Topics in Timed Automata

B. Srivathsan

RWTH-Aachen

Software modeling and Verification group
Timed Automata

A theory of timed automata

R. Alur and D. Dill, TCS’94
Timed Automata

Language theory
Lectures 1, ..., 5

Verification
Lectures 6, ..., 10

A theory of timed automata
R. Alur and D. Dill, TCS'94
Lecture 1:
Timed languages and timed automata
\[ \Sigma : \text{alphabet} \quad \{a, b\} \]

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

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

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

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

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

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

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

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

Finite automata, pushdown automata, Turing machines, …
Σ : alphabet \{a, b\}

\(T\Sigma^*\) : timed words

\[(aa; 0.8, 2.5)\]  
\[(abb; \pi, 203, 312.3)\]
\[ \Sigma : \text{alphabet} \quad \{a, b\} \]

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

\[
\begin{array}{c|ccc}
| & a & \quad & a \\
\hline
0 & 0.8 & 2.5
\end{array}
\]

\[
(aa; \ 0.8, 2.5)
\]

\[
\begin{array}{c|ccc}
| & a & b & b \\
\hline
0 & \pi & 203 & 312.3
\end{array}
\]

\[
(abb; \ \pi, 203, 312.3)
\]

Word \( \mathbf{w} = a_1 \ldots a_n \) \( \quad \)

Time sequence \( \mathbf{\tau} = \tau_1 \ldots \tau_n \)

\[ a_i \in \Sigma \quad \]

\[ \tau_i \in \mathbb{R}_{\geq 0} \quad \]

\[ \tau_1 \leq \cdots \leq \tau_n \]
$L \subseteq T \Sigma^*$: Timed language → property over timed words

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

$L_2 := \{(w, \tau) \mid \tau_{i+1} - \tau_i \geq 2 \text{ for all } i < |w|\}$
$L \subseteq T\Sigma^* : \text{Timed language} \longrightarrow \text{property over timed words}$

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

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

Timed automata
Timed automata
Timed automaton: Finite automaton + Finite no. of \textit{Clocks}

\[
\text{Clock} \\
\hline
\text{time} \\
0
\]
Timed automaton: Finite automaton + Finite no. of Clocks

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

\[
\begin{array}{c}
q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a,b} q_2
\end{array}
\]
Timed automaton: Finite automaton + Finite no. of *Clocks*

\[ \phi := x \leq c \mid x \geq c \mid \neg \phi \mid \phi \land \phi \mid x \in \text{Clocks}, c \in \mathbb{Q} \geq 0 \]

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

\[
q_0 \xrightarrow{a} q_1 \xrightarrow{x \leq 2, b} q_2 \xrightarrow{a, b} q_2
\]
Timed automaton: Finite automaton + Finite no. of *Clocks*

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

Diagram:

- **States:** \( q_0, q_1, q_2 \)
- **Transitions:**
  - \( q_0 \xrightarrow{a} q_1 \)
  - \( q_1 \xrightarrow{x \leq 2, b} q_2 \)
  - \( q_2 \xrightarrow{a, b} \)

- **Clocks:**
  - \( x \leq 2 \)

- **Accepting States:** \( q_2 \)
- **Rejecting State:** \( q_1 \)

- **Input Symbols:** \( a, b \)
Timed automaton: Finite automaton + Finite no. of *Clocks*

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

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

\[
q_0 \xrightarrow{a} q_1 \xrightarrow{x \leq 2, b} q_2 \xrightarrow{a, b} q_2
\]

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

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

accept  reject
Timed automaton: Finite automaton + Finite no. of \textit{Clocks}

Clock

\[ \text{Guards} \]

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

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

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

\[q_0 \xrightarrow{a} q_1 \xrightarrow{x \leq 2, b} q_2 \]

\[a, b\]
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**

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

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

- States: \( q_0, q_1, q_2 \)
- Transitions:
  - \( q_0 \xrightarrow{a} q_1 \) with \( \{x\} \)
  - \( q_1 \xrightarrow{x \leq 2, b} q_2 \)
  - \( q_2 \xrightarrow{a, b} \) (loop)

8/32
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

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

\[
\begin{aligned}
q_0 \xrightarrow{a} q_1 &\{x\} \quad x \leq 2, b \\
q_1 \xrightarrow{x \leq 2, b} q_2 \quad a, b
\end{aligned}
\]

- Transition table:

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>q_0</td>
<td>q_1</td>
<td>q_2</td>
</tr>
<tr>
<td>1</td>
<td>x : 0</td>
<td>x \leq 2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Labels:

- accept
- reject

- Transition table (continued):

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>bb</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>q_0</td>
<td>q_1</td>
</tr>
<tr>
<td>.5</td>
<td>x : 0</td>
<td>x &gt; 2</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Acceptance:

- Reject:

- Time points:
$L_3 := \{ (a^k, \tau) \mid k > 0, \tau_i = i \text{ for all } i \leq k \}$

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
$L_4 := \{ (a^k, \tau) | \text{exist } i, j \text{ s.t. } \tau_j - \tau_i = 1 \}$

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

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

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

The run of \( A \) over \((a, c; 0.4, 0.9)\)

