Mutant Equivalence as Monotonicity in Parametric Timed Games

Davide Basile, **Maurice ter Beek** (ISTI-CNR, Pisa, Italy),

Hendrik Göttmann (TU Darmstadt, Germany) &

Malte Lochau (University of Siegen, Germany)

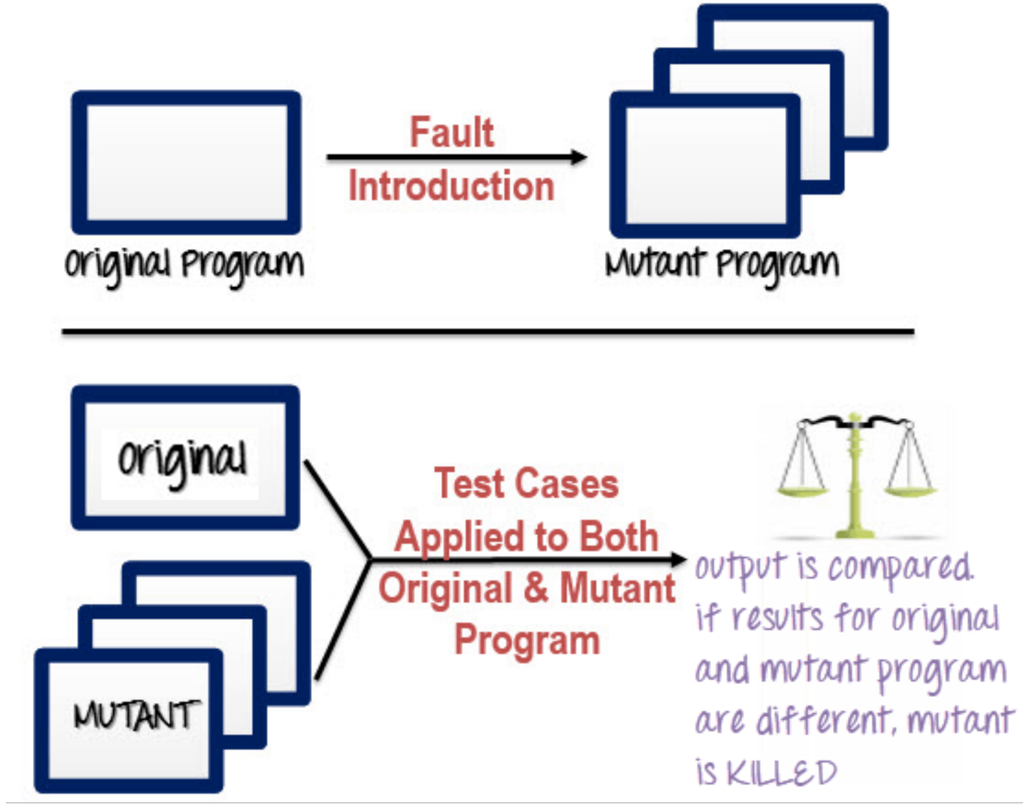
Istituto di Scienza e Tecnologie dell'Informazione "A. Faedo" Consiglio Nazionale delle Ricerche

Formal Methods and Tools Lab

Overview

- Mutation testing
- Model-based testing
- Model-based mutation testing
- Equivalent mutant problem
- Mutant equivalence as monotonicity of parametric timed games (PTG)

Mutation Testing



$$\text{Mutation score} = \frac{\text{Number of killed mutants}}{\text{Total number of mutants (survived and killed)}} * 100\%$$

