OUTLINE

• Model Checking in a Nutshell
• Timed automata and TCTL
• A UPPAAL Tutorial
  • Data structures & central algorithms
  • UPPAAL input languages

Timed Automata, TCTL & Verification Problems

Timed Automata: Syntax

Timed Automata: Semantics

Timed Automata with Invariants

Timed Automata: Example
Timed Automata: Example

Clock Constraints

\[ g ::= x \sim n \mid g \land g \]

where
- \( x \) is a clock variable
- \( \sim \in \{<,>,\leq,\geq\} \)
- \( n \) is a natural number

Semantics (definition)

- **clock valuations**: \( V(C) \rightarrow R_{\geq 0} \)
- **state**: \( (l, v) \) where \( l \in L \) and \( v \in V(C) \)
- **action transition**: \( (l, v) \xrightarrow{a} (l', v') \) iff \( g(v) \) and \( v' = v[r] \) and \( Inv(l')(v') \)
- **delay transition**: \( (l, v) \xrightarrow{\delta} (l, v + d) \) iff \( Inv(l)(v + d') \) whenever \( d \leq d' \in R_{\geq 0} \)

Timed Automata

\[ \overset{\text{=}}{\text{=}} \]

Finite Automata + Clock Constraints + Clock resets
Modeling Concurrency

- Products of automata
- CCS Parallel composition
  - implemented in UPPAAL

CCS Parallel Composition (implemented in UPPAAL)

where \( a \) is an action \( c! \) or \( c? \) or \( \tau \), and \( c \) is a channel name

The UPPAAL Model

= Networks of Timed Automata + Integer Variables + ...

Example transitions:

\[
\begin{align*}
(l_0, m_0, \ldots, x=2, y=3.5, i=3, \ldots) & \rightarrow (l_0, m_0, \ldots, x=2, y=3.5, i=7, \ldots)
\end{align*}
\]

Verication Problems

Location Reachability (def.)

\( n \) is reachable from \( m \) if there is a sequence of transitions:

\[
\begin{align*}
(m, u) & \rightarrow^{\ast} (n, v)
\end{align*}
\]

(Timed) Language Inclusion, \( L(A) \subseteq L(B) \)

\[
\begin{align*}
(a_0, t_0)(a_1, t_1) \ldots (a_n, t_n) \in L(A)
\end{align*}
\]

If

- \( \Lambda \) can perform \( a_0 \) at \( t_0 \), \( a_1 \) at \( t_1 \) ... \( a_n \) at \( t_n \)
- \( \langle l_0, u_0 \rangle \) \( \rightarrow \langle l_0, u_0 + t_2 \rangle \) \( \rightarrow \langle l_0, u_0 \rangle \) \( \ldots \)
Verification Problems

- Timed Language Equivalence & Inclusion
  - 1-clock, finite traces, decidable [Ouaknine & Worrell 04]
  - 1-clock, infinite traces & Buchi-conditions, undecidable [Abdulla et al 05]
- Universality
- Untimed Language Inclusion
- (Un)Timed (B)simulation
- Reachability Analysis/Emptiness
- Optimal Reachability (synthesis problem)
  - If a location is reachable, what is the minimal delay before reaching the location?

Timed CTL = CTL + clock constraints

Note that the semantics of TA defines a transition system where each state has a Computation Tree

Computation Tree Logic, CTL
Clarke & Emerson 1980

Syntax

\[ \phi ::= P \mid \neg \phi \mid \phi \lor \psi \mid \text{EX} \mid \text{E}[\phi U \psi] \mid \text{A}[\phi U \psi] \]

where \( P \in \text{AP (atomic propositions)}\)

Derived Operators

- \( \text{AG} \)
- \( \text{EG} \)
- \( \text{EF} \)
- \( \text{AF} \)

Liveness: \( p - \rightarrow q \) “\( p \) leads to \( q \)”

Timed CTL (a simplified version)

