Automated Temporal Verification for Real-Time Systems
via Implicit Clocks and an Extended Antimirov Algorithm

Yahui Song @ SPLASH–SRC 2022  yahuuis@comp.nus.edu.sg

Overview
To go beyond the existing Timed Automata (TA) based techniques, we propose a novel solution that integrates a modular Hoare-style forward verifier with a term rewriting system (TRS) on Timed Effects (TimEffs). The purposes are to: increase expressiveness, dynamically manipulate clocks, and efficiently solve clock constraints. The main contributions are:

- Language Abstraction, C**: generalizes the real-time systems with mutable variables and timed behavioural patterns.
- Novel Specification, TimEffs: extends regular expressions with dependent values and arithmetic constraints.
- Efficient Term Rewriting System, TRS: solves inclusions between TimEffs, by iterated checking of their derivatives.

TimEffs (Symbolic Timed Automata)

```
1 void addOneSugar()
2 /
3 if (n == 0)
4     void makeCoffee (int n)
5     /
6     /
7 #if: req: true ∧ R
8     /
9     /
10 main int main function
11     /
12     /
13     /
14     /
15     /
16 (Timed Effects) Φ := Π ∧ Θ ∨ Φ₁ ∨ Φ₂
17 (Event Sequences) θ := ⊥ | ε | η + θ₁ + θ₂ | θ₁ + θ₂ | θ₁ ∧ θ₂ | θ₁ ∨ θ₂ | θ₁ ⊕ θ₂ | θ₁ ⊖ θ₂
18 (Real-Time Terms) t := c | x | t₁ + t₂ | t₁ ∗ t₂
19 c ∈ Z, a ∈ var (Real Time Bound) # (Kleene Star) ♦
```

Language Inclusion – the Antimirov Algorithm

Our TRS is an extension of Antimirov and Mosses’ algorithm, which can be deployed to decide the inclusions of two regular expressions (REs) through an iterated process of checking the inclusions of their partial derivatives.

Definition 1 (Derivatives). Given any formal language S over an alphabet Σ and any string u ∈ Σ, the derivatives of S w.r.t. u is defined as: uᵢS = {w ∈ Σ | uw ∈ S}.

Definition 2 (Regular Expression Inclusion). For REs r and s,

r ⊆ s ⇔ ∀ A ∈ Σ, Aᵢ(r) ≤ Aᵢ(s).

Definition 3 (TimEffs Inclusion). For TimEffs Φ₁ and Φ₂,

Φ₁ ⊆ Φ₂ ⇔ ∀ Σ ∈ Σ, ∀ r ≥ 0, |Φ₁| (r) ≤ |Φ₂| (r).

Expressiveness of TimEffs

TimEffs draw similarities to Metric Temporal Logic (MTL), derived from LTL, where a set of non-negative real numbers is added to temporal modal operators. Basic operators are:

```
Φₚ = Φ₀, Φ₁ ∨ Φ₂
```

- “globe ally”; o - “finally”; O - “next”; U - “until”, and their past time-reversed versions: <; ♦; and δ for “previous”; S for “since”.

TimEffs in the preconditions, encode past-time temporal specifications. In MTL is the time interval with concrete upper/lower bounds, whereas in TimEffs they can be symbolic bounds, dependent on program inputs.

A Demonstration of the Automated TRS

```
Φ₁ = (s1=0 ∧ s2 ∧ (s1=0 ∧ s2 ∧ s1=0 ∧ s2) ∧ (s1=0 ∧ s2) ∧ (s1=0 ∧ s2)) \[Proof\]
Φ₂ = (s1=0 ∧ s2 ∧ (s1=0 ∧ s2) ∧ (s1=0 ∧ s2) ∧ (s1=0 ∧ s2)) \[UNFOLD\]
Φ₃ = (s1=0 ∧ s2) \[RENAME\]
```

Limitation of Our TRS:

Our TRS is incomplete, meaning there exist valid inclusions which will be disproved in our system. That is mainly because of insufficient unification in favor of achieving automation.

Scan Me For the Project Repository