Simulation Hemi-Metrics for Timed Systems, with Relations to Ditopology

Uli Fahrenberg

Department of Computer Science
Aalborg University
Denmark

ATMCS 2008

Motivation

- For real-time systems and specifications, timed bisimilarity is a rather merciless concept:

  The gates will be closed 1 minute before the train goes through

  not timed bisimilar to

  The gates will be closed 58 seconds before the train goes through

- Untimed bisimilarity on the other hand is, well, useless:

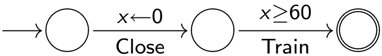
  The gates will be closed 1 minute before the train goes through

  untimed bisimilar to

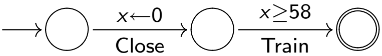
  The gates will be closed 1 second before the train goes through

Motivation

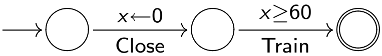
- Or, using timed automata:

$A =$ 

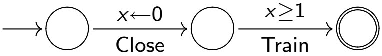
not timed bisimilar to

$B =$ 

- And for the other case:

$A =$ 

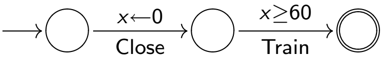
untimed bisimilar to

$C =$ 

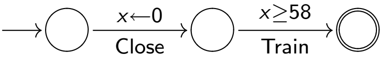
- Intuition: Want notion of bisimilarity up to ε – so that $A \sim_2 B$, but $A \sim_{59} C$.
- Bisimulation metrics

Motivation

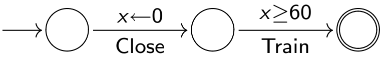
- Or, using timed automata:

$A =$ 

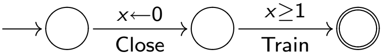
not timed bisimilar to

$B =$ 

- And for the other case:

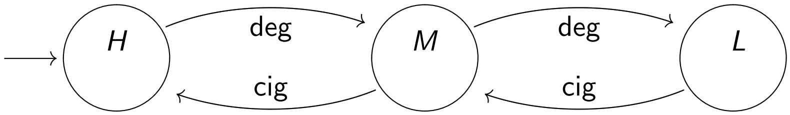
$A =$ 

untimed bisimilar to

$C =$ 

- Intuition: Want notion of bisimilarity up to ε – so that $A \sim_2 B$, but $A \sim_{59} C$.
- Bisimulation pseudometrics

Timed automata

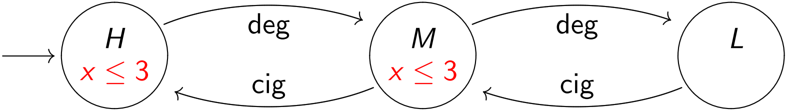


- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

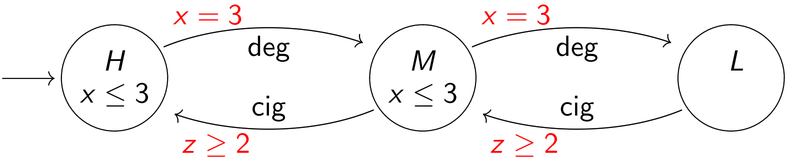
  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  $\Phi(C)$: clock constraints over C:

  $\varphi ::= x < k \mid x \leq k \mid x > k \mid x \geq k \mid \varphi_1 \wedge \varphi_2 \qquad (x \in C, k \in \mathbb{Z})$

Timed automata



- Timed automaton on alphabet Σ:

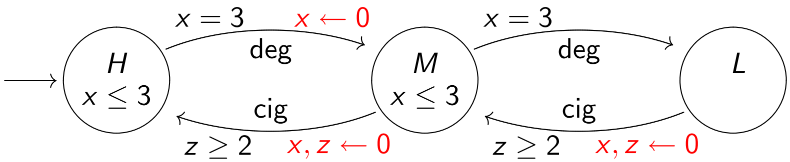
  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  $+$ edge constraints $c : E \to \Phi(C)$

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

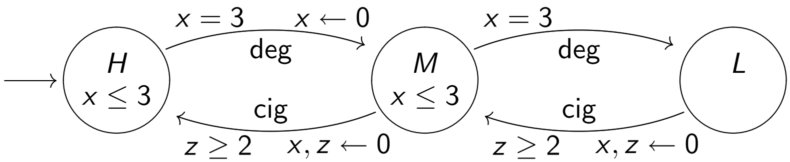
  \+ finite set C of clocks

  \+ location invariants $I : Q \to \Phi(C)$

  \+ edge constraints $c : E \to \Phi(C)$

  \+ edge resets $r : E \to 2^C$

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

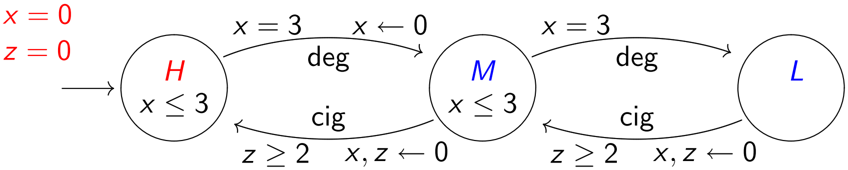
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