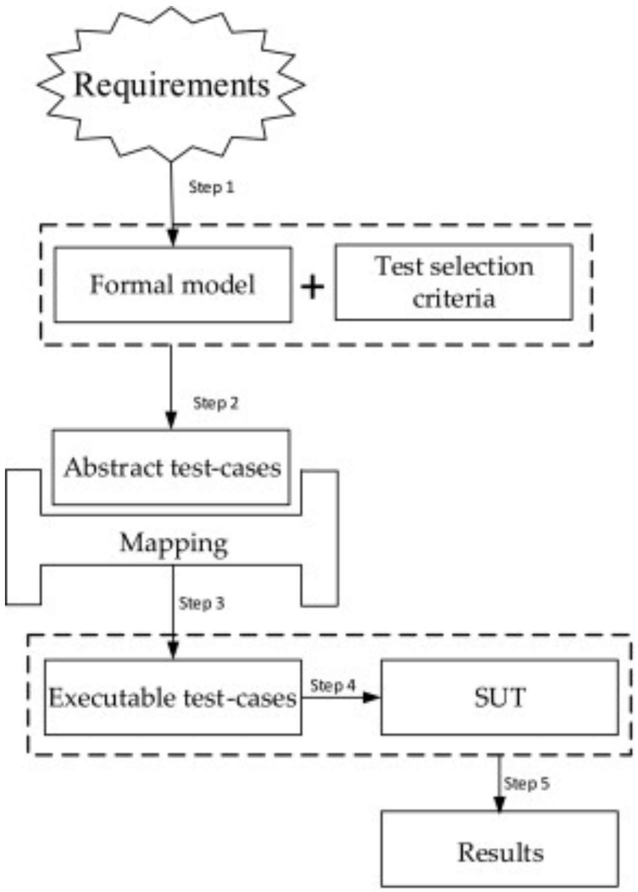
Examples of Mutation Testing

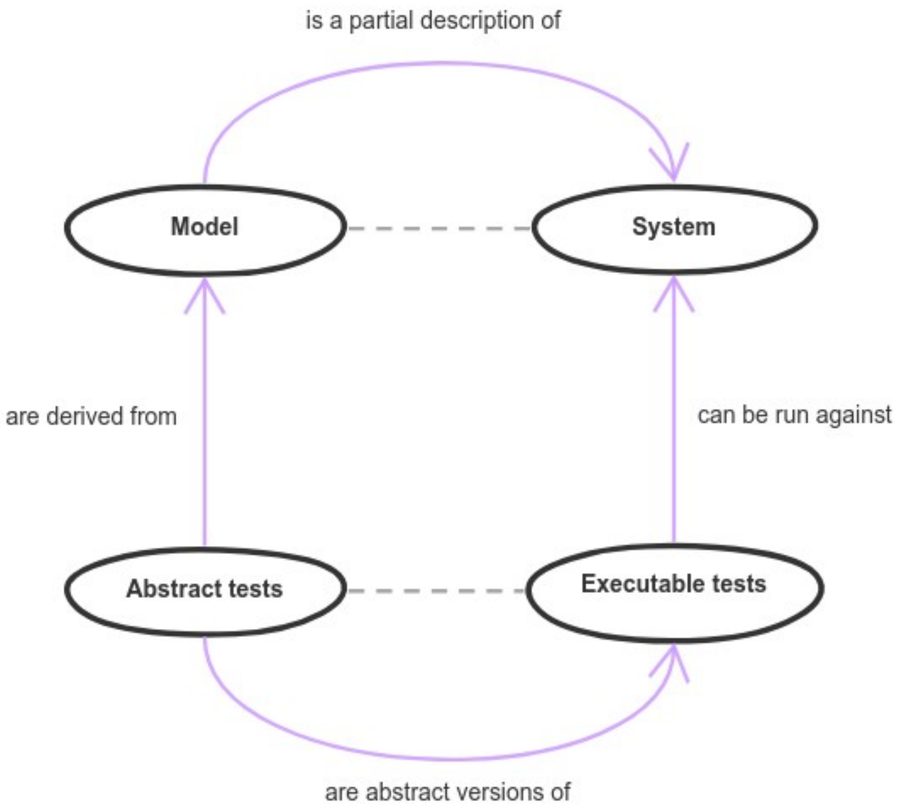
```
if (a && b) {
    c = 1;
} else {
    c = 0;
}
```

```
if (a || b) {
    c = 1;
} else {
    c = 0;
}
```

