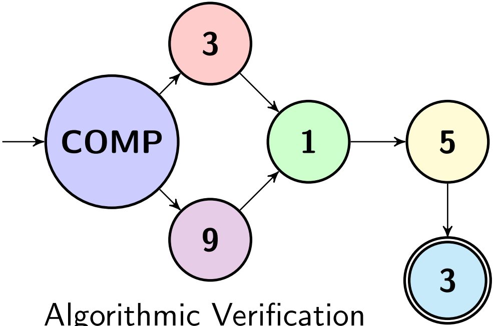

Algorithmic Verification

Timed CTL and TCTL$_C$

Dr. Liam O'Connor
CSE, UNSW (and LFCS, University of Edinburgh)
Term 1 2020

Timed Logic

Timed CTL

TCTL is CTL with clock constraints (as in TA) attached to **U** (and derived operators).

Note: The next-state operator **X** has no meaning in dense time.

Syntax

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{E}\ \varphi\ \mathbf{U}_{\sim c}\ \varphi \mid \mathbf{A}\ \varphi\ \mathbf{U}_{\sim c}\ \varphi$$

Where $p \in \mathcal{P}$ is an atomic proposition and $(\sim) \in \{<, \leq, =, \geq, >\}$.

TCTL Semantics

Semantics are defined on a *timed transition system*.

Timed Transition Systems

A TTS is a timed automaton with a labelling function L associating sets of atomic propositions to states (analogous to Kripke structures).

Our modelling relation is defined on a *configuration* (state). Let $\mathrm{Exec}(s)$ be the set of executions from configuration s, and $\mathrm{Dur}(p|_{\leq k})$ be the sum of delays along the prefix $\rho|_{\leq k}$ of the execution ρ.

$$
\begin{aligned}
s \models p \quad & \Leftrightarrow \quad p \in L(s) \\
s \models \mathbf{A}\ \varphi\ \mathbf{U}_{\sim k}\ \psi \quad & \Leftrightarrow \quad \forall \rho \in \mathrm{Exec}(s).\ \rho \models \varphi\ \mathbf{U}_{\sim k}\ \psi \\
s \models \mathbf{E}\ \varphi\ \mathbf{U}_{\sim k}\ \psi \quad & \Leftrightarrow \quad \exists \rho \in \mathrm{Exec}(s).\ \rho \models \varphi\ \mathbf{U}_{\sim k}\ \psi
\end{aligned}
$$

$$
\rho \models \varphi\ \mathbf{U}_{\sim k}\ \psi \quad \Leftrightarrow \quad \exists i.\ \mathrm{Dur}(\rho|_{\leq i}) \sim k \wedge \rho_i \models \psi \wedge \forall j < i.\ \rho_j \models \varphi
$$

Derived Operators

- Standard $\mathbf{U}$ is just $\mathbf{U}_{\geq 0}$.
- Path formulae $\mathbf{F}_{\sim k}$ and $\mathbf{G}_{\sim k}$ are similar to normal CTL:

$$\begin{aligned} \mathbf{F}_{\sim k}\ \varphi &\equiv \text{True}\ \mathbf{U}_{\sim k}\ \varphi \\ \mathbf{G}_{\sim k}\ \varphi &\equiv \neg \mathbf{F}_{\sim k}\ \neg\varphi \end{aligned}$$

Definition

A timed automaton $\mathcal{A}$ satisfies a formula φ, written $\mathcal{A} \models \varphi$ iff its initial configuration $(q_0, \overline{0})$ satisfies φ i.e. $(q_0, \overline{0}) \models \varphi$.

Example

The alarm is activated at most 10 time units after a problem occurs.

$$\mathbf{AG}(\texttt{problem} \Rightarrow \mathbf{AF}_{\leq 10}\ \texttt{alarm})$$

Converting to Automata

Let's try to construct a timed (Büchi) automaton that accepts all timed words that satisfy this property:

$$\mathbf{AG}(\texttt{problem} \Rightarrow \mathbf{AF}_{\leq 10}\ \texttt{alarm})$$

How do we know where to introduce clocks?

TCTL$_C$

TCTL is CTL with explicit clock constraints and reset.

Syntax

$$\varphi ::= x \sim k \mid x.\varphi \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{E}\ \varphi\ \mathbf{U}\ \varphi \mid \mathbf{A}\ \varphi\ \mathbf{U}\ \varphi$$

Where $x \in X$ is a clock variable and $(\sim) \in \{<, \leq, =, \geq, >\}$.

$x.\varphi$ is a clock reset.

Example (Alarm)

How do we express:

$$\mathbf{AG}(\mathtt{problem} \Rightarrow \mathbf{AF}_{\leq 10}\ \mathtt{alarm})$$

in TCTL$_C$?

Expressivity

Result

All TCTL formulae are expressive in TCTL by introducing a fresh clock for each constrained operator:

$$\mathbf{E}\ \varphi\ \mathbf{U}_{\sim k}\ \psi \quad \equiv \quad (x.\ \mathbf{E}\ \varphi\ \mathbf{U}\ (\psi \wedge x \sim k))$$

The converse direction does not hold (Bouyer et al. 2005):

$$x.\ \mathbf{EF}(\varphi \wedge x < 1 \wedge \mathbf{EG}(x < 1 \Rightarrow \neg\psi))$$

cannot be expressed in TCTL.

Model Checking

Same techniques as reachability:

1. Convert timed system A to discrete systems A' via region automata.
2. Convert TCTL$_C$ formula φ to standard CTL formula φ' on region automata.
3. $A \models \varphi \iff A' \models \varphi'$, so apply standard CTL model checking.
4. Checking is still PSPACE complete.

UPPAAL

A mature model checking framework for timed transition systems.

B. Srivathsan has released a video lecture on using UPPAAL on several examples here:

`https://www.youtube.com/watch?v=tUSxi_rSXwo`