Automata for Real-time Systems

B. Srivathsan

Chennai Mathematical Institute
Overview
Automata (*Finite State Machines*) are **good abstractions** of many real systems

hardware circuits, communication protocols, biological processes, ...
Automata can model many **properties** of systems

- Every request is followed by a response.

![Diagram showing the relationship between request and response.](image-url)
Does system satisfy property?

System
\Downarrow
Automaton \mathcal{A}

Property
\Downarrow
Automaton \mathcal{B}
Does system **satisfy** property?
System $\downarrow$ Automaton $A$

Property $\downarrow$ Automaton $B$

$L(A) \subseteq L(B)$?

Does system satisfy property?
Model-checking

System
↓
Automaton $A$

Property
↓
Automaton $B$

$L(A) \subseteq L(B)$?

Does system satisfy property?
In practice...

Huge system  Property

Some model-checkers: SMV, NuSMV, SPIN, ...

Turing Awards: Clarke, Emerson, Sifakis and Pnueli
In practice...

Huge system

↓

Higher-level description

Property

↓

Higher-level description

Some model-checkers: SMV, NuSMV, SPIN, ...

Turing Awards: Clarke, Emerson, Sifakis and Pnueli
In practice...

Huge system

Higher-level description

Automaton $A$

Property

Higher-level description

Automaton $B$

Model-Checker

$\mathcal{L}(A) \subseteq \mathcal{L}(B)$?
In practice...

Huge system

 Higher-level description

 Automaton $A$

 translation

 Model-Checker

\[ \mathcal{L}(A) \subseteq \mathcal{L}(B) ? \]

 Property

 Higher-level description

 Automaton $B$

 translation

 Some model-checkers: SMV, NuSMV, SPIN, …
In practice...

Huge system

Higher-level description

Automaton $A$

Property

Higher-level description

Automaton $B$

Model-Checker

$\mathcal{L}(A) \subseteq \mathcal{L}(B)$?

Some model-checkers: SMV, NuSMV, SPIN, ...
Automata are **good abstractions** of many real systems
Automata are **good abstractions** of many real systems

**Our course:** Automata for **real-time** systems

*Picture credits: F. Herbreteau*

pacemaker, vehicle control systems, air traffic controllers, ...
Timed Automata

R. Alur and D. Dill in early 90s
Timed Automata

R. Alur and D. Dill in early 90s

Some model-checkers: UPPAAL, KRONOS, RED, …
Goals of our course

Study *language theoretic* and *algorithmic* properties of timed automata
Lecture 7:
Timed languages and timed automata
\[ \Sigma : \text{alphabet} \quad \{a, b\} \]

\[ \Sigma^* : \text{words} \quad \{\varepsilon, a, b, aa, ab, ba, bb, aab, \ldots\} \]

\[ L \subseteq \Sigma^* : \text{language} \quad \rightarrow \quad \text{property over words} \]

\[ L_1 := \{\text{set of words starting with an “a”}\} \]
\[ \{a, aa, ab, aaa, aab, \ldots\} \]

\[ L_2 := \{\text{set of words with a non-zero even length}\} \]
\[ \{aa, bb, ab, ba, abab, aaaa, \ldots\} \]
\[ \Sigma \quad : \text{alphabet} \quad \{a, b\} \]

\[ \Sigma^* \quad : \text{words} \quad \{\varepsilon, a, b, aa, ab, ba, bb, aab, \ldots\} \]

\[ L \subseteq \Sigma^* \quad : \text{language} \quad \rightarrow \quad \text{property over words} \]

\[ L_1 := \{\text{set of words starting with an “} a \text{”}\} \]
\[ \{a, aa, ab, aaa, aab, \ldots\} \]

\[ L_2 := \{\text{set of words with a non-zero even length}\} \]
\[ \{aa, bb, ab, ba, abab, aaaa, \ldots\} \]

Finite automata, pushdown automata, Turing machines, \ldots
\[ \Sigma \text{ : alphabet} \quad \{a, b\} \]

\[ T\Sigma^* \text{ : timed words} \]

\begin{align*}
\text{(aa; 0.8, 2.5)} & \quad \text{(abb; } \pi, 203, 312.3) \\
\end{align*}
$\Sigma$ : alphabet \{\textit{a}, \textit{b}\}

$T\Sigma^*$ : timed words

Word $w = a_1 \ldots a_n$
$a_i \in \Sigma$

Time sequence $\tau = \tau_1 \ldots \tau_n$
$\tau_i \in \mathbb{R}_{\geq 0}$
$\tau_1 \leq \cdots \leq \tau_n$

