Model Checking Timed Automata
Material from “Principles of Model Checking” by C. Baier and J.-P Katoen

Borzoo Bonakdarpour

School of Computer Science
University of Waterloo

November 24, 2013
Outline

1 Timed Computation Tree Logic (TCTL)

2 TCTL Model Checking
Presentation outline

1. Timed Computation Tree Logic (TCTL)

2. TCTL Model Checking
Definition (Syntax of TCTL)

Formulae in TCTL are either state or path formulae. TCTL state formulae over the set $AP$ of atomic propositions and set $C$ of clocks are formed according to the following grammar:

$$\Phi ::= true \mid a \mid g \mid \Phi \land \Phi \mid E\varphi \mid A\varphi$$

where $a \in AP$, $g \in ACC(C)$ and $\varphi$ is a path formula defined by:

$$\varphi ::= \Phi U^J \Phi$$

where $J \subseteq \mathbb{R}_{\geq 0}$ is an interval whose bounds are natural numbers.
Timed Computation Tree Logic (TCTL)

**TCTL Temporal Abbreviations**

\[ \Diamond^J \Phi = true \mathbf{U}^J \Phi \]
\[ E \Box^J \Phi = \neg A \Diamond^J \neg \Phi \]
\[ A \Box^J \Phi = \neg E \Diamond^J \neg \Phi \]
Timed Computation Tree Logic (TCTL)

TCTL Temporal Abbreviations

\[ \diamond_j \Phi = \text{true} \land_j \Phi \]
\[ E \square_j \Phi = \neg A \diamond_j \neg \Phi \]
\[ A \square_j \Phi = \neg E \diamond_j \neg \Phi \]

TCTL Interval Abbreviations

Intervals are often denoted by shorthand, e.g., \( \diamond \leq 2 \) denotes \( \diamond [0, 2] \) and \( \square > 8 \) denotes \( \square (8, \infty) \)
Consider the following timed automata

The property: "the light cannot be continuously switched on for more than 2 minutes" is expressed by the TCTL formula:

$$\text{A} \square (\text{on} \rightarrow \text{A} \diamond > 2 \neg \text{on})$$
The property: 

"the light cannot be continuously switched on for more than 2 minutes"

is expressed by the TCTL formula:

$\mathbf{A} \Box (on \rightarrow \mathbf{A} \Diamond >^2 \neg on)$
**Semantics of TCTL**

**Definition (Satisfaction relation for TCTL)**

Let $TA = (L, \Sigma, E, C, L_0, I)$ be a timed automaton, $a \in AP$, $g \in ACC(C)$, and $J \subseteq \mathbb{R}_{\geq 0}$. For state $s = \langle l, \nu \rangle$ in $TS(TA)$ and TCTL formulae $\Phi$ and $\Psi$, and TCTL path formula $\varphi$, the satisfaction relation $\models$ is defined for state formulae by

<table>
<thead>
<tr>
<th>$s \models \Phi$</th>
<th>iff</th>
<th>$s \models \varphi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$true$</td>
<td></td>
<td>$a \in Label(l)$</td>
</tr>
<tr>
<td>$a$</td>
<td></td>
<td>$\nu \models g$</td>
</tr>
<tr>
<td>$\neg \Phi$</td>
<td></td>
<td>not $s \models \Phi$</td>
</tr>
<tr>
<td>$\Phi \land \Psi$</td>
<td></td>
<td>$(s \models \Phi) \land (s \models \Psi)$</td>
</tr>
<tr>
<td>$E \varphi$</td>
<td></td>
<td>$\pi \models \varphi$ for some $\pi \in Paths_{div}(s)$</td>
</tr>
<tr>
<td>$A \varphi$</td>
<td></td>
<td>$\pi \models \varphi$ for all $\pi \in Paths_{div}(s)$</td>
</tr>
</tbody>
</table>
Definition (Satisfaction relation for TCTL (con’d))

For a time-divergent path \( \pi = s_0 \xrightarrow{d_0} s_1 \xrightarrow{d_1} \ldots \), the satisfaction relation \( \models \) for path formulae is defined by:

\[
\pi \models \Phi U^J \Psi \iff \exists i \geq 0. s_i + d \models \Psi \text{ for some } d \in [0, d_i] \text{ with }
\]

\[
i-1 \sum_{k=0}^{i-1} d_k + d \in J \quad \text{and}
\]

\[
\forall j \leq i. s_j + d' \models \Phi \lor \Psi \text{ for any } d' \in [0, d_j] \text{ with }
\]

\[
j-1 \sum_{k=0}^{j-1} d_k + d' \leq i-1 \sum_{k=0}^{i-1} d_k + d
\]

where for \( s_i = \langle l_i, \nu_i \rangle \) and \( d \geq 0 \), we have \( s_i + d = \langle l_i, \nu_i + d \rangle \)
Semantics of TCTL (cont'd)

Definition (TCTL Semantics for Timed Automata)

A timed automaton $TA$ satisfies a TCTL formula $\Phi$ iff $s_0 \models \Phi$ for each initial state $s_0$ of $TA$. 
1. Timed Computation Tree Logic (TCTL)

2. TCTL Model Checking
Reduction to CTL Model Checking

**Idea**

Given a time automaton $TA$ and a TCTL formula $\Phi$, our goal is to find a finite transition system $S$ and an CTL formula $\hat{\Phi}$, such that

$$TA \models_{TCTL} \Phi \iff R(TA, \Phi) \models_{CTL} \hat{\Phi}$$
**Input:** timed automaton $TA$ and TCTL formula $\Phi$ (both over propositions $AP$ and clocks $C$).