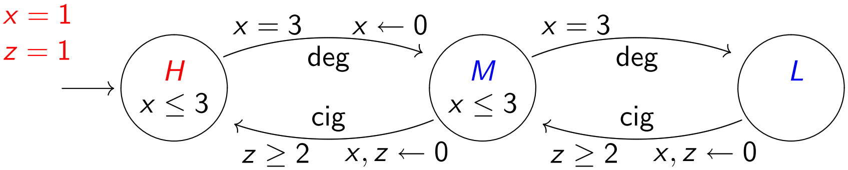
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

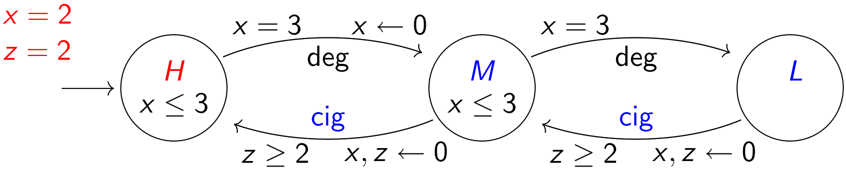
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  + finite set C of clocks

  + location invariants $I : Q \to \Phi(C)$

    location q is enabled iff clock values satisfy $I(q)$

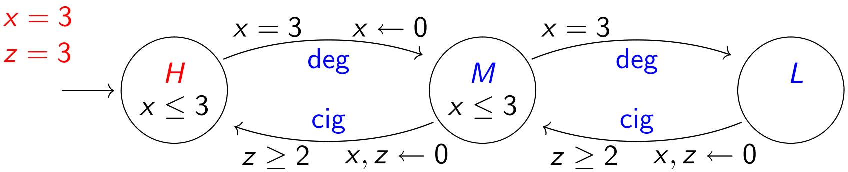
  + edge constraints $c : E \to \Phi(C)$

    edge e is enabled iff clock values satisfy $c(e)$

  + edge resets $r : E \to 2^C$

    on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  \+ finite set C of clocks

  \+ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

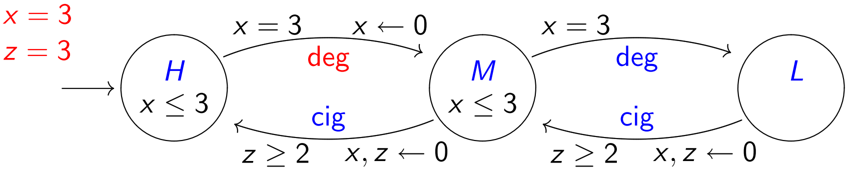
  \+ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  \+ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

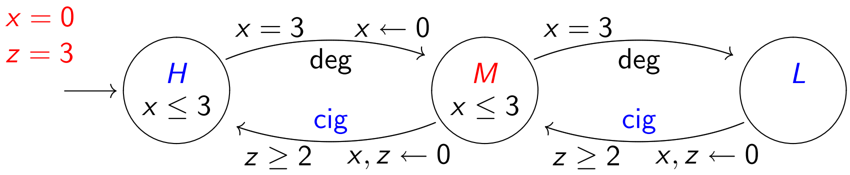
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

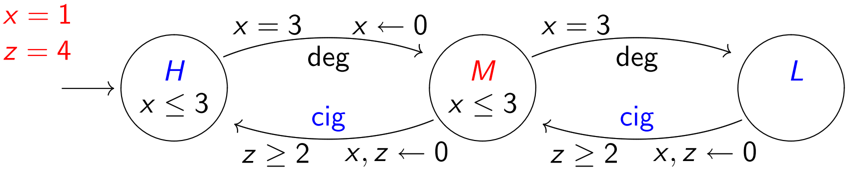
  + finite set C of clocks
  + location invariants $I : Q \to \Phi(C)$

    location q is enabled iff clock values satisfy $I(q)$
  + edge constraints $c : E \to \Phi(C)$

    edge e is enabled iff clock values satisfy $c(e)$
  + edge resets $r : E \to 2^C$

    on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

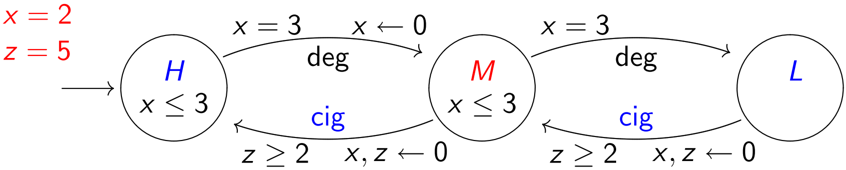
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