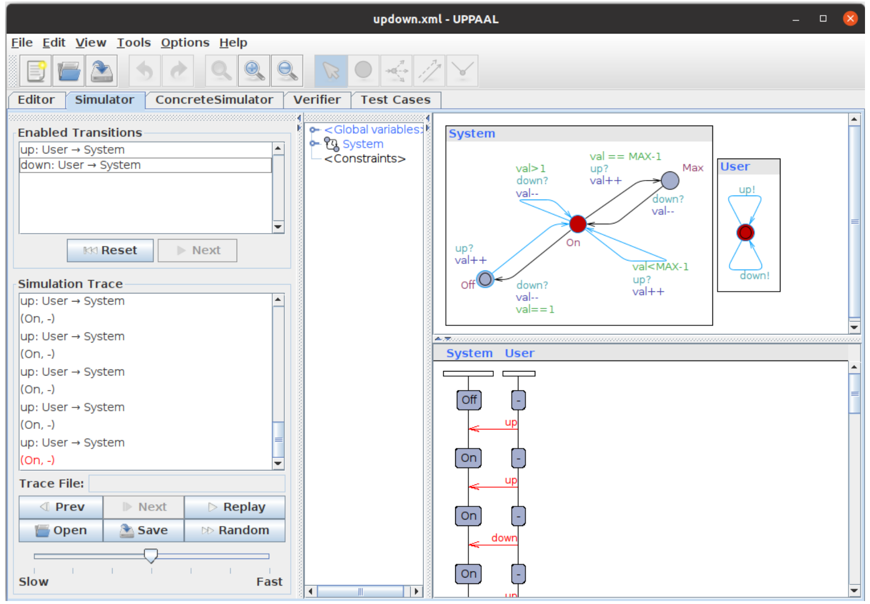
Further examples:	
	Replacement of some Boolean relations with others, such as > with >=, == and <=
	Replacement of Boolean subexpressions with true and false
	Replacing returned value with null

