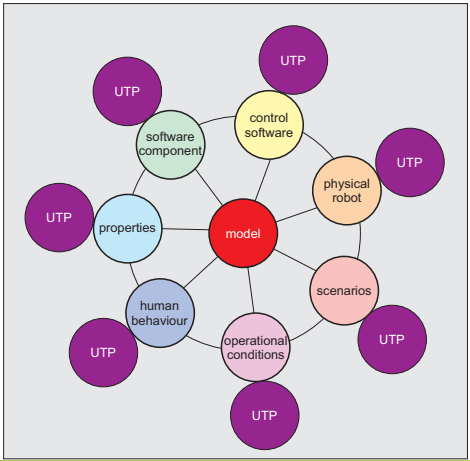
RoboStar vision



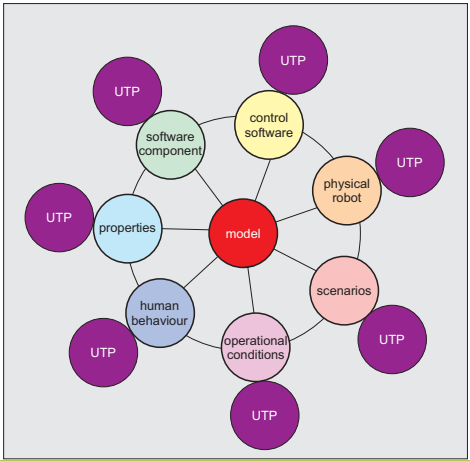
RoboStar vision



RoboStar vision



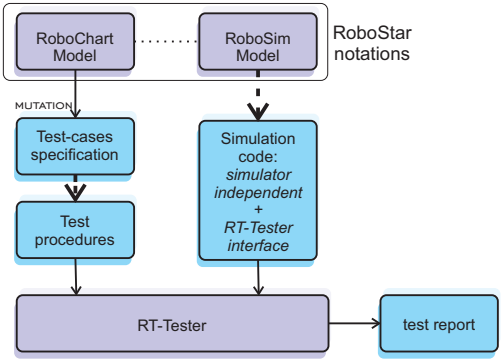
RoboStar vision



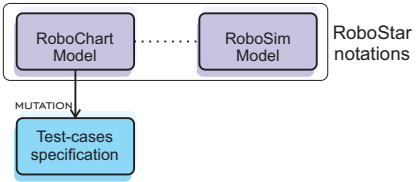
1. Simulation
2. Code
3. Tests
   - simulation
   - deployment
4. Proof
   - model checking
   - theorem proving
5. Evidence of properties

Testing approach

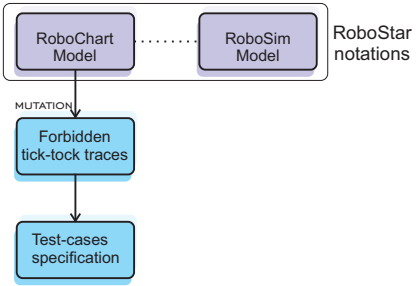
Testing approach



Testing approach



Testing approach



tock-CSP: an overview

Timed process algebra

- Mechanisms modelled by processes that communicate via atomic and instantaneous events
- Special event: *tock*
- Reuse of tools
- Reuse of semantics
- Several challenges
  - Passage of time cannot decide choice
  - Time interrupt is not strict
- FDR support

tock-CSP requirements

- Deadlines: refusal of *tock*
- Termination
- Within each time unit, the standard (untimed) failures semantics of CSP holds.
- Zeno behaviours
- Timewise refinement
- Operators
  - Events are instantaneous: passage of time explicit
  - Maximal progress of internal events
  - Operators as defined in a timed section

✓-tock traces

Example: *RD*

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

✓-tock traces

Example: *RD* – Traces

$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$

$\Box$

$turnoff \rightarrow \mathbf{Skip}$

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle\rangle$,

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle takeoff \rangle$,

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle takeoff \rangle, \langle turnoff \rangle,$

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\} \rangle,$$

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\} \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff, move\} \rangle,$
$\ldots,$
$\langle \varnothing \rangle,$