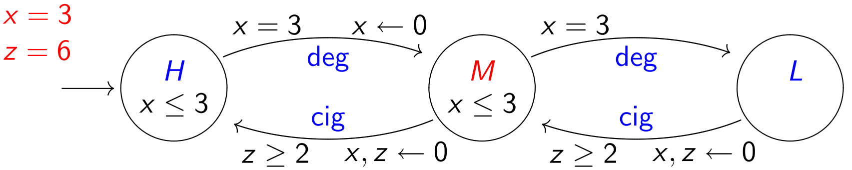
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

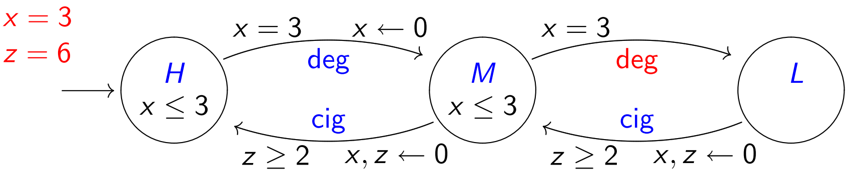
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

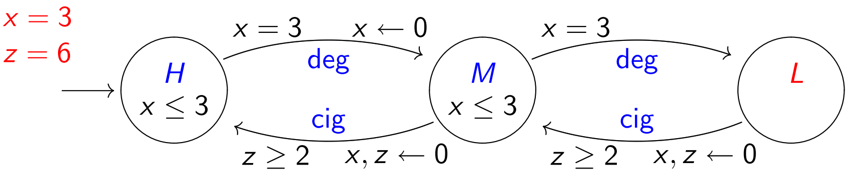
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

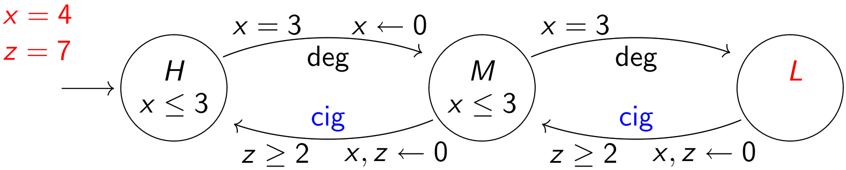
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

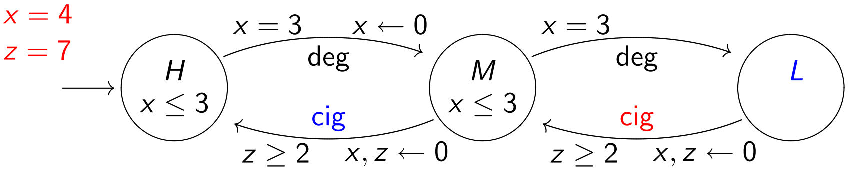
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

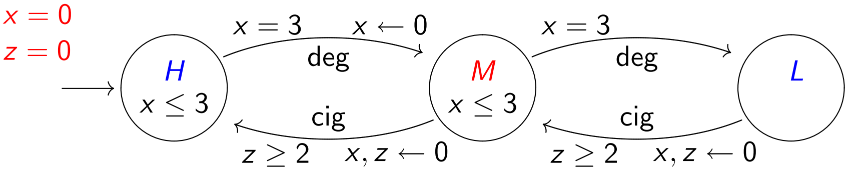
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  \+ finite set C of clocks

  \+ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

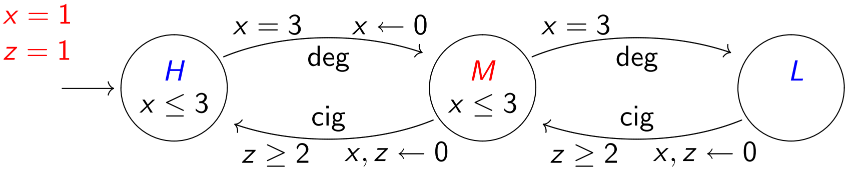
  \+ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  \+ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

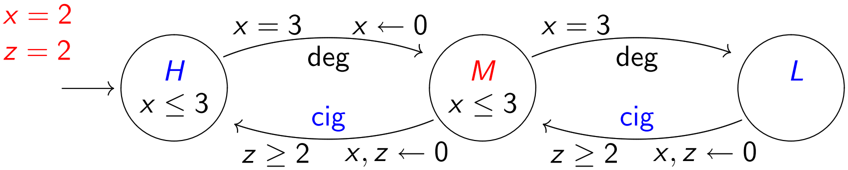
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  \+ finite set C of clocks