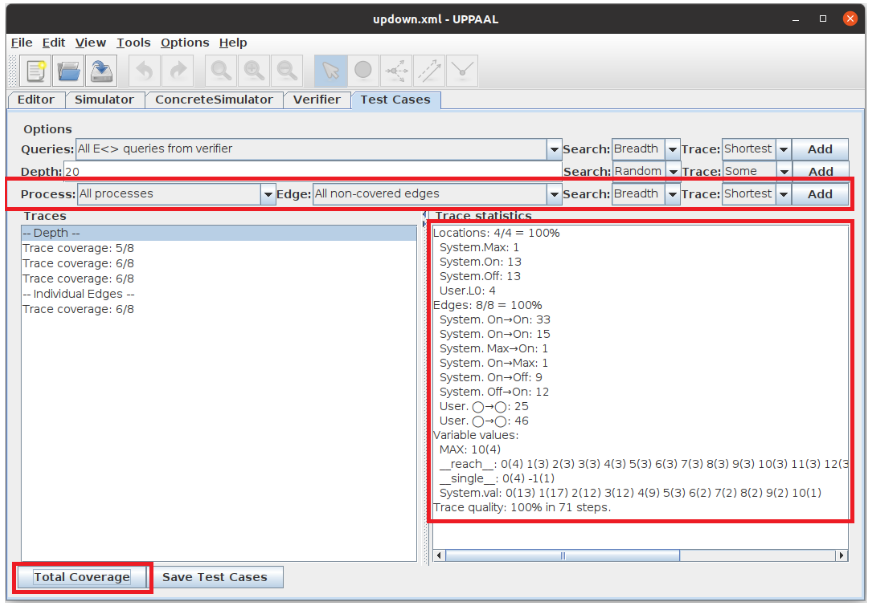
Model-based Testing





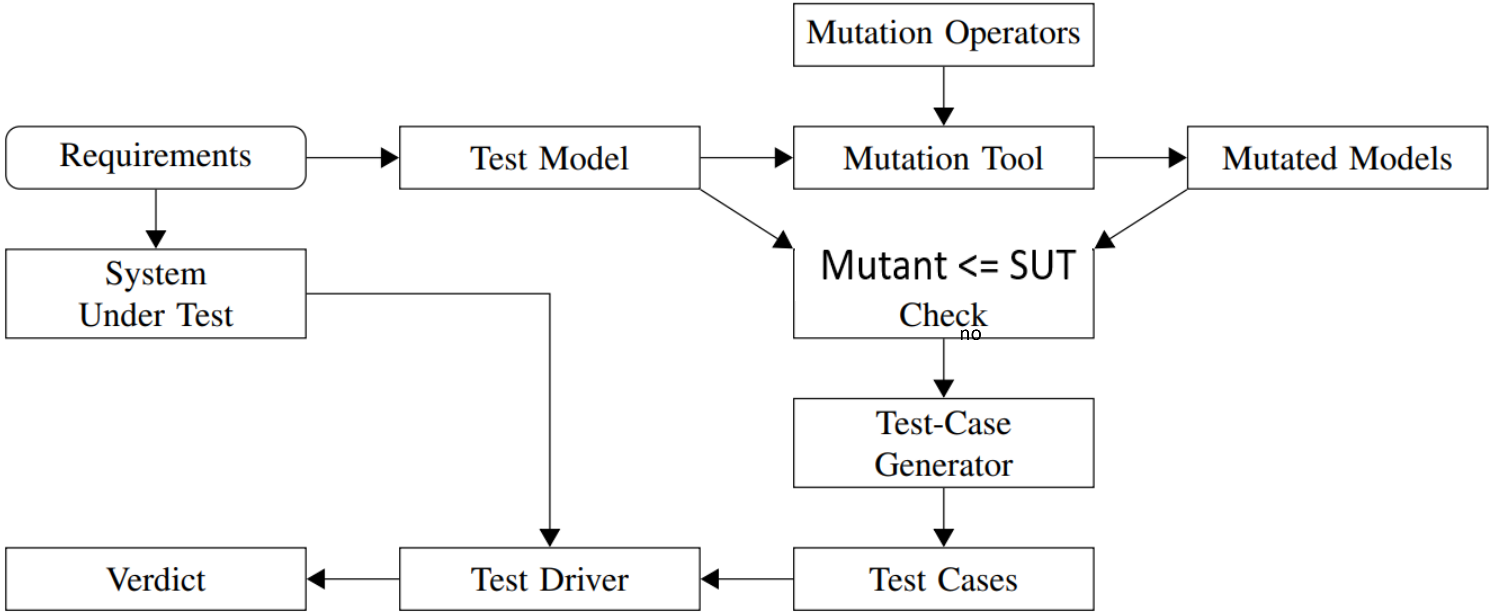
Model-based Testing with Uppaal (example)

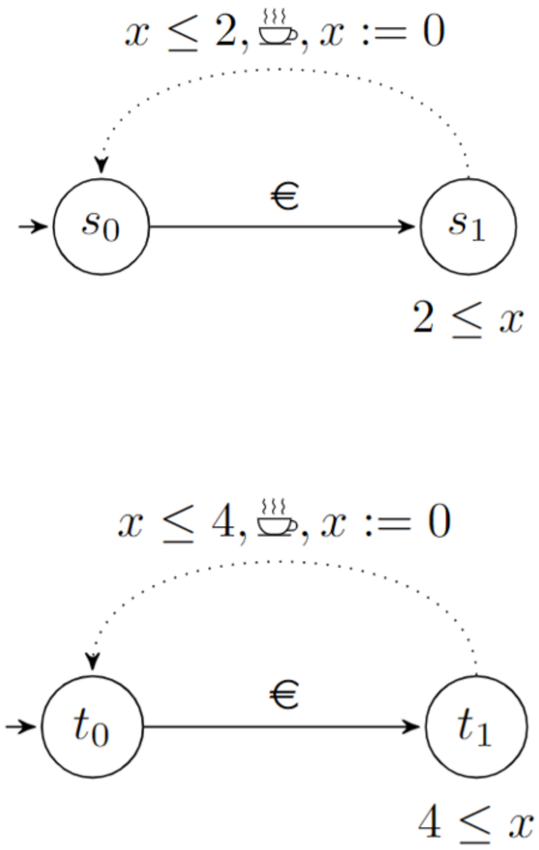





```java
package app;
import app.App;

class Test extends App {

    public static void main(String[] args) {


expect_off();

up();

expect_on(1);

up();

expect_on(2);

up();

expect_on(3);

down();

expect_on(2);

up();
  <... snip ...>
    }
}
```

Model-based Mutation Testing





If Mutant <= SUT, then Mutant is *subsumed* by SUT:

- the Mutant can mimic all the controllable (i.e., input) behaviour of the SUT
- the SUT can mimic all the uncontrollable (i.e., output, delay) behaviour of the Mutant

K.G. Larsen et al., Mutation-based test-case generation with Ecdar. ICSTW 2017

Mutant <=? SUT

Equivalent Mutant Problem

- Mutants may exhibit the same observable external behaviour
- From a tester's point of view, the subsumed (i.e., equivalent) Mutant is not distinguishable from the original system
- No test case can be generated to differentiate the Mutant from the original system:
  - useless analyses and a waste of computational resources

