UNIVERSITY OF TWENTE.

Symbolic Model Checking of Timed Automata using LTSmin

Sybe van Hijum

Overview

Timed Automata

Symbolic Model Checking

LTSmin

Working with LaTeX

Second Section

Third Section

Transition System

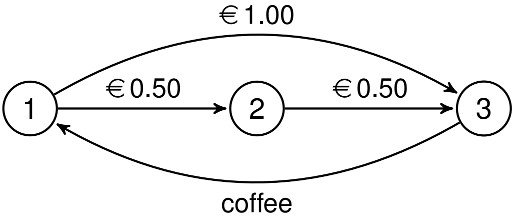
Definition (Labeled Transition System)

A labeled transition system is a 3-tuple $A = \langle S, Act, s_o \rangle$ *where*

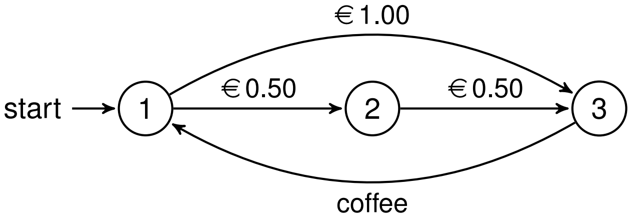
- S *is a finite set of states*
- Act *is a finite set of labelled actions*
- $s_o \in S$ *is a finite set of actions*

Transition System

Transition System



Transition System



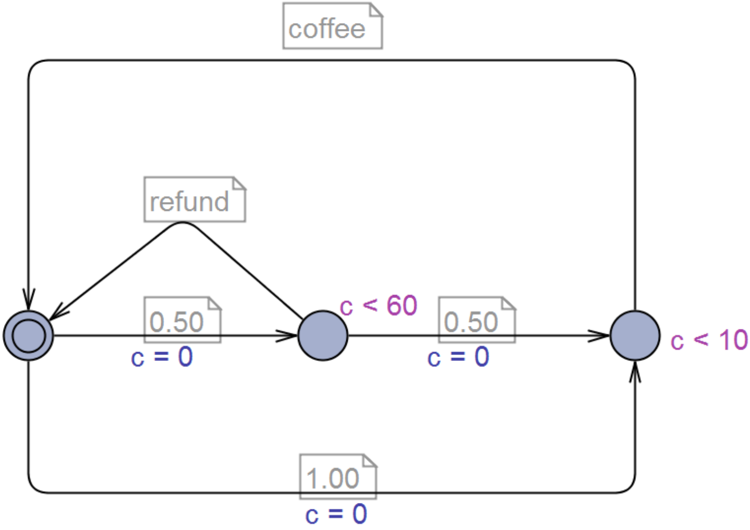
Timed Automata

Definition (Timed Automata)

An extended timed automaton is a 6-tuple $A = \langle L, C, Act, l_0, \rightarrow, I_c \rangle$ *where*

- L *is a finite set of locations, typically denoted by* l
- C *is a finite set of clocks, typically denoted by* c
- Act *is a finite set of actions*
- $l_0 \in L$ *is the initial location*
- $\rightarrow \subseteq L \times G(C) \times Act \times 2^C \times L$ *is the (non-deterministic) transition relation.*
- $I_C : L \rightarrow G(C)$ *is a function mapping locations to downwards closed clock invariants.*

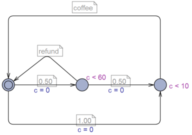
Timed Automata



Time Zones

Time not represented as a variable, but as a zone. Most used structure to represent zones: Different Bound Matrix (DBM)

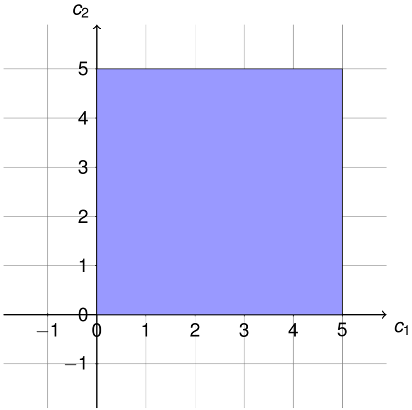
- Only convex zones
- Memory inefficient



$$0 \le c < 60$$
$$\Downarrow$$
$$c - 0 < 60$$
$$0 - c \le 0$$

$$\begin{array}{ccc} & \mathbf{0} & c \\ \mathbf{0} & \multicolumn{2}{c}{\multirow{2}{*}{$\begin{pmatrix} (0, \le) & (0, \le) \\ (60, <) & (0, \le) \end{pmatrix}$}} \\ c & & \end{array}$$

$$
\begin{array}{c|ccc}
 & \mathbf{0} & c_1 & c_2 \\
\hline
\mathbf{0} & & & \\
c_1 & & & \\
c_2 & & &
\end{array}
\qquad
\begin{array}{cc}
 & \begin{array}{ccc} \mathbf{0} & c_1 & c_2 \end{array} \\
\begin{array}{c} \mathbf{0} \\ c_1 \\ c_2 \end{array} &
\begin{pmatrix}
(0, \leq) & (0, \leq) & (0, \leq) \\
(5, <) & (0, \leq) & (5, <) \\
(5, <) & (5, <) & (0, \leq)
\end{pmatrix}
\end{array}
$$