  \+ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

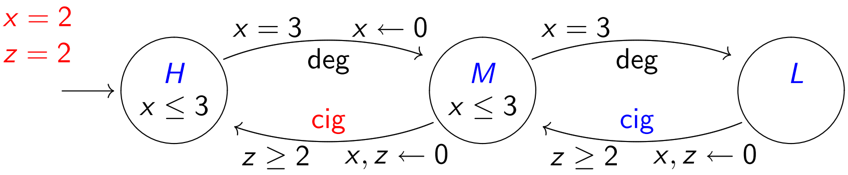
  \+ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  \+ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  $+$ finite set C of clocks

  $+$ location invariants $I : Q \to \Phi(C)$

  location q is enabled iff clock values satisfy $I(q)$

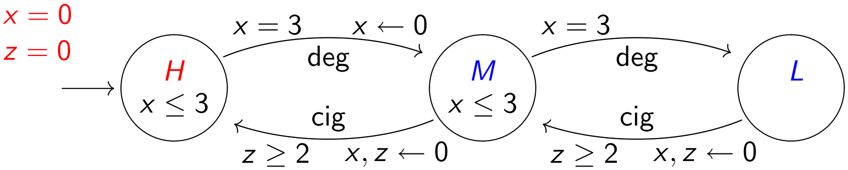
  $+$ edge constraints $c : E \to \Phi(C)$

  edge e is enabled iff clock values satisfy $c(e)$

  $+$ edge resets $r : E \to 2^C$

  on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata

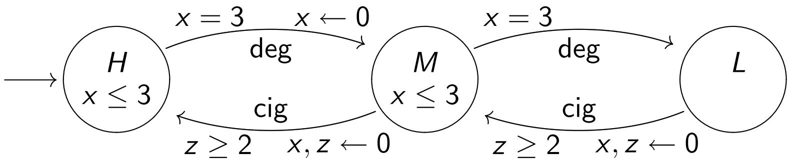


- Timed automaton on alphabet Σ:

  Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$

  + finite set C of clocks
  + location invariants $I : Q \to \Phi(C)$
    location q is enabled iff clock values satisfy $I(q)$
  + edge constraints $c : E \to \Phi(C)$
    edge e is enabled iff clock values satisfy $c(e)$
  + edge resets $r : E \to 2^C$
    on transition along edge e, values of clocks in $r(e)$ are reset to 0

Timed automata



- Timed automaton on alphabet Σ:
  - Finite automaton $(Q, q_0, E \subseteq Q \times \Sigma \times Q)$
  - $+$ finite set C of clocks
  - $+$ location invariants $I : Q \to \Phi(C)$
  - $+$ edge constraints $c : E \to \Phi(C)$
  - $+$ edge resets $r : E \to 2^C$
- Semantics of timed automata as (uncountable!) timed transition systems

Timed traces

- Recap: Want bisimulation pseudometrics on timed automata (or, technically, on timed transition systems)
- Easier to define: metrics on timed languages (in the "linear domain")
- Timed automata generate timed traces:

$$L(A) = \Big\{ (t_0, a_0, t_1, a_1, \dots) \;\Big|\; \text{exists alternating path } s_0 \xrightarrow{t_0} s_0' \xrightarrow{a_0} s_1 \xrightarrow{t_1} s_1' \xrightarrow{a_1} \cdots \text{ in } A \Big\}$$

(In this talk, we consider only infinite timed traces)

- Coming up: Different metrics on timed traces $\rightsquigarrow$ Hausdorff metrics construction $\rightsquigarrow$ different metrics on timed languages

Metrics on timed traces

- Two timed traces: $\tau = (t_0, a_0, t_1, a_1, t_2, a_2, \dots)$
  $\tau' = (t'_0, a'_0, t'_1, a'_1, t'_2, a'_2 \dots)$
- If $a_i \neq a'_i$ for some i (difference in actions), we set $d(\tau, \tau') = \infty$.
- Otherwise: $d_{\text{pair}}(\tau, \tau') = \sup_i \big\{ |t_i - t'_i| \big\}$

$$d_{\text{sum}}(\tau, \tau') = \sup_i \Big\{ \big| \textstyle\sum_{j=1}^{i} t_j - \sum_{j=1}^{i} t'_j \big| \Big\}$$

$$d_{\text{pair,drift}}(\tau, \tau') = \log \Big(\sup_i \Big\{ \max \Big(\frac{t_i}{t'_i}, \frac{t'_i}{t_i} \Big) \Big\} \Big)$$

$$d_{\text{sum,drift}}(\tau, \tau') = \log \Big(\sup_i \Big\{ \max \Big(\frac{\sum_{j=1}^{i} t_j}{\sum_{j=1}^{i} t'_j}, \frac{\sum_{j=1}^{i} t'_j}{\sum_{j=1}^{i} t_j} \Big) \Big\} \Big)$$