Syntax

\[ \phi ::= P \mid \neg \phi \mid \phi \lor \psi \mid \text{EX} \mid \text{E}[\phi U \psi] \mid \text{A}[\phi U \psi] \]

where \( P \in \text{AP (atomic propositions)} \) OR Clock constraint

Derived Operators

- \( \text{AG} \)
- \( \text{EG} \)
- \( \text{EF} \)
- \( \text{AF} \)

Timed CTL (a simplified version)

Syntax

\[ \phi ::= P \mid \neg \phi \mid \phi \lor \psi \mid \text{EX} \mid \text{E}[\phi U \psi] \mid \text{A}[\phi U \psi] \]

where \( P \in \text{AP (atomic propositions)} \) OR Clock constraint

Derived Operators

- \( \text{AG} \)
- \( \text{EG} \)
- \( \text{EF} \)
- \( \text{AF} \)
Derived Operators (cont.)

\[ AG(p \implies AF q) \]

\[ p \implies q \text{ in UPPAAL} \]

Bounded Liveness

Verify: "whenever p is true, q should be true within 10 sec

\[ p \implies (q \text{ and } x<10) \]

Use extra clock x
Add \( x:=0 \) on all edges leading to P

Bounded Liveness/Responsiveness

(reachability analysis, more efficient?)

Verify: "whenever p is true, q should be true within 10 sec

\[ AG((P_b \text{ and } x>10) \implies q) \]

Use extra clock x and boolean \( P_b \)
Add \( P_b:=tt \) and \( x:=0 \) on all edges leading to location P

Problem with Zenoness/Time-stop

EXAMPLE

We want to specify "whenever P is true, Q should be true within 10 time units

\[ y\leq5 \]

\[ (P_y:=5) \]

\[ y\leq5 \]
We want to specify "whenever P is true, Q should be true within 10 time units"

\[ p := \text{true} \quad x := 0 \]

\[ \text{AG } (\text{P} \land x > 10) \Rightarrow Q \]

We want to specify "whenever P is true, Q should be true within 10 time units"

\[ p := \text{true} \quad x := 0 \]

\[ \text{AG } (\text{P} \land x > 10) \Rightarrow q \]

is satisfied !!!

Solution with UPPAAL

Check Zeno-freeness by an extra observer

System || ZenoCheck

A

X=1

Check (yes means "no zeno loops")

ZenoCheck.A -> ZenoCheck.B

Committed location!

Infinite State Space!

Region: From infinite to finite

Concrete State

\((n, x=2.2, y=1.5)\)

Symbolic state (region)

\((n, \ldots)\)

An equivalence class (i.e., a region)

There are only finite many such!!
Region equivalence (Intuition)

\[ u \equiv v \iff (l, u) \text{ and } (l, v) \text{ may reach the same set of equivalence classes} \]

Region equivalence (alternatively)

\[ u \equiv v \iff u \text{ and } v \text{ satisfy exactly the same set of constraints in the form of } x_i \sim m \text{ and } x_i \sim n \text{ where } \sim \text{ is in } \{<, >, \leq, \geq\} \text{ and } m, n < \text{MAX} \]

This is not quite correct; we need to consider the MAX more carefully.

Region equivalence \([\text{Alur and Dill 1990}]\)

- \(u, v\) are clock assignments
- \(u \equiv v\) iff
  - For all clocks \(x\),
    - either (1) \(u(x) > C_x\) and \(v(x) > C_x\)
    - or (2) \(\lfloor u(x) \rfloor = \lfloor v(x) \rfloor\)
  - For all clocks \(x\), if \(u(x) \leq C_x\),
    - \(\{u(x)\} = 0\) \(\iff\) \(\{v(x)\} = 0\)
  - For all clocks \(x, y\), if \(u(x) \leq C_x\) and \(u(y) \leq C_y\)
    - \(\{u(x)\} \leq \{u(y)\} \iff \{v(x)\} \leq \{v(y)\}\)

Region Graph

Finite-State Transition System!!