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock \rangle,$

✓-tock traces

Example: *RD* – Traces

$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$
$\Box$
$turnoff \rightarrow \mathbf{Skip}$

$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, takeoff \rangle,$

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, takeoff \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, \ldots \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, \ldots, takeoff \rangle,$

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$$\langle takeoff, \Sigma^{\checkmark} \rangle,$$

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle takeoff, \Sigma^{\checkmark} \rangle, \langle takeoff, \Sigma^{\checkmark}, tock \rangle,$

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle takeoff, \Sigma^{\checkmark} \rangle, \langle takeoff, \Sigma^{\checkmark}, tock \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, move \rangle,$

✓-tock traces

Example: *RD* – Traces

$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$

$\Box$

$turnoff \rightarrow \mathbf{Skip}$

$\langle takeoff, \Sigma^{\checkmark} \rangle, \langle takeoff, \Sigma^{\checkmark}, tock \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, move \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, \Sigma^{\checkmark} \setminus \{move\} \rangle, \langle takeoff, \Sigma^{\checkmark}, tock, \Sigma^{\checkmark} \setminus \{move\}, tock \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, \Sigma^{\checkmark} \setminus \{move\}, tock, \ldots, move \rangle,$

✓-tock traces

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle turnoff, \checkmark \rangle$,
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, turnoff, \checkmark \rangle$,
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, \ldots, turnoff, \checkmark \rangle$

✓-tock traces - with inputs and outputs

Example: *RD*

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$

$\Box$

$turnoff \rightarrow \mathbf{Skip}$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle\,\rangle$,

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle takeoff \rangle$,

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle takeoff \rangle, \langle turnoff \rangle,$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

takeoff $\rightarrow$ *wait* 1; *move* $\rightarrow$ *found* $\rightarrow$ *land* $\rightarrow$ **Stop**

$\Box$

turnoff $\rightarrow$ **Skip**

$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\} \rangle$,

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\} \rangle$,
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff, move\} \rangle$,
$\ldots$,
$\langle \varnothing \rangle$,

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock \rangle,$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, takeoff \rangle,$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, takeoff \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, \ldots \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, \ldots, takeoff \rangle,$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$$\langle takeoff, \Sigma^{\checkmark} \rangle,$$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$(\mathit{takeoff} \rightarrow \mathit{wait}\ 1;\ \mathit{move} \rightarrow \mathit{found} \rightarrow \mathit{land} \rightarrow \mathbf{Stop}) \ \Box\ \mathit{turnoff} \rightarrow \mathbf{Skip}$$

$\langle \mathit{takeoff}, \Sigma^{\checkmark} \rangle, \langle \mathit{takeoff}, \Sigma^{\checkmark}, \mathit{tock} \rangle,$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle takeoff, \Sigma^{\checkmark} \rangle, \langle takeoff, \Sigma^{\checkmark}, tock \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, move \rangle,$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle takeoff, \Sigma^{\checkmark} \rangle, \langle takeoff, \Sigma^{\checkmark}, tock \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, move \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, \Sigma^{\checkmark} \setminus \{move\} \rangle, \langle takeoff, \Sigma^{\checkmark}, tock, \Sigma^{\checkmark} \setminus \{move\}, tock \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, \Sigma^{\checkmark} \setminus \{move\}, tock, \ldots, move \rangle,$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$

$\Box$

$turnoff \rightarrow \mathbf{Skip}$

$\langle takeoff, \Sigma^{\checkmark} \rangle, \langle takeoff, \Sigma^{\checkmark}, tock \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, move \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, \Sigma^{\checkmark} \setminus \{move\} \rangle, \langle takeoff, \Sigma^{\checkmark}, tock, \Sigma^{\checkmark} \setminus \{move\}, tock \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, \Sigma^{\checkmark} \setminus \{move\}, tock, \ldots, move \rangle,$
$\langle takeoff, \Sigma^{\checkmark}, tock, move, found \rangle, \langle takeoff, \Sigma^{\checkmark}, tock, move, \Sigma^{\checkmark} \setminus \{found\}, tock, found \rangle,$