Metrics on timed traces

- Two timed traces: $\tau = (t_0, a_0, t_1, a_1, t_2, a_2, \dots)$
  $\tau' = (t'_0, a'_0, t'_1, a'_1, t'_2, a'_2 \dots)$
- If $a_i \neq a'_i$ for some i (difference in actions), we set $d(\tau, \tau') = \infty$.
- Otherwise: $d_{\text{pair}}(\tau, \tau') = \sup_i \{|t_i - t'_i|\}$

(measures maximal difference in pairs of delays)

$$d_{\text{sum}}(\tau, \tau') = \sup_i \{|\textstyle\sum_{j=1}^i t_j - \sum_{j=1}^i t'_j|\}$$

$$d_{\text{pair,drift}}(\tau, \tau') = \log\Big(\sup_i\Big\{\max\Big(\frac{t_i}{t'_i}, \frac{t'_i}{t_i}\Big)\Big\}\Big)$$

$$d_{\text{sum,drift}}(\tau, \tau') = \log\Big(\sup_i\Big\{\max\Big(\frac{\sum_{j=1}^i t_j}{\sum_{j=1}^i t'_j}, \frac{\sum_{j=1}^i t'_j}{\sum_{j=1}^i t_j}\Big)\Big\}\Big)$$

Metrics on timed traces

- Two timed traces: $\tau = (t_0, a_0, t_1, a_1, t_2, a_2, \dots)$
  $\tau' = (t'_0, a'_0, t'_1, a'_1, t'_2, a'_2 \dots)$
- If $a_i \neq a'_i$ for some i (difference in actions), we set $d(\tau, \tau') = \infty$.
- Otherwise: $d_{\text{pair}}(\tau, \tau') = \sup_i \{ |t_i - t'_i| \}$

(measures maximal difference in pairs of delays)

$$d_{\text{sum}}(\tau, \tau') = \sup_i \left\{ \left| \sum_{j=1}^i t_j - \sum_{j=1}^i t'_j \right| \right\}$$

(measures maximal difference in accumulated delay)

$$d_{\text{pair,drift}}(\tau, \tau') = \log \left(\sup_i \left\{ \max \left(\frac{t_i}{t'_i}, \frac{t'_i}{t_i} \right) \right\} \right)$$

$$d_{\text{sum,drift}}(\tau, \tau') = \log \left(\sup_i \left\{ \max \left(\frac{\sum_{j=1}^i t_j}{\sum_{j=1}^i t'_j}, \frac{\sum_{j=1}^i t'_j}{\sum_{j=1}^i t_j} \right) \right\} \right)$$

Metrics on timed traces

- Two timed traces: $\tau = (t_0, a_0, t_1, a_1, t_2, a_2, \dots)$
  $\tau' = (t'_0, a'_0, t'_1, a'_1, t'_2, a'_2 \dots)$
- If $a_i \ne a'_i$ for some i (difference in actions), we set $d(\tau, \tau') = \infty$.
- Otherwise: $d_{\text{pair}}(\tau, \tau') = \sup_i \left\{ |t_i - t'_i| \right\}$

(measures maximal difference in pairs of delays)

$$d_{\text{sum}}(\tau, \tau') = \sup_i \left\{ \left| \sum_{j=1}^{i} t_j - \sum_{j=1}^{i} t'_j \right| \right\}$$

(measures maximal difference in accumulated delay)

$$d_{\text{pair,drift}}(\tau, \tau') = \log \left(\sup_i \left\{ \max \left(\frac{t_i}{t'_i}, \frac{t'_i}{t_i} \right) \right\} \right)$$

$$d_{\text{sum,drift}}(\tau, \tau') = \log \left(\sup_i \left\{ \max \left(\frac{\sum_{j=1}^{i} t_j}{\sum_{j=1}^{i} t'_j}, \frac{\sum_{j=1}^{i} t'_j}{\sum_{j=1}^{i} t_j} \right) \right\} \right)$$

(similar, but now we measure quotients (drift) instead of difference)

Metrics on timed traces

$$d_{\text{pair}}(\tau,\tau') = \sup_i \left\{ \left| t_i - t'_i \right| \right\}$$
$$d_{\text{sum}}(\tau,\tau') = \sup_i \left\{ \left| \sum_{j=1}^{i} t_j - \sum_{j=1}^{i} t'_j \right| \right\}$$
$$d_{\text{pair,drift}}(\tau,\tau') = \log\left(\sup_i \left\{ \max\left(\frac{t_i}{t'_i}, \frac{t'_i}{t_i} \right) \right\} \right)$$
$$d_{\text{sum,drift}}(\tau,\tau') = \log\left(\sup_i \left\{ \max\left(\frac{\sum_{j=1}^{i} t_j}{\sum_{j=1}^{i} t'_j}, \frac{\sum_{j=1}^{i} t'_j}{\sum_{j=1}^{i} t_j} \right) \right\} \right)$$