$(aa; 0.8, 2.5)$

$(abb; \pi, 203, 312.3)$
$L \subseteq T\Sigma^*$: Timed language \rightarrow property over timed words

$L_1 := \{ (ab(a + b)^*, \tau) \mid \tau_2 - \tau_1 = 1 \}$

$\begin{array}{cccc}
a & b & ab & b \\
0 & 1 & 2 & \\
\end{array}$

$L_2 := \{ (w, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |w| \}$

$\begin{array}{cccc}
a & a \\
0 & 1.2 & 3.5 & 6 \\
\end{array}$
$L \subseteq T\Sigma^*$: Timed language → property over timed words

$L_1 := \{(ab(a + b)^*, \tau) \mid \tau_2 - \tau_1 = 1\}$

$L_2 := \{(w, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |w|\}$

Timed automata
Timed automaton: Finite automaton + Finite no. of Clocks

Clock

\[ \begin{align*}
\text{Clock} & \quad \text{time} \\
0 & \quad \tau
\end{align*} \]
Timed automaton: Finite automaton + Finite no. of *Clocks*

\[ \{ (ab(a + b)^*, \tau) \mid \tau_2 \leq 2 \} \]
Timed automaton: Finite automaton + Finite no. of Clocks

\[ \{ (ab(a+b)^*, \tau) \mid \tau_2 \leq 2 \} \]
Timed automaton: Finite automaton + Finite no. of Clocks

\[ \{ (ab(a + b)^*, \tau) \mid \tau_2 \leq 2 \} \]

\[
\begin{array}{c}
\begin{array}{c}
q_0 \\
q_1 \\
q_2
\end{array}
\end{array}
\begin{array}{c}
a \\
x \leq 2, b
\end{array}
\begin{array}{c}
a, b
\end{array}
\]

\[
\begin{array}{c}
\begin{array}{c}
0 \\
1 \\
2
\end{array}
\end{array}
\begin{array}{c}
a \\
b
\end{array}
\begin{array}{c}
q_0 \\
q_1 \\
q_2
\end{array}
\]

accept

\[
\begin{array}{c}
\begin{array}{c}
0 \\
1 \\
2
\end{array}
\end{array}
\begin{array}{c}
a \\
b
\end{array}
\begin{array}{c}
q_0 \\
q_1 \\
\times
\end{array}
\]

reject
Timed automaton: Finite automaton + Finite no. of *Clocks*

 Guards

\[ \phi := x \leq c \mid x \geq c \mid \neg \phi \mid \phi \land \phi \]

\[ x \in \text{Clocks}, \ c \in \mathbb{Q}_{\geq 0} \]

\[ \{ (ab(a + b)^*, \tau) \mid \tau_2 \leq 2 \} \]

![Diagram of a timed automaton with states and transitions](image)

- **Accept:** Path from \( q_0 \) through \( q_1 \) to \( q_2 \) with inputs \( a, b \) accepted.
- **Reject:** Path from \( q_0 \) through \( q_1 \) to \( q_0 \) with inputs \( a, b \) rejected.
Timed automaton: Finite automaton + Finite no. of *Clocks*

- **Clock**
  - Clock time
  - $0$
  - $\uparrow$
  - $\rightarrow$ time

- **Guards**
  - $\phi := x \leq c \mid x \geq c \mid \neg \phi \mid \phi \land \phi$
  - $x \in \text{Clocks}, \ c \in \mathbb{Q}_{\geq 0}$

- **Transitions**
  - $\{(ab(a + b)^*, \tau) \mid \tau_2 - \tau_1 \leq 2\}$

- **Diagram**
  - $q_0 \rightarrow a \rightarrow q_1 \rightarrow x \leq 2, b \rightarrow q_2 \rightarrow a, b$
Timed automaton: Finite automaton + Finite no. of Clocks

Guards

\[ \phi := x \leq c \mid x \geq c \mid \neg \phi \mid \phi \land \phi \]

\[ x \in \text{Clocks} \ , \ c \in \mathbb{Q}_{\geq 0} \]

Resets

\[ \{ (ab(a + b)^*, \tau) \mid \tau_2 - \tau_1 \leq 2 \} \]
Timed automaton: Finite automaton + Finite no. of Clocks

Clocks

 Guards

\[ \phi := x \leq c \mid x \geq c \mid \neg \phi \mid \phi \land \phi \]

\[ x \in \text{Clocks}, \ c \in \mathbb{Q}_{\geq 0} \]

