Playing Games with Metrics

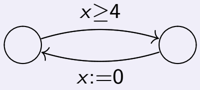
Uli Fahrenberg   Claus Thrane   Kim G. Larsen

CS @ Aalborg

Maths @ Aalborg seminar
Nov. 2010

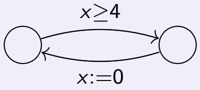
Quantitative Analysis

Quantitative *Models*



Quantitative Quantitative Analysis

Quantitative *Models*

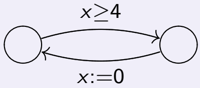


Quantitative *Logics*

$$\mathrm{Pr}_{\leq .1}(\Diamond \mathit{error})$$

Quantitative Quantitative Quantitative Analysis

Quantitative *Models*



Quantitative *Logics*

$$\Pr_{\leq .1}(\Diamond \mathit{error})$$

Quantitative *Verification*

$$[\![\varphi]\!](s) = 3.14$$
$$d(s, t) = 42$$

Boolean world	**"Quantification"**
Trace equivalence $\equiv$	Linear distance d_L
Bisimilarity $\sim$	Branching distance d_B
$s \sim t$ implies $s \equiv t$	$d_L(s, t) \leq d_B(s, t)$

Weighted Automata and Traces

Definition

A weighted automaton: states S, transitions $T \subseteq S \times \mathbb{K} \times S$

- $\mathbb{K}$: Set of weights.
- Standard example: $\mathbb{K} = L \times \mathbb{R}$. Discrete labels L, real weights $\mathbb{R}$.

Definition

A trace is an infinite sequence of weights; an element of $\mathbb{K}^\omega$.

- Notation: For $s \in S$ in a weighted automaton (S, T), $\mathrm{Tr}(s)$ is the set of traces from s.

Framework for Quantitative Analysis

Trace distance

Assume given a hemimetric $d_T : \mathbb{K}^\omega \times \mathbb{K}^\omega \to [0, \infty]$.

That's it. We only assume some way to measure distance between traces.

- Think of the trace distance as application defined
- May or may not come from some metric on $\mathbb{K}$

(Hemimetric: not necessarily symmetric pseudometric:

- $d_T(x, x) = 0$ (indiscernibility of identicals)
- $d_T(x, y) + d_T(y, z) \geq d_T(x, z)$ (triangle inequality))

Examples of Trace Distances

- Let $\mathbb{K} = L \times \mathbb{R}$. Notation: Trace $\sigma = \big((\sigma_0^\ell, \sigma_0^w), (\sigma_1^\ell, \sigma_1^w), \dots\big)$.

Point-wise trace distance

$$d_T^\bullet(\sigma, \tau) = \begin{cases} \sup_i \quad |\sigma_i^w - \tau_i^w| & \text{if } \sigma_i^\ell = \tau_i^\ell \text{ for all } i \\ \infty & \text{otherwise} \end{cases}$$

Accumulating trace distance

$$d_T^+(\sigma, \tau) = \begin{cases} \sum_i \quad |\sigma_i^w - \tau_i^w| & \text{if } \sigma_i^\ell = \tau_i^\ell \text{ for all } i \\ \infty & \text{otherwise} \end{cases}$$

Maximum-lead trace distance

$$d_T^\pm(\sigma, \tau) = \begin{cases} \sup_i \big|\sum_{j=0}^i \sigma_j^w - \sum_{j=0}^i \tau_j^w\big| & \text{if } \sigma_i^\ell = \tau_i^\ell \text{ for all } i \\ \infty & \text{otherwise} \end{cases}$$

Examples of Trace Distances

- Let $\mathbb{K} = L \times \mathbb{R}$. Notation: Trace $\sigma = \big((\sigma_0^\ell, \sigma_0^w), (\sigma_1^\ell, \sigma_1^w), \dots\big)$.

Point-wise trace distance

$$d_T^\bullet(\sigma,\tau) = \begin{cases} \sup_i \lambda^i |\sigma_i^w - \tau_i^w| & \text{if } \sigma_i^\ell = \tau_i^\ell \text{ for all } i \\ \infty & \text{otherwise} \end{cases}$$

Accumulating trace distance

$$d_T^+(\sigma,\tau) = \begin{cases} \sum_i \lambda^i |\sigma_i^w - \tau_i^w| & \text{if } \sigma_i^\ell = \tau_i^\ell \text{ for all } i \\ \infty & \text{otherwise} \end{cases}$$

Maximum-lead trace distance

$$d_T^\pm(\sigma,\tau) = \begin{cases} \sup_i \big|\sum_{j=0}^i \sigma_j^w - \sum_{j=0}^i \tau_j^w\big| & \text{if } \sigma_i^\ell = \tau_i^\ell \text{ for all } i \\ \infty & \text{otherwise} \end{cases}$$

Linear Distance

- (Recall: We assume given a hemimetric $d_T : \mathbb{K}^\omega \times \mathbb{K}^\omega \to [0, \infty]$ on traces.)
- Let $(S, T \subseteq S \times \mathbb{K} \times S)$ be a weighted automaton.
- Linear distance between states $s, t \in S$: use Hausdorff construction:

Definition: Linear distance

$$d_L(s,t) = \sup_{\sigma \in \mathrm{Tr}(s)} \inf_{\tau \in \mathrm{Tr}(t)} d_T(\sigma, \tau)$$

Linear vs. Branching Distance

Definition: Linear distance

$$d_L(s,t) = \sup_{\sigma \in \mathrm{Tr}(s)} \inf_{\tau \in \mathrm{Tr}(t)} d_T(\sigma,\tau)$$

- This is a game!
- Player 1 chooses the worst trace $\sigma \in \mathrm{Tr}(s)$.
- Player 2 matches it with the best trace $\tau \in \mathrm{Tr}(t)$.
- $d_L(s,t)$ = value of the "1-blind weighted simulation game": Player 2 has perfect information, Player 1 is blind.