**Output:** $TA \models \Phi$

1. $\hat{\Phi} :=$ eliminate the timing parameters from $\Phi$;
2. determine the clock equivalence classes under $\sim$;
3. construct the region transition system $TS = R(TA, \Phi)$;
4. apply the CTL model checking algorithm to check $TS \models \hat{\Phi}$
5. $TA \models \Phi$ if and only if $TS \models \hat{\Phi}$

**Algorithm 1:** A recipe for TCTL model checking
Elimination of Timing Parameters

Notation

For clock evaluation $\nu$, $z \notin C$, and $d \in \mathbb{R}_{\geq 0}$, let $\nu\{z := d\}$ denote the clock valuation for $C \cup \{z\}$ that extends $\nu$ by setting $z$ to $d$ while keeping the value of all other clocks unchanged:

$$\nu\{z := d\}(x) = \begin{cases} \nu(x) & \text{if } x \in C \\ d & \text{if } x = z \end{cases}$$

(1)
Elimination of Timing Parameters

Notation

For clock evaluation $\nu$, $z \notin C$, and $d \in \mathbb{R}_{\geq 0}$, let $\nu\{z := d\}$ denote the clock valuation for $C \cup \{z\}$ that extends $\nu$ by setting $z$ to $d$ while keeping the value of all other clocks unchanged:

$$\nu\{z := d\}(x) = \begin{cases} 
\nu(x) & \text{if } x \in C \\
 d & \text{if } x = z 
\end{cases}$$  \hfill (1)

Notation

Let $TA$ be a timed automaton over clocks $C$. For state $s = \langle l, \nu \rangle$ in $TS(TA)$ let $s\{z := d\}$ denote the state, $\nu\{z := d\}$. Note that $s\{z := d\}$ is a state in $TS(TA \oplus z)$ where $TA \oplus z$ is the timed automaton $TA$ with the set of clocks $C \cup \{z\}$. 
Elimination of Timing Parameters

**Theorem**

Let $TA$ be timed automaton $(L, \Sigma, C, E, L^0, I)$, and $\Phi U^J \Psi$ a TCTL formula over $C$ and $AP$. For clock $z \not\in C$ and for any state $s$ of $TS(TA)$ it holds that

1. $s \models_{TCTL} E(\Phi U^J \Psi)$ iff $s\{z := 0\} \models_{CTL} E((\Phi \lor \Psi) U ((z \in J) \land \Psi))$.

2. $s \models_{TCTL} A(\Phi U^J \Psi)$ iff $s\{z := 0\} \models_{CTL} A((\Phi \lor \Psi) U ((z \in J) \land \Psi))$. 
Example

Light Switch Consider the following timed automaton $TA$ and the TCTL formula $\Phi = E\Diamond \leq 1$ on.

As a first step, $\Phi$ is replaced by $\hat{\Phi} = E\Diamond (z \leq 1 \land \text{on})$ and $TA$ is equipped with an additional clock $z$. The maximal constants for the clocks $x$ and $z$ are $c_x = 1$ and $c_z = 1$. The region transition system $TS = R(TA \oplus z, \Phi)$ is on the next slide.
Example (con’d)

B. Bonakdarpour (School of Computer Science, University of Waterloo)
Model Checking Timed Automata
November 24, 2013 16 / 18

Example

Light Switch (cont’d)

\[
\begin{align*}
\text{off} & \quad x = 0 \quad z = 0 \\
\text{off} & \quad 0 < x < 1 \quad 0 < z < 1 \quad x = z \\
\text{off} & \quad x = 1 \quad z = 1 \\
\text{off} & \quad x > 1 \quad z > 1 \\
\text{on} & \quad x = 0 \quad z = 1 \\
\text{on} & \quad 0 < x < 1 \quad z > 1 \\
\text{on} & \quad x = 1 \quad z > 1 \\
\text{off} & \quad x = 1 \quad z > 1 \\
\text{on} & \quad x = 0 \quad z > 1 \\
\end{align*}
\]
Example (con’d)

Example

Light Switch (cont’d) The state region

\[ \langle on, [x = 0, z = 1] \rangle \models (z \leq 1) \land on \]

and is reachable from the initial state region. Therefore,

\[ TS \models_{CTL} E\Box((z \leq 1) \land on) \]

and thus

\[ TA| = E\Box^{\leq 1} on \]
Eliminating Multiple Clocks

A simple way of treating formulae with nested time bounds is to introduce a fresh clock for each subformula.

Example

For example, the following TCTL formula

$$\Phi = A\Box^{\geq 3} E\Diamond [1,2] on$$

is transformed into:

$$\hat{\Phi} = A\Box((z_1 \geq 3) \Rightarrow E\Diamond(z_2 \in ]1, 2]) \land on))$$