OBS: there are only Finite many regions

\[ (m, [u]) \rightarrow (n, [v]) \text{ if } (m, u) \rightarrow (n, v) \]
Theorem

\[ u \equiv v \implies \]
\[ u(x:=0) \equiv v(x:=0) \]
\[ u+n \equiv v+n \text{ for all natural number } n \]
\[ \text{for all } d<1: u+d \equiv v+d' \text{ for some } d'<1 \]

"Region equivalence" is preserved by "addition" and reset.
(Also preserved by "subtraction" if clock values are "bounded")

Region graph of a simple timed automata

Fischers again

Problems with Region Construction

- Too many 'regions'
  - Sensitive to the maximal constants
  - e.g. \( x > 1,000,000, y > 1,000,000 \) as guards in TA
- The number of regions is highly exponential in the number of clocks and the maximal constants.

Zones: From infinite to finite

REACHABILITY ANALYSIS using ZONES
Symbolic Transitions

Fischer's Protocol analysis using zones

Thus \((n, 1 \leq x \leq 4, 1 \leq y \leq 3) \Rightarrow (m, 3 < x, y = 0)\)

Fischer's Protocol cont.

Taking time into account

Fischer's Protocol cont.

Taking time into account

Fischer's Protocol cont.

Taking time into account

Fischer's Protocol cont.

Taking time into account

Fischer's Protocol cont.
Fischers cont.

Untimed case

Taking time into account

Zones = Conjunctive constraints

- A zone $Z$ is a conjunctive formula: $g_1 \land g_2 \land \ldots \land g_n$
  where $g_i$ may be $x_i \sim b_i$ or $x_i-x_j \sim b_{ij}$
- Use a zero-clock $x_0$ (constant 0), we have
  $(x_i-x_j \sim b_{ij} \mid i,j \leq n)$
- This can be represented as a MATRIX, DBM
  (Difference Bound Matrices)

Solution set as semantics

- Let $Z$ be a zone (a set of constraints)
- Let $[Z]=\{u \mid u$ is a solution of $Z\}$
  (We shall simply write $Z$ instead $[Z]$)

Operations on Zones

- Post-condition (Delay): $SP(Z)$ or $Z^\uparrow$
  $\{Z^\uparrow\} = \{u+d \mid d \in \mathbb{R}, u \in [Z]\}$
- Pre-condition: $WP(Z)$ or $Z^\downarrow$ (the dual of $Z^\uparrow$)
  $\{Z^\downarrow\} = \{u \mid u+d \in [Z]$ for some $d \in \mathbb{R}\}$
- Reset: $\{x\}Z$ or $Z(x:=0)$
  $\{x\}Z = \{u[0/x] \mid u \in [Z]\}$
- Conjunction
  $\{Z_1 \land Z_2\} = \{Z_1 \land Z_2\}$

Two more operations on Zones

- Inclusion checking: $Z_1 \subseteq Z_2$
  solution sets
- Emptiness checking: $Z = \emptyset$
  no solution
Theorem on Zones

The set of zones is closed under all zone operations

- That is, the result of the operations on a zone is a zone
- Thus, there will be a zone to represent the sets: $[Z^\uparrow]$, $[Z^\downarrow]$, $[(x)Z]

One-step reachability: $S_i \rightarrow S_j$

- **Delay:** $(n, Z) \rightarrow (n, Z')$ where $Z' = Z^\uparrow \land \text{inv}(n)$
- **Action:** $(n, Z) \rightarrow (m, Z')$ where $Z' = (x)(Z \land g)$
  
  ![Diagram of one-step reachability](image)

- **Reach:** $(n, Z) \sim (m, Z')$ if $(n, Z) \rightarrow (m, Z')$
- **Successors** $(n, Z) = \{(m, Z') | (n, Z) \sim (m, Z'), Z' \neq \emptyset\}$

Now, we have a search problem

![Diagram of a search problem](image)