 Resets

\[ \{ (ab(a + b)^*, \tau) \mid \tau_2 - \tau_1 \leq 2 \} \]

\[
\begin{array}{c}
q_0 \xrightarrow{a} q_1 \\
\{x\} \xrightarrow{x \leq 2, b} q_2
\end{array}
\]

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

\[ q_0 \]
\[ x:0 \]
\[ x \leq 2 \]

accept

\[
\begin{array}{c}
q_0 \xrightarrow{a} q_1 \\
\xrightarrow{x:0} q_2 \xrightarrow{\times} q_1
\end{array}
\]

reject

\[ q_0 \xrightarrow{bb} q_1 \xrightarrow{x:0} q_2 \xrightarrow{x > 2} q_1 \]
$L_3 := \{ (a^k, \tau) \mid k > 0, \tau_i = i \text{ for all } i \leq k \}$

An “a” occurs in every integer from 1, \ldots, k
\[ L_3 := \{ (\, a^k, \tau \, ) \mid k > 0, \, \tau_i = i \text{ for all } i \leq k \} \]

An “\( a \)” occurs in every integer from 1, \ldots, \( k \)
\[ L_4 := \{ (a^k, \tau) \mid \text{exist } i, j \text{ s.t. } \tau_j - \tau_i = 1 \} \]

There are 2 “a”s which are at distance 1 apart.
\[ L_4 := \{ (a^k, \tau) \mid \text{exist } i, j \text{ s.t. } \tau_j - \tau_i = 1 \} \]

There are 2 “a”s which are at distance 1 apart
Three mechanisms to exploit:

- **Reset**: to start measuring time
- **Guard**: to impose time constraint on action
- **Non-determinism**: for existential time constraints
\[ A = (Q, \Sigma, X, T, Q_0, F) \]

\[ T \subseteq Q \times \Sigma \times \text{guard} \times \text{reset} \times Q \]

The diagram shows a transition system with states \( s_0, s_1, s_2, s_3 \) and transitions labeled with input symbols and conditions:

- \( s_0 \) to \( s_1 \): \( a, (y < 1), \{y\} \)
- \( s_1 \) to \( s_2 \): \( c, (x < 1) \)
- \( s_2 \) to \( s_3 \): \( c, (x < 1) \)
- \( s_1 \) to \( s_3 \): \( a, (y < 1), \{y\} \)
- \( s_2 \) to \( s_3 \): \( d, (x > 1) \)
- \( s_0 \) to \( s_1 \): \( b, (y = 1) \)
\( A = (Q, \Sigma, X, T, Q_0, F) \)

\[ T \subseteq Q \times \Sigma \times \text{guard} \times \text{reset} \times Q \]
\( A = (Q, \Sigma, X, T, Q_0, F) \)

\( T \subseteq Q \times \Sigma \times \text{guard} \times \text{reset} \times Q \)

Run of \( A \) over \((a_1a_2 \ldots a_k; \tau_1\tau_2 \ldots \tau_k)\)

\[ (q_0, v_0) \xrightarrow{\delta_1} (q_0, v_0 + \delta_1) \xrightarrow{a_1} (q_1, v_1) \xrightarrow{\delta_2} (q_1, v_1 + \delta_2) \cdots \xrightarrow{a_k} (q_k, v_k) \]

\((w, \tau) \in \mathcal{L}(A) \) if \( A \) has an accepting run over \((w, \tau)\)
\[ L_5 := \{ (abcd.\Sigma^*, \tau) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5 \} \]

Interleaving distances
\[ L_5 := \{ (abcd.\Sigma^*, \tau) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5 \} \]

Interleaving distances
$n$ interleavings $\Rightarrow$ need $n$ clocks

$n + 1$ clocks more expressive than $n$ clocks
Timed automata

Runs
1 clock < 2 clocks < …
\[ L_6 := \{ (a^k, \tau) \mid \tau_i \text{ is some integer for each } i \} \]
$L_6 := \{ (a^k, \tau) \mid \tau_i \text{ is some integer for each } i \}$

Claim: **No timed automaton** can accept $L_6$
Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let $c_{\text{max}}$ be the maximum constant appearing in a guard of $A$.
Step 1: Suppose $L_6 = \mathcal{L}(A)$

