Temporal Proof Methodologies for Real-time Systems

Authors:
T. A. Henzinger*,
Z. Manna*+, A. Pnueli+

* Department of Computer Science, Stanford University

+Department of Applied Mathematics, Weizman Institute of Science

Source:
Proceedings of POPL Symposium, 1991

Presenter: Nikolay V. Shilov

Date: 24^{th} of May, 2000

ROPAS workshop on Resource Bound Checking

Real-time Transition Systems

A Real-time Transition Systems (RTS) is a tuple $\langle V, \Sigma, \Theta, \mathcal{T}\ , l, u\rangle$ where

- V is a finite set of variables,
  $V \ni first : boolean$;
- Σ is a set of states,
  i.e. evaluations of V;
- $\Theta \subseteq \Sigma$ are initial states,
  where $first = true$;
- $\mathcal{T}$ is a set of transitions,
  i.e. binary relations on Σ, $idle \in \Sigma$;
- $l, u : \mathcal{T} \rightarrow \mathcal{N} \cup \{\infty\}$
  are minimal and maximal delays,
  $l \leq u$, $l(idle) = 0$, $u(idle) = \infty$.

Initialized Computations

A initialized computation $\rho = (\sigma, T)$ is a pair of infinite sequences
– $\sigma = \sigma_0\sigma_1...$ of states,
– $T = T_0T_1...$ of integers
with $\sigma_0 \in \Theta$, $T_0 = 0$ such that

- (Consecution) $\sigma_0 \xrightarrow{\tau_0} \sigma_1 \xrightarrow{\tau_1} ...$ for some infinite sequence of transitions where a transition τ can occur after $l(\tau)$ clock ticks of inability and must occur up to $u(\tau)$ clock ticks of inability;
- (Progress) $T_0T_1...$ is a non-decreasing sequence with with $sup(T) = \infty$ where the increment (clock tick) 1 can be associated with $idle$ transitions only.

A run is a suffix of an initialized computation: $(\sigma, T) = (\sigma_0, T_0) \; ... \; \underbrace{(\sigma_i, T_i) \; ... \; ...}_{(\sigma,T)^i}$

Multyprocecessing Systems with Shared Variables

A Multyprocecessing Systems with Shared Variables (MSSV) has the form

$$\{\theta\}[P_1\|...\|P_m]$$

where θ is a data precondition for initial states and each process is a sequential non-deterministic program over its privet and shared variables.

A graphical form for process presentation:

- a start label is marked by $\bullet\!\!\rightarrow l_0$,
- a transition is presented by $l_i \xrightarrow[{[l,u]}]{c \rightarrow x := e} l_j$

MSSV generates a RTS:

- variables V are all privet, shared and special control flow (location) variables of all processes;
- in initial states Θ values of privet and shared variables meet data precondition θ while values of location variables are start labels;
- all functions in assignments are total while inability conditions are presented by boolean guards.

State Formulae

We assume a first-order language with interpreted function and predicate symbols for properties of states of a RTS. In particular this language fits to express the following properties:

- arithmetic operations $+,\ -,\ \times,\ /$ and (in)equalities $=,\ \leq,\ \geq,\ <,\ >$,
- control flow locators $at,\ in$;
- a condition $start$ which holds in the initial states Θ,
- Hoar triples $\{p\}\tau\{q\}$ for state formulae and transitions,
- inability conditions $enable(\tau)$ for transitions.

Temporal Formulae

- every state formula p is a temporal formula such that $(\sigma, T) \models p$ iff σ_0 satisfies p;
- every boolean combination of temporal formulae is a temporal formula while its validity is defined in a natural way;
- if p is a state formula, ψ is a temporal formula and $l \in \mathcal{N}$ then $p\ U_{\geq l}\ \psi$ is a temporal formula; $(\sigma, T) \models p\ U_{\geq l}\ \psi$ iff
  - or all states σ satisfies p,
  - or for some $i \geq 0$, $T_i \geq T_0 + l$ and $\underbrace{(\sigma_0, T_0)}_{p} \ldots \underbrace{(\sigma_{i-1}, T_{i-1})}_{p} \underbrace{(\sigma_i, T_i)\ \ldots\ \ldots}_{\models \psi}$;
- if ψ is a temporal formula and $u \in \mathcal{N}$ then $\Diamond_{\geq l}\ \psi$ is a temporal formula; $(\sigma, T) \models \psi$ iff $T_i \leq T_0 + u$ and $(\sigma_0, T_0) \ldots (\sigma_{i-1}, T_{i-1}) \underbrace{(\sigma_i, T_i)\ \ldots\ \ldots}_{\models \psi}$ for some $i \geq 0$.