Overview

Timed Automata

Symbolic Model Checking

LTSmin

Working with LaTeX

Second Section

Third Section

Boolean Decision Diagram

- Expresses boolean expressions
- States can be seen as boolean expressions
- Memory efficient

Boolean Decision Diagram

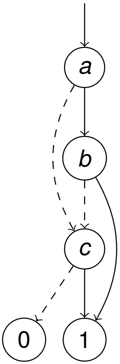
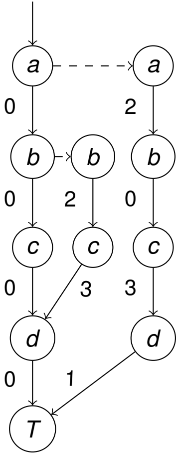


Figure: A BDD representing $(a \wedge b) \vee c$

List Decision Diagram



Overview

Timed Automata

Symbolic Model Checking

LTSmin

Working with LaTeX

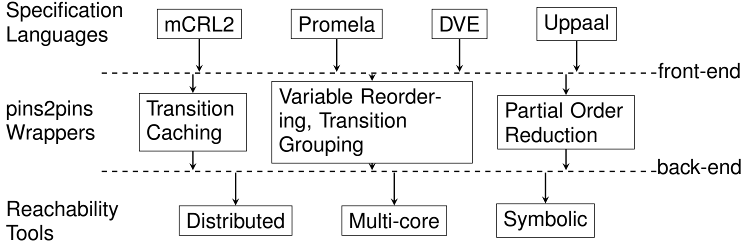
Second Section

Third Section

LTSmin

- Language independent model checker
- Multiple algorithmic back ends
- Internal optimization wrappers

LTSmin


Specification Languages
mCRL2
Promela
DVE
Uppaal
front-end
pins2pins Wrappers
Transition Caching
Variable Reordering, Transition Grouping
Partial Order Reduction
back-end
Reachability Tools
Distributed
Multi-core
Symbolic

LTSmin

- States as integer vectors
- Partitioned next-state function
- Optimizations based on matrices
  - Read(r)
  - Must-write(w)
  - May-write(W)
  - Copy(-)

Matrices

$$\begin{array}{lll} 1: & x = 1 \vee a[1] = 0 & \rightarrow a[1] := 1, x := 0, y := 5 \\ 2: & a[0] = 1 \vee y = 5 & \rightarrow a[x] := 0, x := 1 \end{array}$$

$$\begin{array}{c} \\ 1 \\ 2 \end{array} \begin{array}{c} \begin{array}{cccc} x & y & a[0] & a[1] \end{array} \\ \begin{bmatrix} + & w & - & + \\ + & r & + & W \end{bmatrix} \end{array}$$

Problem: Model checkers are designed for discrete variables (integers), clocks have real values.

- Can we use the LTSmin symbolic model checker for timed automata?
- Can we optimize the symbolic back end for clocks?

Current LTSmin Uppaal setup

States as a vector of discrete locations and a pointer to a DBM.
Implemented in explicit-state multi-core tool.
First approach: values from DBM directly into an LDD

$$
\begin{array}{c c}
 & \begin{array}{ccc} \mathbf{0} \quad\quad & c_1 \quad\quad & c_2 \end{array} \\
\begin{array}{c} \mathbf{0} \\ c_1 \\ c_2 \end{array} &
\begin{pmatrix}
(0, \leq) & (0, \leq) & (0, \leq) \\
(5, <) & (0, \leq) & (5, <) \\
(5, <) & (5, <) & (0, \leq)
\end{pmatrix}
\end{array}
$$

Old situation: $\{l_0, ..., l_n, ptr\}$
New situation:
$\{l_0, ..., l_n, (0, \leq), (0, \leq), (5, <), (5, <), (5, <), (5, <)\}$

LDD solution

- Correct, working solution
- Variable reordering possible
- All variables seen as discrete values
- No optimizations based on time

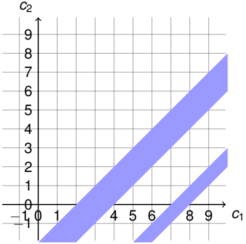
Difference Decision Diagram

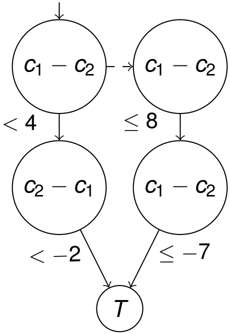
Definition (Difference Decision Diagram)

A difference decision diagram (DDD) is a directed acyclic graph (V, E). The vertex set V contains two terminals 0 and 1 with out-degree zero, and a set of non-terminal vertices with out-degree two and the following attributes.