Let $c_{\text{max}}$ be the maximum constant appearing in a guard of $A$

Step 2: For a clock $x$,

$$x = \lceil c_{\text{max}} \rceil + 1 \text{ and } x = \lceil c_{\text{max}} \rceil + 1.1$$

satisfy the same guards
Step 1: Suppose $L_6 = \mathcal{L}(A)$

Let $c_{\text{max}}$ be the maximum constant appearing in a guard of $A$

Step 2: For a clock $x$,

$x = \lceil c_{\text{max}} \rceil + 1$ and $x = \lceil c_{\text{max}} \rceil + 1.1$

satisfy the same guards

Step 3: $(a; \lceil c_{\text{max}} \rceil + 1) \in L_6$ and so $A$ has an accepting run

$$(q_0, v_0) \xrightarrow{\delta = \lceil c_{\text{max}} \rceil + 1} (q_0, v_0 + \delta) \xrightarrow{a} (q_F, v_F)$$
Step 1: Suppose $L_6 = \mathcal{L}(A)$
Let $c_{\text{max}}$ be the maximum constant appearing in a guard of $A$

Step 2: For a clock $x$,

\[ x = \lceil c_{\text{max}} \rceil + 1 \quad \text{and} \quad x = \lceil c_{\text{max}} \rceil + 1.1 \]

satisfy the same guards

Step 3: $(a; \lceil c_{\text{max}} \rceil + 1) \in L_6$ and so $A$ has an accepting run

\[
(q_0, v_0) \xrightarrow{\delta = \lceil c_{\text{max}} \rceil + 1} (q_0, v_0 + \delta) \xrightarrow{a} (q_F, v_F)
\]

Step 4: By Step 2, the following is an accepting run

\[
(q_0, v_0) \xrightarrow{\delta' = \lceil c_{\text{max}} \rceil + 1.1} (q_0, v_0 + \delta') \xrightarrow{a} (q_F, v'_F)
\]
Step 1: *Suppose* $L_6 = \mathcal{L}(A)$

Let $c_{\text{max}}$ be the maximum constant appearing in a guard of $A$

Step 2: For a clock $x$,

$x = \lceil c_{\text{max}} \rceil + 1$ and $x = \lceil c_{\text{max}} \rceil + 1.1$

satisfy the same guards

Step 3: $(a; \lceil c_{\text{max}} \rceil + 1) \in L_6$ and so $A$ has an accepting run

$$(q_0, v_0) \xrightarrow{\delta = \lceil c_{\text{max}} \rceil + 1} (q_0, v_0 + \delta) \xrightarrow{a} (q_F, v_F)$$

Step 4: By Step 2, the following is an accepting run

$$(q_0, v_0) \xrightarrow{\delta' = \lceil c_{\text{max}} \rceil + 1.1} (q_0, v_0 + \delta') \xrightarrow{a} (q_F, v'_F)$$

Hence $(a; \lceil c_{\text{max}} \rceil + 1.1) \in \mathcal{L}(A) \neq L_6$

Therefore **no timed automaton** can accept $L_6$
Timed automata

Runs

1 clock < 2 clocks < ... 

Role of max constant
Timed automata

Runs

1 clock < 2 clocks < ... 

Role of max constant

Timed regular lngs.
A timed language is called **timed regular** if it can be **accepted** by a timed automaton.
Timed regular languages are closed under union.
Timed regular languages are **closed** under intersection

\[
A = (Q, \Sigma, X, T, Q_0, F)
\]

\[
A' = (Q', \Sigma, X', T', Q'_0, F')
\]

\[
A \cap = (Q \times Q', \Sigma, X \cup X', T \cap, Q_0 \times Q'_0, F \times F')
\]

\[
T \cap : \quad (q_1, q'_1) \xrightarrow{a, g \land g'}_{R \cup R'} (q_2, q'_2) \text{ if } q_1 \xrightarrow{a, g}_{R} q_2 \in T \text{ and } q'_1 \xrightarrow{a, g'}_{R'} q'_2 \in T'
\]
\( L \) : a timed language over \( \Sigma \)

\[
\text{Untime}(L) \equiv \{ w \in \Sigma^* \mid \exists \tau. (w, \tau) \in L \}
\]

**Untiming construction**

For *every* timed automaton \( A \) there is a finite automaton \( A_u \) s.t.

\[
\text{Untime}( \mathcal{L}(A) ) = \mathcal{L}(A_u)
\]

more about this later . . .
Complementation

\[ \Sigma : \{a, b\} \]

\[ L = \{ (w, \tau) \mid \text{there is an } a \text{ at some time } t \text{ and no action occurs at time } t + 1 \} \]

\[ \overline{L} = \{ (w, \tau) \mid \text{every } a \text{ has an action at a distance 1 from it} \} \]
Complementation

\[ \Sigma : \{a, b\} \]

\[ L = \{ (w, \tau) \mid \text{there is an } a \text{ at some time } t \text{ and no action occurs at time } t + 1 \} \]

\[ \bar{L} = \{ (w, \tau) \mid \text{every } a \text{ has an action at a distance } 1 \text{ from it} \} \]