- For all of the above, $d(\tau,\tau') = 0$ implies $\tau = \tau'$ (hence they are indeed metrics)
- General p-metrics can be defined – above are the cases $p = \infty$; for $p = 1$ *e.g.*, $\sup_i$ is replaced by $\sum_i$
- The above four are not topologically equivalent

Metrics on timed traces

$$d_{\text{pair}}(\tau, \tau') = \sup_i \left\{ |t_i - t_i'| \right\}$$

$$d_{\text{sum}}(\tau, \tau') = \sup_i \left\{ \left| \sum_{j=1}^i t_j - \sum_{j=1}^i t_j' \right| \right\}$$

$$d_{\text{pair,drift}}(\tau, \tau') = \log \left(\sup_i \left\{ \max \left(\frac{t_i}{t_i'}, \frac{t_i'}{t_i} \right) \right\} \right)$$

$$d_{\text{sum,drift}}(\tau, \tau') = \log \left(\sup_i \left\{ \max \left(\frac{\sum_{j=1}^i t_j}{\sum_{j=1}^i t_j'}, \frac{\sum_{j=1}^i t_j'}{\sum_{j=1}^i t_j} \right) \right\} \right)$$

- For all of the above, $d(\tau, \tau') = 0$ implies $\tau = \tau'$ (hence they are indeed metrics)
- General p-metrics can be defined – above are the cases $p = \infty$; for $p = 1$ *e.g.*, $\sup_i$ is replaced by $\sum_i$
- The above four are not topologically equivalent
- (Two metrics, d_1 and d_2, are topologically equivalent iff they generate the same topology, iff there are constants m and M such that $md_1(x, y) \leq d_2(x, y) \leq Md_1(x, y)$ for all x, y)

Pseudometrics on timed languages

- For measuring differences of timed languages (which is what we want), use Hausdorff pseudometric:

  Given a set X with pseudometric d, the Hausdorff pseudometric on the power set of X is d^{H} defined as follows:

$$d^{\mathrm{H}}(A, B) = \max\left(\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b) \right)$$

- Hence for timed languages L_1, L_2 we have $d(L_1, L_2) \le \varepsilon$ iff any timed trace in L_1 can be matched by a timed trace in L_2 with distance $\le \varepsilon$, and vice versa – quite natural!
- So we have metrics $d^{\mathrm{H}}_{\mathrm{pair}}$, $d^{\mathrm{H}}_{\mathrm{sum}}$, $d^{\mathrm{H}}_{\mathrm{pair,drift}}$, $d^{\mathrm{H}}_{\mathrm{sum,drift}}$ for timed languages
- And $d^{\mathrm{H}}(L_1, L_2) = 0$ iff $\overline{L_1} = \overline{L_2}$ (topological closure)
- Lemma: Two pseudometrics are topologically equivalent iff their Hausdorff pseudometrics are.

Bisimulation pseudometrics

- Problem: for timed automata A, B, it is undecidable whether $\overline{L(A)} = \overline{L(B)}$
- *i.e.* it is undecidable whether $d(L(A), L(B)) = 0$
- hence all our pseudometrics on timed languages are most probably uncomputable in general!

Bisimulation pseudometrics

- Problem: for timed automata A, B, it is undecidable whether $\overline{L(A)} = \overline{L(B)}$
- *i.e.* it is undecidable whether $d(L(A), L(B)) = 0$
- hence all our pseudometrics on timed languages are most probably uncomputable in general!

- Back to the "branching domain": It is decidable whether two timed automata are bisimilar

$\Longrightarrow$ Want to introduce bisimulation pseudometrics on timed automata which correspond to our pseudometrics on timed languages

- correspond should mean: $d(A, B) = \varepsilon < \infty \Longrightarrow d\big(L(A), L(B)\big) = \varepsilon$
- in other words: For automata with finite bisimulation distance, the language mapping should be distance preserving.

Bisimulation pseudometrics

- Pair version: For states s_1, s_2 in timed transition systems A, B, say that $s_1 \sim_\varepsilon^{\text{pair}} s_2$ iff

$$\begin{aligned} &\forall s_1 \xrightarrow{a} s_1' \in A : \exists s_2 \xrightarrow{a} s_2' \in B : s_1' \sim_\varepsilon^{\text{pair}} s_2' \\ \wedge\ &\forall s_2 \xrightarrow{a} s_2' \in B : \exists s_1 \xrightarrow{a} s_1' \in A : s_1' \sim_\varepsilon^{\text{pair}} s_2' \\ \wedge\ &\forall s_1 \xrightarrow{t_1} s_1' \in A : \exists s_2 \xrightarrow{t_2} s_2' \in B : s_1' \sim_\varepsilon^{\text{pair}} s_2' \wedge |t_1 - t_2| \le \varepsilon \\ \wedge\ &\forall s_2 \xrightarrow{t_2} s_2' \in B : \exists s_1 \xrightarrow{t_1} s_1' \in A : s_1' \sim_\varepsilon^{\text{pair}} s_2' \wedge |t_1 - t_2| \le \varepsilon \end{aligned}$$

- (Recall that for timed traces, $d_{\text{pair}}(\tau, \tau') = \sup_i \{|t_i - t_i'|\}$)
- Define $d_{\text{pair}}(A, B) = \inf\{\varepsilon \mid A \sim_\varepsilon^{\text{pair}} B\}$
- Then the L mapping is indeed distance preserving
- Similar can be done for $d_{\text{pair,drift}}$
- What about computability?