Attribute	*Type*	*Description*
pos(v), neg(v)	***Var***	*Positive variable x_i, and negative variable x_j.*
op(v)	$\{<, \leq\}$	*Operator $<$ or $\leq$.*
const(v)	$\mathbb{D}$	*Constant c.*
high(v), low(v)	*V*	*High-branch h, and low-branch l.*

The set E contains the edges $(v, low(v))$ and $(v, high(v))$, where $v \in V$ is a non-terminal vertex.


c_2
c_1


$c_1 - c_2$
$c_1 - c_2$
< 4
≤ 8
$c_2 - c_1$
$c_1 - c_2$
< -2
≤ -7
T

Definition (Ordered DDD)

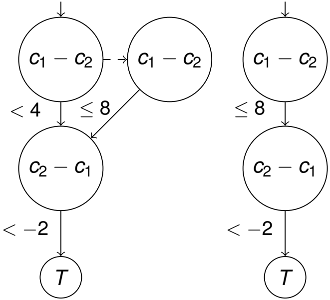
An ordered DDD (ODDD) is a DDD where each non-terminal vertex v satisfies:

1. $neg(v) \prec pos(v)$,
2. $var(v) \prec var(high(v))$,
3. $var(v) \prec var(low(v))$ *or*
   $var(v) = var(low(v))$ *and* $bound(v) \prec bound(low(v))$.

Definition (Locally Reduced DDD)

A locally reduced DDD ($R_L DDD$) is an ODDD satisfying, for all non-terminals u and v:

1. $\mathbb{D} = \mathbb{Z}$ *implies* $\forall v.op(v) = '\leq'$,
2. $(cstr(u), high(u), low(u)) = (cstr(v), high(v), low(v))$ *implies* $u = v$,
3. $low(v) \neq high(v)$,
4. $var(v) = var(low(v))$ *implies* $high(v) \neq high(low(v))$.



Overview

Timed Automata

Symbolic Model Checking

LTSmin

Working with LaTeX

Second Section

Third Section

Font Sizes

Table: The different font sizes within LaTeX

tiny	sample text
scriptsize	sample text
footnotesize	sample text
small	sample text
normalsize	sample text
large	sample text
Large	sample text
LARGE	sample text
huge	sample text
Huge	sample text

Creation of a new frame

The text within the frame

Creation of a new frame - `source`

```
\begin{frame}{Creation of a new frame}
    The text within the frame
\end{frame}
```

Frame with `pause` itemes

- First item

Frame with `pause` itemes

- First item
- Second item

Frame with `pause` itemes

- First item
- Second item
- You get the point.

Frame with `pause` itemes - `source`

```
\begin{frame}{Frame with \texttt{pause} itemes}
\begin{itemize}
\item First item \pause
\item Second item \pause
\item You get the point.
\end{itemize}
\end{frame}
```

Frame with `pause` tables

Table: Caption

Class	A	B	C	D
X	1	2	3	4

Frame with `pause` tables

Table: Caption

Class	A	B	C	D
X	1	2	3	4
Y	3	4	5	6

Frame with `pause` tables

Table: Caption

Class	A	B	C	D
X	1	2	3	4
Y	3	4	5	6
Z	5	6	7	8

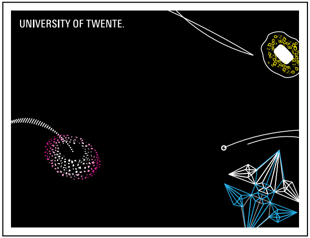
Frame with `pause` tables - `source`

```
\begin{frame}{Frame with \texttt{pause} tables}

\rowcolors[]{1}{blue!20}{blue!10}
\begin{table}
\caption{Caption}
\begin{tabular}{l!{\vrule}cccc}
Class & A & B & C & D \\\hline
X & 1 & 2 & 3 & 4 \pause \\
Y & 3 & 4 & 5 & 6 \pause \\
Z & 5 & 6 & 7 & 8
\end{tabular}
\end{table}
\end{frame}
```

Two Column Output

Text here.
Text here.
Text here.



Two Column Output - source

```
\begin{frame}{Two Column Output}
  \begin{columns}[c]
    \column{1.5in}
      Text here.\\
      Text here.\\
      Text here.
    \column{1.5in}
    \framebox{\includegraphics[width=1.5in]{img/back2}}
  \end{columns}
\end{frame}
```

Overview

Timed Automata

Symbolic Model Checking

LTSmin

Working with LaTeX

Second Section

Third Section

First frame of the Second Section

Each new section starts with an Table Of Contents.

Overview

Timed Automata

Symbolic Model Checking

LTSmin

Working with LaTeX

Second Section

Third Section

First frame of the Second Section

The Table Of Contents is clickable

41