Definition: Branching distance

$d_B(s,t)$ = value of the same game, but with perfect information

- Hence "$d_B(s,t) = \sup_{s \xrightarrow{\sigma_0} s_1} \inf_{t \xrightarrow{\tau_0} t_1} \sup_{s_1 \xrightarrow{\sigma_1} s_2} \inf_{t_1 \xrightarrow{\tau_1} t_2} \cdots \; d_T(\sigma,\tau)$".

Simulation Games

Precise definition of how this works:

- Given: Weighted automaton $(S, T \subseteq S \times \mathbb{K} \times S)$, states $s, t \in S$
- (Imagine a game of two players taking turns to build two paths:)
- A strategy from s, t: $\theta : \mathrm{fPa}(s) \times \mathrm{fPa}(t) \to T$
  - for Player 1: $\mathrm{start}(\theta(\pi_1, \pi_2)) = \mathrm{end}(\pi_1)$
  - for Player 2: $\mathrm{start}(\theta(\pi_1, \pi_2)) = \mathrm{end}(\pi_2)$
- A round of the game under strategies θ_1, θ_2:
  $\mathrm{Round}_{(\theta_1,\theta_2)}(\pi_1, \pi_2) = \big(\pi_1 \cdot \theta_1(\pi_1, \pi_2), \pi_2 \cdot \theta_2(\pi_1 \cdot \theta_1(\pi_1, \pi_2), \pi_2)\big)$
- The limit of the game under strategies θ_1, θ_2:
  $\lim_{j\to\infty} \mathrm{Round}^j_{(\theta_1,\theta_2)}(s_0, t_0)$ (a pair of infinite paths)
- The utility of the strategies θ_1, θ_2:
  $u(\theta_1, \theta_2) = d_T\big(\mathrm{tr}(\lim_{j\to\infty} \mathrm{Round}^j_{(\theta_1,\theta_2)}(s_0, t_0))\big)$
- The value of the game: $v(s, t) = \sup_{\theta_1} \inf_{\theta_2} u(\theta_1, \theta_2)$

Perfect vs. Imperfect Information

- $\Theta_1(s,t)$, $\Theta_2(s,t)$: sets of all Player-1 resp. Player-2 strategies $\text{fPa}(s) \times \text{fPa}(t) \to T$
- Games with imperfect information: Restrict available strategies to proper subsets of Θ_1 or Θ_2
- Special case: blind Player-1 strategies $\tilde{\Theta}_1 = T^{\text{fPa}(s)}$
- Do not depend on Player-2 choices: Player 1 cannot "see" what Player 2 is doing

- Branching distance: $d_B(s,t) = \sup_{\theta_1 \in \Theta_1(s,t)} \inf_{\theta_2 \in \Theta_2(s,t)} u(\theta_1, \theta_2)$
- Linear distance: $d_L(s,t) = \sup_{\theta_1 \in \tilde{\Theta}_1(s,t)} \inf_{\theta_2 \in \Theta_2(s,t)} u(\theta_1, \theta_2)$

Properties

Proposition

- d_L is a hemimetric.
- If the simulation game is *determined*, d_B is a hemimetric.

- Need determinacy to show triangle inequality
- (But have no counterexample)

Theorem

For all $s, t \in S$, $d_L(s,t) \leq d_B(s,t)$.

Proof:

For d_B, Player 1 (the *sup player*) has more strategies to choose from!

Properties

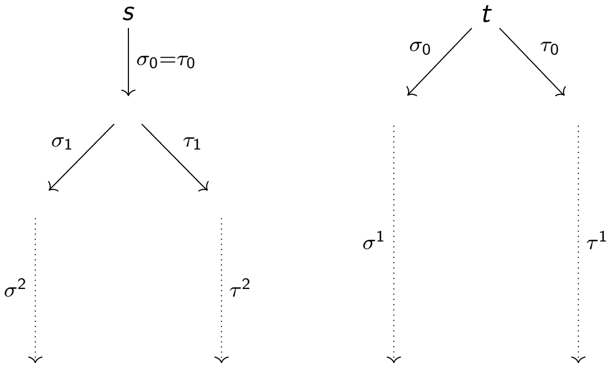
Theorem

There exists a weighted automaton on which d_L and d_B are topologically inequivalent.

- Unless for all traces σ, τ : $\sigma_0 = \tau_0$ implies $d_T(\sigma, \tau) = 0$.
- (*i.e.* d_T measures only on *first* trace element; not very useful!)

Proof

Let $\sigma, \tau \in \mathbb{K}^\omega$ such that $\sigma_0 = \tau_0$, $d_T(\sigma, \tau) > 0$, and $d_T(\tau, \sigma) > 0$.



We have $\mathrm{Tr}(s) = \mathrm{Tr}(t)$, hence $d_L(s,t) = 0$. On the other hand, $d_B(s,t) = \min\big(d_T(\sigma,\tau), d_T(\tau,\sigma)\big) > 0$. That's it.

Wish List

- Relate equivalence of trace distances to equivalence of linear distances. Like this:

Theorem

If trace distances d_T^1 and d_T^2 are Lipschitz equivalent, then the corresponding linear distances d_L^1 and d_L^2 are topologically equivalent.

- Relate equivalence of trace distances to equivalence of branching distances.
- Classify trace distances (up to equivalence).