Bisimulation pseudometrics

- The sum version is more difficult: Need to remember differences in delays across transitions
- For states s_1, s_2 in timed transition systems A, B, say that $s_1 \sim^{\text{sum}}_{\varepsilon,\delta} s_2$ iff

$$\begin{aligned} &\forall s_1 \xrightarrow{a} s_1' \in A : \exists s_2 \xrightarrow{a} s_2' \in B : s_1' \sim^{\text{sum}}_{\varepsilon,\delta} s_2' \\ \land\ &\forall s_2 \xrightarrow{a} s_2' \in B : \exists s_1 \xrightarrow{a} s_1' \in A : s_1' \sim^{\text{sum}}_{\varepsilon,\delta} s_2' \\ \land\ &\forall s_1 \xrightarrow{t_1} s_1' \in A : \exists s_2 \xrightarrow{t_2} s_2' \in B : s_1' \sim^{\text{sum}}_{\varepsilon,\delta+t_1-t_2} s_2' \land |\delta + t_1 - t_2| \leq \varepsilon \\ \land\ &\forall s_2 \xrightarrow{t_2} s_2' \in B : \exists s_1 \xrightarrow{t_1} s_1' \in A : s_1' \sim^{\text{sum}}_{\varepsilon,\delta+t_1-t_2} s_2' \land |\delta + t_1 - t_2| \leq \varepsilon \end{aligned}$$

- (δ is the lead which A hitherto has worked up compared to B)
- Define $d_{\text{sum}}(A, B) = \inf\{\varepsilon \mid A \sim^{\text{sum}}_{\varepsilon,0} B\}$
- This is work by Henzinger, Majumdar, Prabhu (FORMATS 2005)
- (Similar can be done for $d_{\text{sum,drift}}$)
- Yes, the L mapping is again distance preserving
- And HMP'05 shows that d_{sum} is computable!

Summary

What we have:

- Four different interesting pseudometrics on the set **TA** of timed automata (or, if you wish, on the set **TS** of timed transition systems)
- For each of them, a corresponding pseudometric on the set **TL** of timed languages
- such that the language mapping $L : \mathbf{TA} \to \mathbf{TL}$ is continuous and distance preserving

Summary

What we have:

- Four different interesting pseudometrics on the set **TA** of timed automata (or, if you wish, on the set **TS** of timed transition systems)
- For each of them, a corresponding pseudometric on the set **TL** of timed languages
- such that the language mapping $L : \mathbf{TA} \to \mathbf{TL}$ is continuous and distance preserving

What we want to know:

- Computability: One of the bisimulation pseudometrics is computable; what about the other three?
- Feasibility: Even though this pseudometric is computable, the algorithm is not in any way feasible. But maybe there are other, feasible, algorithms?

Summary

What we have:

- Four different interesting pseudometrics on the set **TA** of timed automata (or, if you wish, on the set **TS** of timed transition systems)
- For each of them, a corresponding pseudometric on the set **TL** of timed languages
- such that the language mapping $L : \mathbf{TA} \to \mathbf{TL}$ is continuous and distance preserving

What we also want to know:

- Topological properties of **TA**, **TS**, and **TL** with these pseudometrics:
  - not T_0
  - the four topologies on **TA** are not the same
  - neither are there any refinement relations
  - More!

Summary

What we have:

- Four different interesting pseudometrics on the set **TA** of timed automata (or, if you wish, on the set **TS** of timed transition systems)
- For each of them, a corresponding pseudometric on the set **TL** of timed languages
- such that the language mapping $L : \mathbf{TA} \to \mathbf{TL}$ is continuous and distance preserving

What we also want to know:

- Properties of the L mapping
  - expecially interesting: What can be said about $d(L(A), L(B))$ for points $A, B \in \mathbf{TA}$ with $d(A, B) = \infty$
  - Conjecture: $d(A, B) = \infty$ implies $d(L(A), L(B)) = \infty$ or $d(L(A), L(B)) = 0$ (for all four pseudometrics).