Static Detection of Equivalent Mutants

Mutation operators:

- **TMI** – Transition MIssing
- **TAD** – Transition ADd
- **SMI** – State MIssing
- **CXS** – Constant eXchange S decrease
- **CXL** – Constant eXchange L increase

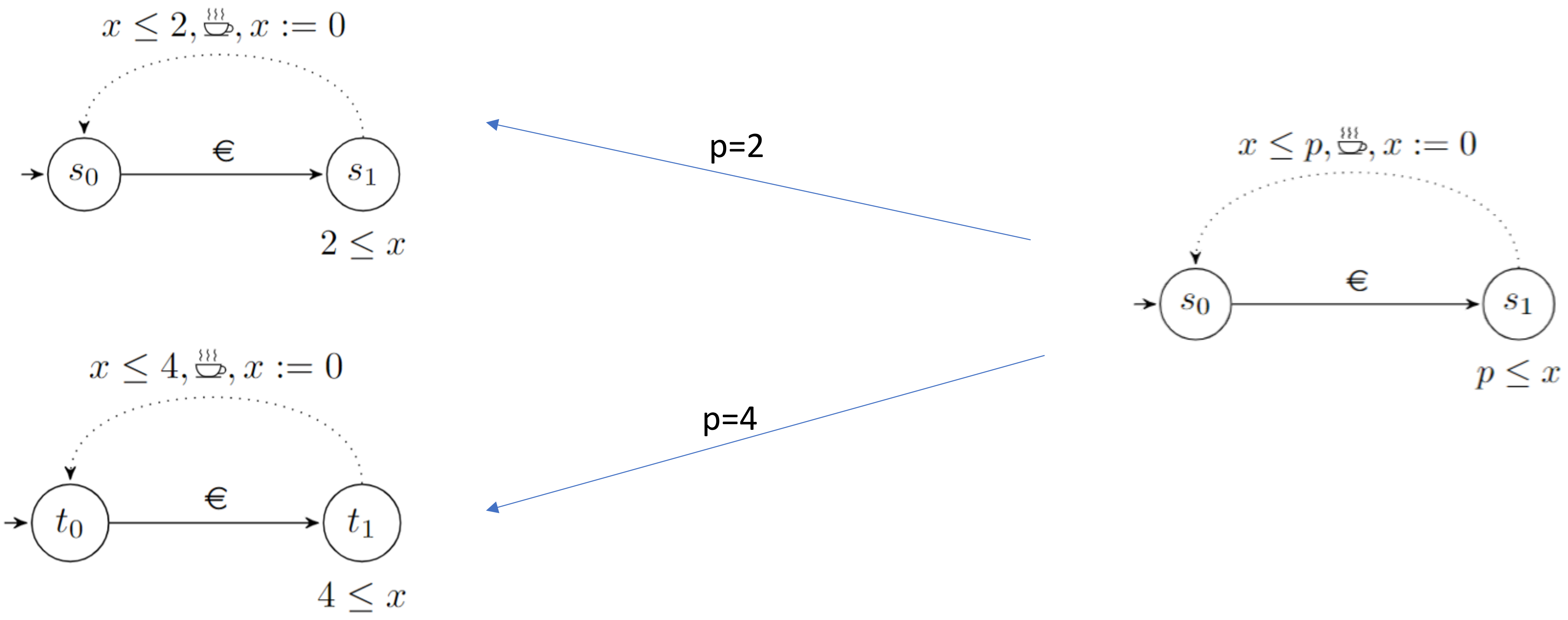
The 10 Commandments of Model-Based Mutation Testing:

(1) TMI shall not be applied to uncontrollable transitions;
(2) TAD shall not be applied to controllable transitions;
(3) SMI shall not be applied if all incoming transitions are uncontrollable;
(4) CXL shall not be applied to controllable transitions with guards of the form $x \leq k$;
(5) CXL shall not be applied to uncontrollable transitions with guards of the form either (i) $x \geq k$ or (ii) $x == k$, and source invariant $x \leq k$;
(6) CXL shall not be applied to invariants of the form $x \geq k$ whenever all incoming transitions are uncontrollable;
(7) CXS shall not be applied to controllable transitions with guards of the form $x \geq k$;
(8) CXS shall not be applied to uncontrollable transitions with guards of the form $x \leq k$;
(9) CXS shall not be applied to invariants of the form $x \leq k$ whenever all incoming transitions are uncontrollable;
(10) CXS shall not be applied to invariants of the form $x \geq k$ whenever all incoming transitions are controllable.