✓-tock traces - with inputs and outputs

Example: *RD* – Traces

$$takeoff \rightarrow wait\ 1;\ move \rightarrow found \rightarrow land \rightarrow \mathbf{Stop}$$
$$\Box$$
$$turnoff \rightarrow \mathbf{Skip}$$

$\langle turnoff, \checkmark \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, turnoff, \checkmark \rangle,$
$\langle \Sigma^{\checkmark} \setminus \{takeoff, turnoff\}, tock, \ldots, turnoff, \checkmark \rangle$

tock-CSP with inputs and outputs: Main results

Given processes P and Q such that Q is input-enabled, $P \sqsubseteq_{IOTT} Q \Rightarrow Q \text{ tioco } P$.

There are P and Q such that $Q \text{ tioco } P$, but not $P \sqsubseteq_{IOTT} Q$.

Given input-enabled processes P and Q, $P \sqsubseteq_{IOTT} Q \Leftrightarrow Q \text{ tioco } P$.

tock-CSP Testing Theory

Main components

- Verdict events: *inc*, *pass*, *fail*
- SUT is divergence free
- Test execution: SUT behaves like an unknown process *SUT*

$$Execution(SUT, TCS) \mathrel{\widehat=} ((SUT;\ ticktest \to_U \mathbf{Stop}_U) \,[\![\, \Sigma \,]\!]\, TCS) \setminus \Sigma$$

- Fairness condition: all behaviours can be observed
- Verdict

$$SUT\ fails\ TCS \mathrel{\widehat=} \exists\, \rho : TTTrace \bullet \rho \frown \langle evt\ fail, ref\ (\Sigma^{\checkmark}_{tock} \cup V)\rangle \in tt[\![Execution(SUT, TCS)]\!]$$

tock-CSP Testing Theory

Example: *RD*

$$\mathit{takeoff} \rightarrow \mathit{wait}\,1;\ \mathit{move} \rightarrow \mathit{found} \rightarrow \mathit{land} \rightarrow \mathbf{Stop}$$
$$\Box$$
$$\mathit{turnoff} \rightarrow \mathbf{Skip}$$

Forbidden trace: $\langle \mathit{takeoff}, \Sigma^{\checkmark}, \mathit{tock}, \mathit{found} \rangle$

$$\mathit{inc} \rightarrow_U \mathit{takeoff} \rightarrow_U \mathit{inc} \rightarrow_U (\ \mathit{takeoff} \rightarrow \mathbf{Stop}_U \ \Box\ \mathit{move} \rightarrow \mathbf{Stop}_U \ \Box\ \mathit{found} \rightarrow \mathbf{Stop}_U$$
$$\Box$$
$$\mathit{land} \rightarrow \mathbf{Stop}_U \ \Box\ \mathit{turnoff} \rightarrow \mathbf{Stop}_U \ \Box\ \mathit{ticktest} \rightarrow \mathbf{Stop}_U\ )$$
$$\triangle_1\ \mathit{pass} \rightarrow_U \mathit{found} \rightarrow_U \mathit{fail} \rightarrow_U \mathbf{Stop}_U$$

tock-CSP Testing Theory: Main results

Soundness

The test-case specifications formed from forbidden traces that are minimal with respect to a prefixing relation and are output saturated are sound.

Completeness

The test suite containing all test-case specifications formed from forbidden traces that are minimal with respect to a prefixing relation and are output saturated is complete.

Current work: testing with robot in the loop

Hybrid model

- RoboChart + RoboSim p-model: *tock*-CSP + *CyPhyCircus*
- Assumptions:
  1. Forbidden event traces of the software are available.
  2. The software has been verified: at the very least, tests corresponding to the forbidden event traces in a have passed
- Approach: convert the forbidden traces to traces described in terms of inputs of the sensors and outputs of the actuators
- A continuous test contains: trajectories for the inputs and outputs, and verdicts and as function of time.
- For a test drive: project the continuous test over the sample time.

Conclusions

Thanks

- Discrete time in CSP sorted out
- If events are atomic and instantaneous
- Testing theory in place: inputs, outputs, soundness, completeness, ...
- Termination, nondeterminism
- RoboStar notations: practical front-end
- Ongoing: automation