Claim: **No timed automaton** can accept \( \bar{L} \)

Decision problems for timed automata: A survey

Alur, Madhusudhan. *SFM'04: RT*
Step 1: \( \overline{L} = \{ (\omega, \tau) \mid \text{every } a \text{ has an action at a distance 1 from it} \} \)

*Suppose* \( \overline{L} \) is timed regular
Step 1: \( \overline{L} = \{ (w, \tau) \mid \text{every } a \text{ has an action at a distance 1 from it} \} \)

*Suppose* \( \overline{L} \) is timed regular

Step 2: Let \( L' = \{ (a^* b^*, \tau) \mid \text{all } a's \text{ occur before time 1 and no two } a's \text{ happen at same time} \} \)

Clearly \( L' \) is timed regular
Step 1: \( \overline{L} = \{ (w, \tau) \mid \text{every } a \text{ has an action at a distance 1 from it} \} \)

*Suppose \( \overline{L} \) is timed regular*

Step 2: Let \( L' = \{ (a^*b^*, \tau) \mid \text{all } a's \text{ occur before time 1 and no two } a's \text{ happen at same time} \} \)

Clearly \( L' \) is timed regular

Step 3: \( \text{Untime}( \overline{L} \cap L') \) should be a regular language

\( \frac{30}{35} \)
Step 1: $\overline{L} = \{ (w, \tau) \mid \text{every } a \text{ has an action at a distance 1 from it } \} $

*Suppose* $\overline{L}$ is timed regular

Step 2: Let $L' = \{ (a^* b^*, \tau) \mid \text{all } a's \text{ occur before time 1 and no two } a's \text{ happen at same time } \} $

Clearly $L'$ is timed regular

Step 3: Untime($\overline{L} \cap L'$) should be a regular language

Step 4: But, Untime($\overline{L} \cap L'$) = $\{ a^n b^m \mid m \geq n \}$, *not regular!*
Step 1: \( \bar{L} = \{ (w, \tau) \mid \text{every } a \text{ has an action at a distance 1 from it} \} \)

Suppose \( \bar{L} \) is timed regular

Step 2: Let \( L' = \{ (a^*b^*, \tau) \mid \text{all } a's \text{ occur before time 1 and no two } a's \text{ happen at same time} \} \)

Clearly \( L' \) is timed regular

Step 3: Untime( \( \bar{L} \cap L' \) ) should be a regular language

Step 4: But, Untime( \( \bar{L} \cap L' \) ) = \( \{a^n b^m \mid m \geq n\} \), not regular!

Therefore \( \bar{L} \) cannot be timed regular
Timed regular languages are not closed under complementation
Timed automata

- Runs
- 1 clock $<$ 2 clocks $<$ \ldots
- Role of max constant

Timed regular lns.

- Closure under $\cup$, $\cap$
- Non-closure under complement
Timed automata

Runs
1 clock < 2 clocks < …
Role of max constant

Timed regular lns.

Closure under $\cup$, $\cap$
Non-closure under complement

$\varepsilon$-transitions
Claim: No timed automaton can accept $L_6$
\[ L_6 := \{ (a^k, \tau) \mid \tau_i \text{ is some integer for each } i \} \]
$L_6 := \{ (a^k, \tau) \mid \tau_i \text{ is some integer for each } i \}$

\[
\begin{array}{cccccccc}
\text{a} & \varepsilon & \text{a} & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \text{a} \\
0 & 1 & 2 & 3 & 4 & 5 & 6 & 7
\end{array}
\]

Claim: No timed automaton can accept $L_6$.

\[x = 1, \varepsilon, \{x\}\]

\[x = 1, a, \{x\}\]
$\varepsilon$-transitions

$\varepsilon$-transitions add expressive power to timed automata.

Characterization of the expressive power of silent transitions in timed automata

Bérard, Diekert, Gastin, Petit. *Fundamenta Informaticae*’98
ε-transitions

ε-transitions add expressive power to timed automata. However, they add power only when a clock is reset in an ε-transition.

Characterization of the expressive power of silent transitions in timed automata

Bérard, Diekert, Gastin, Petit. Fundamenta Informaticae’98
Timed automata

Runs
1 clock < 2 clocks < ... 
Role of max constant

Timed regular lngs.

Closure under $\cup$, $\cap$
Non-closure under complement

$\varepsilon$-transitions

More expressive

$\varepsilon \rightarrow$ without reset $\equiv$ TA