D. Basile et al., Static detection of equivalent mutants in real-time model-based mutation testing: An Empirical Evaluation. EMSE, 2022

Contribution

Commandments [4-10] = Monotonicity of *Extended* L/U PTG



L/U PTG:

A. Jovanović et al., A game approach to the parametric control of real-time systems. Int. J. of Control, 2019

Extended

Theorem 1 (Monotonicity of L/U parametric timed games): Let $ptg = (L, \ell_0, \Sigma, C, P, I, E)$ be an L/U parametric timed game, let σ_P, σ'_P be such that $\forall p_+ \in P^+, p_- \in P^-. \sigma'_P(p_+) \geq \sigma_P(p_+)$ and $\sigma'_P(p_-) \leq \sigma_P(p_-)$. Then the following holds:

$$\pi(\sigma'_P, ptg) \preceq \pi(\sigma_P, ptg)$$

Decidability of refinement for (extended) L/U PTG (undecidable in general for PTG) relies on the property of monotonicity

É. André, What's decidable about parametric timed automata? STTT, 2019

Extended L/U PTG

- Parameters partitioned into P^+ and P^-
- P^- is related to the commandments for CXS [7-10]
- Parameters in P^- only occur as
  - Lowerbound of controllable transitions (comm. 7)
    (we can increase the controllable behaviour in a refinement)
  - Upperbound of uncontrollable transitions (comm. 8)
    (we can decrease the uncontrollable behaviour in a refinement)
  - Lowerbound of the invariant of a location ℓ whenever all incoming transitions are controllable (comm. 10)
  - Upperbound of the invariant of a location ℓ whenever all incoming transitions are uncontrollable (comm. 9)

Extended L/U PTG

- Parameters partitioned into P^+ and P^-
- P^+ is related to the commandments for CXL [4-6]
- Parameters in P^+ only as
  - Lowerbound of uncontrollable transitions (comm. 5.i)

    (we can decrease the uncontrollable behaviour in a refinement)
  - Upperbound of controllable transitions (comm. 4)

    (we can increase the controllable behaviour in a refinement)
  - Lowerbound of the invariant of a location ℓ whenever all incoming transitions are uncontrollable (comm. 6)

Generating a test-suite for a timed game

1) Let *tg* be a timed game such that $ptg = \pi^{-1}(tg)$ is the abstracted parametric timed game, and σ_P is the parameter evaluation such that $tg = \pi(\sigma_P, ptg)$. Repeat the following steps.
2) Let $m : \Delta^P \mapsto \Delta^P$ be a *mutation* and let $\sigma_{P'} = m(\sigma_P)$ be the *mutant*.
3) Let $\pi(\sigma'_P, tg)$ be *subsumed* by $\pi(\sigma_P, ptg)$ if and only if $\pi(\sigma'_P, ptg) \preceq \pi(\sigma_P, ptg)$, and let *EquMut* be the set of mutation functions yielding subsumed mutants. **Select $m \notin$ *EquMut*.**
4) Add $Ctx(\pi(\sigma'_P, ptg), \pi(\sigma_P, ptg))$ to the test-suite under construction.

Theorem 2 (Monotonicity as Equivalent Mutants Detection): Let tg be a timed game and let ptg be an L/U parametric timed game obtained from tg using Definition 8 such that $tg = \pi(\sigma_P, ptg)$. Let m be a mutation of ptg, where m is either `CXL` or `CXS`, and let $\sigma'_P = m(\sigma_P)$ be the *mutant*. Consider the 7 commandments in Figure 6. Then the following holds:

m violates one of the 7 commandments if and only if

$$\forall p_+ \in P^+, p_- \in P^-. \sigma'_P(p_+) \geq \sigma_P(p_+) \text{ and } \sigma'_P(p_-) \leq \sigma_P(p_-)$$

Commandments [4-10]

Conclusion and Future Work

- We presented the problem of checking equivalent mutants in model-based mutation testing
- We showed how the violation of 7 out of the 10 commandments for detecting equivalent mutants is related to the monotonicity of a suitable extension of L/U PTG (essential for decidability)
- Future developments: refinement checking on PTG?
  (currently no tooling available)