Example: Bounded Invariance

Let $p \rightarrow \Box_{<l} q$ be $\neg p \vee (p \ U_{\geq l} \ true)$. $(\sigma, T) \models p \rightarrow \Box_{<l} q$ iff

- or p does not hold in (σ_0, T_0),
- or q always holds before l ticks pass.

A typical application of bounded invariance is lower bounds on system termination: if being started, it can not halts before time l, i.e. $start \rightarrow \Box_{<l} \neg(at\ end)$.

Example: Bounded Response

Let $p \to \Diamond_{\leq u} q$ be $\neg p \vee \Diamond_{\leq u} q$. $(\sigma, T) \models p \to \Diamond_{\leq u} q$ iff

- or p does not hold in (σ_0, T_0),
- or q eventually holds
  no later then l ticks pass.

A typical application of bounded response is upper bounds on system termination: if being started, it halts no later then at time l, i.e. $start \to \Diamond_{\leq u}(at\ end)$.

Problem-Oriented Approach:

- classify problems with respect to a syntactical structure of the temporal specification;
- formulate and validate problem-oriented high-level proof principles for abstract RTS;
- design mannually proof-outlines with problem-oriented proof principles;
- check (semi)automatically proof-outlines for RTS generated by a MSSV.

Monotonicity

- **U-MON**

$$\frac{p \to p' \quad \phi \to \phi' \quad l \leq l'}{(p\ U_{\geq l}\ \phi) \to (p'\ U_{\geq l'}\ \phi')}$$

- **$\Diamond$-MON**

$$\frac{\phi \to \phi' \quad u \geq u'}{(\Diamond_{\leq l}\ \phi) \to (\Diamond_{\leq l'}\ \phi')}$$

Single Step

- **U-SS**

$$\frac{\begin{array}{l} p \rightarrow (start \vee \neg enable(\tau)) \\ p \rightarrow s \\ \{s\}\tau'\{s\} \text{ for all } \tau' \not\equiv \tau \\ s \rightarrow q \\ (s \wedge enable(\tau)) \rightarrow r \end{array}}{p \rightarrow (q\ U_{\geq l(\tau)}\ r)}$$

- **◇-SS**

$$\frac{\begin{array}{l} p \rightarrow (s \vee q) \\ s \rightarrow enable(\tau) \\ \{s\}\tau'\{s \vee q\} \text{ for all } \tau' \not\equiv \tau \\ \{s\}\tau\{q\} \end{array}}{p \rightarrow \Diamond_{\leq u(\tau)}\ q}$$

Transitivity

- **U-TRANS**

$$\frac{\begin{array}{c}\phi \rightarrow (p\ U_{\geq l_1} \chi) \\ \chi \rightarrow (q\ U_{\geq l_2} \psi)\end{array}}{\phi \rightarrow ((p \vee q)\ U_{\geq l_1 + l_2}\ \psi)}$$

- **$\Diamond$-TRANS**

$$\frac{\begin{array}{c}\phi \rightarrow \Diamond_{\leq l_1} \chi \\ \chi \rightarrow \Diamond_{\leq l_2} \psi\end{array}}{\phi \rightarrow \Diamond_{\leq l_1 + l_2}\ \psi}$$

Induction

- **U-IND**

$$\frac{(s(i) \wedge i > 0) \rightarrow (p\; U_{\geq l}\; s(i-1))}{s(n) \rightarrow (p\; U_{\geq n \times l}\; s(0))}$$

- **$\Diamond$-IND**

$$\frac{(s(i) \wedge i > 0) \rightarrow \Diamond_{\leq u_i} s(i-1))}{s(n) \rightarrow \Diamond_{\leq u_1 + \ldots u_n} s(0))}$$

where i is a new *static* variable.