- \( s_0 \) to \( s_1 \) on \( a \) and \( \{y\} \)
- \( s_1 \) to \( s_2 \) on \( b \) and \( (y = 1) \)
- \( s_1 \) to \( s_3 \) on \( c \) and \( (x < 1) \)
- \( s_3 \) to \( s_1 \) on \( d \) and \( (x > 1) \)

The table represents the transitions:

<table>
<thead>
<tr>
<th>State</th>
<th>Action</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>( s_0 )</td>
<td>( x = 0 )</td>
<td>0.4</td>
</tr>
<tr>
<td>( s_0 )</td>
<td>( x = 0 )</td>
<td>0.4</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>( a )</td>
<td>0.4</td>
</tr>
<tr>
<td>( s_1 )</td>
<td>( c )</td>
<td>0.5</td>
</tr>
<tr>
<td>( s_3 )</td>
<td>( c )</td>
<td>0.9</td>
</tr>
</tbody>
</table>

The transitions follow the labels and actions specified in the diagram.
\[(ac; 0.4, 0.9)\]

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

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

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

\[\delta_i := \tau_i - \tau_{i-1}; \quad \tau_0 := 0\]

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

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

Interleaving distances

\[ \begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
\bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\end{array} \]

Exercise: Prove that \( L_5 \) cannot be accepted by a one-clock TA.
\[ L_5 := \{ ( \text{abcd.}\Sigma^*, \tau) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5 \} \]

Interleaving distances

Exercise: Prove that \( L_5 \) cannot be accepted by a one-clock TA.
$$L_5 := \{ (abcd.\Sigma^*, \tau) \mid \tau_3 - \tau_1 \leq 2 \text{ and } \tau_4 - \tau_2 \geq 5 \}$$

Interleaving distances

Exercise: Prove that $L_5$ cannot be accepted by a one-clock TA.
$n$ interleavings ⇒ 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: \textit{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 $x$,

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

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

Step 2: For a clock $x$,

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

satisfy the same guards

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

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

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

\[
(q_0, v_0) \xrightarrow{\delta' = \lceil c_{max} \rceil + 1.1} (q_0, v_0 + \delta') \xrightarrow{a} (q_F, v'_F)
\]
Step 1: \textit{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 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 \textbf{no timed automaton} can accept $L_6$
\[ L_7 = \{ (a b)^k, \tau ) \mid \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \text{ for each } i \geq 1 \} \]

Converging \(ab\) distances
\[ L_7 = \left\{ ( (ab)^k, \tau ) \mid \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \text{ for each } i \geq 1 \right\} \]

Converging \(ab\) distances

Exercise: Prove that no timed automaton can accept \(L_7\)
\[ L_7 = \{ \left( (ab)^k, \tau \right) \mid \tau_{2i} = i \text{ and } \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \} \]

Pivoted converging \(ab\) distances
$L_7 = \{ ( (ab)^k, \tau ) \mid \tau_{2i} = i \text{ and } \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \}$

Pivoted converging $ab$ distances
\[ L_7 = \{ ( (ab)^k, \tau) \mid \tau_{2i} = i \text{ and } \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \} \]

Pivoted converging \( ab \) distances

\[
\tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \iff \tau_{2i+2} - \tau_{2i} < \tau_{2i+1} - \tau_{2i-1} \iff 1 < \tau_{2i+1} - \tau_{2i-1} \]
\[ L_7 = \{ ( (ab)^k, \tau ) \mid \tau_{2i} = i \text{ and } \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \} \]

Pivoted converging \( ab \) distances

\[ \tau_{2i+2} - \tau_{2i+1} < \tau_{2i} - \tau_{2i-1} \iff \tau_{2i+2} - \tau_{2i} < \tau_{2i+1} - \tau_{2i-1} \]
\[ \iff 1 < \tau_{2i+1} - \tau_{2i-1} \]
Timed automata

Runs

1 clock < 2 clocks < \ldots

Role of max constant
Timed automata

Runs

1 clock < 2 clocks < \ldots

Role of max constant

Timed regular lns.
Timed regular languages

Definition

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

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

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

\[ A_{\cup} = (Q \cup Q', \Sigma, X \cup X', T \cup T', Q_0 \cup Q'_0, F \cup F') \]

\[ \mathcal{L}(A) \cup \mathcal{L}(A') = \mathcal{L}(A_{\cup}) \]
Timed regular languages are **closed** under intersection
$L$ : a timed language over $\Sigma$

Untime($L$) $\equiv$ \{ $w \in \Sigma^*$ | $\exists \tau$. ($w$, $\tau$) $\in L$ \}

**Untiming construction**

For every timed automaton $A$ there is a finite automaton $A_u$ s.t.

Untime( $\mathcal{L}(A)$ ) = $\mathcal{L}(A_u)$

more about this in Lecture 6 . . .
Complementation

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

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

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

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

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

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

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

Decision problems for timed automata: A survey

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

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

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

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

Clearly \( L' \) is timed regular

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

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

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

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

Clearly \( L' \) is timed regular

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

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

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

Runs
1 clock < 2 clocks < ... 
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 lngs.

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

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

Claim:
No timed automaton can accept $L_6$.
\( \varepsilon \)-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
\(\varepsilon\)-transitions add expressive power to timed automata. However, they add power only when a clock is reset in an \(\varepsilon\)-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

Next lecture: Determinization
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

Next lecture: Determinization