Summary

What we have:

- Four different interesting pseudometrics on the set **TA** of timed automata (or, if you wish, on the set **TS** of timed transition systems)
- For each of them, a corresponding pseudometric on the set **TL** of timed languages
- such that the language mapping $L : \mathbf{TA} \to \mathbf{TL}$ is continuous and distance preserving

What we also want to know:

- Properties of the associated metric spaces $\mathbf{TA}_* = \mathbf{TA}/_{d(A,B)=0}$, $\mathbf{TS}_*$, $\mathbf{TL}_*$
  - (these are metric, hence nice spaces)
  - What about the properties of the induced L_* mapping?

Simulation, and directed topology

Simulation hemimetrics

- Like bisimulation, but one-way:
- Pair version: For states s_1, s_2 in timed transition systems A, B, say that $s_1 \preceq_\varepsilon^{\text{pair}} s_2$ iff

$$\forall s_1 \xrightarrow{a} s_1' \in A : \exists s_2 \xrightarrow{a} s_2' \in B : s_1' \preceq_\varepsilon^{\text{pair}} s_2'$$

$$\wedge \ \forall s_1 \xrightarrow{t_1} s_1' \in A : \exists s_2 \xrightarrow{t_2} s_2' \in B : s_1' \preceq_\varepsilon^{\text{pair}} s_2' \wedge |t_1 - t_2| \le \varepsilon$$

- Sum version: Say that $s_1 \preceq_{\varepsilon,\delta}^{\text{sum}} s_2$ iff

$$\forall s_1 \xrightarrow{a} s_1' \in A : \exists s_2 \xrightarrow{a} s_2' \in B : s_1' \preceq_{\varepsilon,\delta}^{\text{sum}} s_2'$$

$$\wedge \ \forall s_1 \xrightarrow{t_1} s_1' \in A : \exists s_2 \xrightarrow{t_2} s_2' \in B : s_1' \preceq_{\varepsilon,\delta+t_1-t_2}^{\text{sum}} s_2' \wedge |\delta + t_1 - t_2| \le \varepsilon$$

- and define $\vec{d}_{\text{pair}}(A,B) = \inf\{\varepsilon \mid A \preceq_\varepsilon^{\text{pair}} B\}$, $\vec{d}_{\text{sum}}(A,B) = \inf\{\varepsilon \mid A \preceq_{\varepsilon,0}^{\text{sum}} B\}$
- these are hemimetrics (or δ-metrics; asymmetric pseudometrics)
- (and $\vec{d}_{\text{sum}}$ is also in HMP05, and is computable)
- (and $\vec{d}_{\text{pair,drift}}$, $\vec{d}_{\text{sum,drift}}$ can be defined similarly)

Hemimetrics on timed languages

- Hausdorff hemimetric:

  Given a set X with hemimetric $\vec{d}$, the Hausdorff hemimetric on the power set of X is $\vec{d}^{\mathrm{H}}$ defined as follows:

  $$\vec{d}^{\mathrm{H}}(A, B) = \sup_{a \in A} \inf_{b \in B} \vec{d}(a, b)$$

- hence: have hemimetrics $\vec{d}_{\text{pair}}$, $\vec{d}_{\text{sum}}$, $\vec{d}_{\text{pair,drift}}$, $\vec{d}_{\text{sum,drift}}$ on **TA**, **TS**, and **TL**
- and the L mapping **TA** $\to$ **TL** (or **TS** $\to$ **TL** if you wish) is d-distance preserving

From hemimetrics to d-spaces

- Marco Grandis, "The Fundamental Weighted Category of a Weighted Space", HHA 9 (2007) is paving the way from hemimetrics to directed spaces:
  - Given a set X with hemimetric $\vec{d}$, define a metric on X by $\bar{d}(a, b) = \min(\vec{d}(a, b), \vec{d}(b, a))$,
  - take the topology on X generated by $\bar{d}$,
  - and say that a continuous path $p : I \to X$ is directed if $\sup\{\sum_{i=1}^{p} \vec{d}(a_{i-1}, a_i) \mid 0 = t_0 < \cdots < t_p = 1, p \in \mathbb{N}\}$ is finite.
- Interpretation:
  - $\bar{d}$, *i.e.* the topology, measures how close one of the systems is to the other
  - Along directed paths, "completeness" of systems is increasing: For a directed path p and $t \leq t'$, the system $p(t')$ simulates $p(t)$ up to $\vec{d}(p(t), p(t'))$

Conclusion

What we have:

- Four different interesting hemimetrics on each of **TA**, **TS**, and **TL**
- such that the L mapping is (continuous and) d-distance preserving
- An interesting interpretation of the d-spaces arising from these hemimetrics

What we would like to know:

- Properties of these d-spaces:
  - not T_0; saturated; not locally partially ordered
  - maybe convenient in the sense of Fajstrup-Rosický? maybe streams in the sense of Sanjeevi? what about a cubical structure?
- We know what d-paths "mean". What about d-homotopies?
- Hemimetrics give also rise to w-spaces (also in Grandis07). Do these have an interesting interpretation?
- *etc.*