Soundness & Completeness

Soundnes:

The above axiomatic system is sound for all temporal formulae and Real-time Transition Systems.

Completeness:

The above axiomatic system is complete for all bounded-invariance and bounded-response formulae with respect to all deterministic Real-time Transition Systems where

- *or all lower bounds are 0,*
- *or all upper bounds are* ∞*.*

Explicit Clock

Let $\langle V, \Sigma, \Theta, \mathcal{T}, l, u\rangle$ be a RTS. Let us introduce a new *explicit clock* variable t for a time, and a new *delay counter* variable δ_τ for every transition τ for counting how long it is enable continuously. An Explicit-clock Transition System (ETS) is a tuple $\langle V^*, \Sigma^*, \Theta^*, \mathcal{T}^*, \mathcal{W}, \mathcal{F}\rangle$ where

- V^* and Σ^* are extended by the clock variable and delay counters with 0 as values in the initial states Θ^*;
- $\mathcal{T}^*$ comprises transitions τ^* for $\tau \in \mathcal{T}$ and a special new transition *tick* for counting time;
- $\mathcal{W}$ is a *weak-fairness* for $\tau \in \mathcal{T}$: if being continuously enable, a corresponding transition τ^* fires evanually;
- $\mathcal{F}$ is a *strong-fairness* for *tick*: if being infinitely often enable, *tick* fires evanually;

Time and Transitions

$\underline{(\sigma_1^*, \sigma_2^*) \in \tau^*}$

- $(\sigma_1, \sigma_2) \in \tau$,
- $\sigma_1^*(\delta_\tau) \geq l_\tau$,
- $\sigma_1^*(t) = \sigma_2^*(t)$,
- $\sigma_2^*(\delta_\tau) = 0$,
- $\sigma_2^*(\delta_{\tau'}) = \sigma_1^*(\delta_{\tau'})$ for all $\tau' \not\equiv \tau$;

$\underline{(\sigma_1^*, \sigma_2^*) \in tick}$

- $\sigma_1 = \sigma_2$,
- $\sigma_2^*(t) = \sigma_1^*(t) + 1$,
- $\sigma_2^*(\delta_\tau) \leq u_\tau$ for all $\tau \in \mathcal{T}$,
- $\sigma_2^*(\delta_\tau) = 0$ iff $\tau \in \mathcal{T}$ is disable in σ_1,
- $\sigma_2^*(\delta_\tau) = \sigma_1^*(\delta_\tau) + 1$ iff $\tau \in \mathcal{T}$ is enable in σ_1;

Reduction to Time-Free Reasonning

Let T be a new *static* integer variable. Let U be time-free *unless* operator (i.e. $U_{\geq 0}$). Then

- $(p \land t = T) \rightarrow (q\ U\ (t \geq T + l))$ expresses a bounded invariance property $p \rightarrow \Box_{<l} q$;
- $(p \land t = T) \rightarrow ((t \leq T + u)\ U\ q)$ expresses a bounded response property $p \rightarrow \Diamond_{\leq u} q$.

General Case:

lower bounds may affect upper bound

$$\theta : x = 1 \wedge y = 0$$

$$\begin{array}{rcrcl}
 & \bullet & & \bullet & \\
 & \downarrow & & \downarrow & \\
 & \downarrow & & \downarrow & \\
 & l_0^1 & & l_0^2 & \overset{x \neq 0?/[0,0]}{\underset{y := y+1/[1,10]}{\rightleftarrows\rightleftarrows\rightleftarrows\rightleftarrows}} \; l' \\
x := 0 & \downarrow & x = 0? & \downarrow & \\
[1,10] & \downarrow & [0,0] & \downarrow & \\
 & l_1^1 & & l_1^2 & \overset{y \neq 0?/[0,0]}{\underset{y := y-1/[1,10]}{\rightleftarrows\rightleftarrows\rightleftarrows\rightleftarrows}} \; l'' \\
 & & y = 0? & \downarrow & \\
 & & [0,0] & \downarrow & \\
 & & & l_2^2 &
\end{array}$$

P_2 consumes $130 ticks$ iff it is executes the first loop 11 times. But if the lower bounds of all assignments are $2 ticks$ then the first loop can be executed 5 times at most, and in this case P_2 consumes $80 ticks$ only!