On Global Measures for the Evaluation of Systems

David de Frutos Escrig
David Romero Hernández

Departamento de Sistemas Informáticos y Computación,
Universidad Complutense de Madrid, Spain

Open Problems in Concurrency Theory - Bertinoro, Italy
June 19^{th}, 2014

Outline

Outline

Visiting Iceland as Motivation

Madrid

Working at UCM

Reykjavik

Visiting Anna and Luca at Reykjavik University ... and more

Visiting Iceland as Motivation

Visiting Iceland as Motivation

Visiting Iceland as Motivation

Visiting Iceland as Motivation

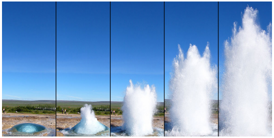


Visiting Iceland as Motivation



Visiting Iceland as Motivation

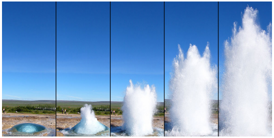


Visiting Iceland as Motivation





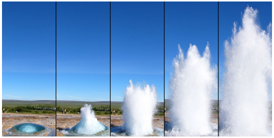



Visiting Iceland as Motivation







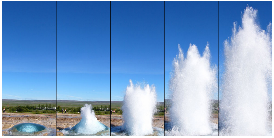



Visiting Iceland as Motivation







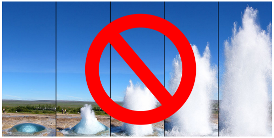




Visiting Iceland as Motivation







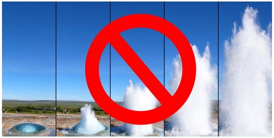




Visiting Iceland as Motivation







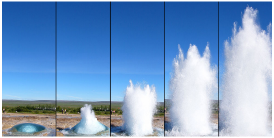




Visiting Iceland as Motivation








Outline

Outline

Previous definitions

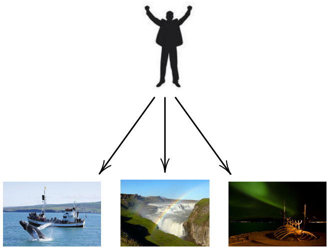
- Distances based on the (bi)simulation game for both branching and linear semantics.[1]
  - $A = (S, T)$ weighted transition system with $s, t \in S$ and $d_T : \mathbb{K}^w \times \mathbb{K}^w \to [0, \infty]$ a trace distance.
  - Using A as a game graph, the simulation game played on A from (s, t) is an infinite two-player game.
  - The strategy space of Player i is denoted by θ_i.
  - The utility function $u : \theta_1 \times \theta_2 \to [0, \infty]$ leads the pay-off of Player 1.
- The value of the game is the optimal Player-1 pay-off

$$v(s,t) = \sup_{\theta_1 \in \Theta_1} \inf_{\theta_2 \in \Theta_2} u(\theta_1, \theta_2)$$

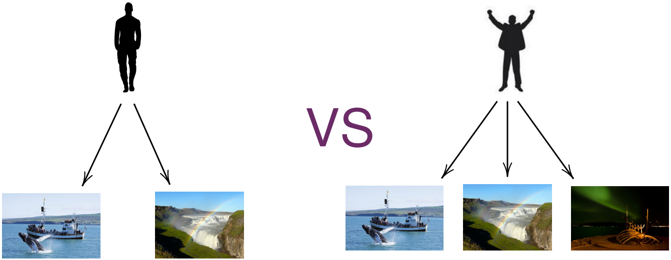
- We use 1-blind strategy $\theta_1 \in \widetilde{\Theta_1}$ for the case of linear semantics.

[1] U. Fahrenberg, A. Legay, and C. R. Thrane. The quantitative linear-time–branching-time spectrum. In S. Chakraborty and A. Kumar, editors, FSTTCS, volume 13 of LIPIcs, pages 103–114. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.

When I visited the beautiful Iceland

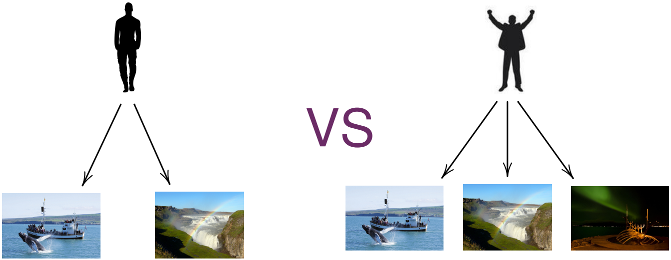


When I visited the beautiful Iceland



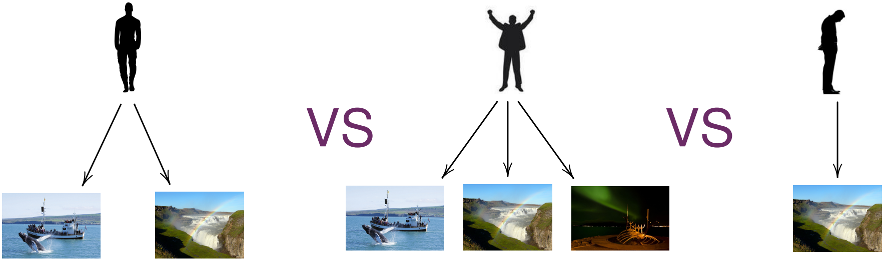
At least I' ve seen whales

When I visited the beautiful Iceland



At least I' ve seen whales

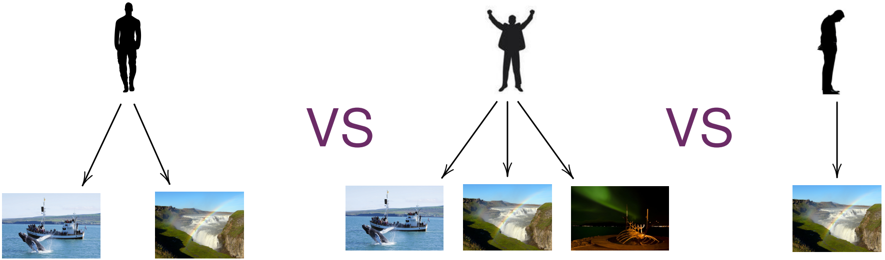
When I visited the beautiful Iceland



At least I' ve seen whales

No whales, No Auroras!!

When I visited the beautiful Iceland



At least I've seen whales

No whales, No Auroras!!

What is the problem?

- Bisimulation characterizes tree equality (up to idempotency).
- The (bi)simulation game works because a single disagreement produces non equality.
- The distance game compares pairs of computations, even if generated following an alternating procedure, and then it looks for the maximal disagreement using the *max* operator.
- We need a more global approach that takes into account how many disagreements there are.

What is the problem?

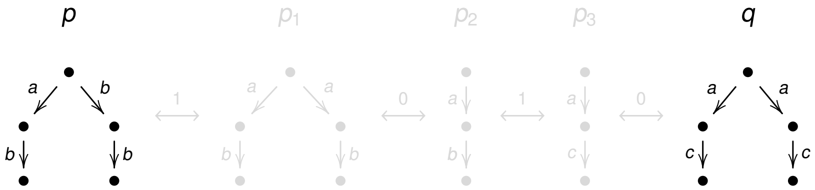
- Bisimulation characterizes tree equality (up to idempotency).
- The (bi)simulation game works because a single disagreement produces non equality.
- The distance game compares pairs of computations, even if generated following an alternating procedure, and then it looks for the maximal disagreement using the *max* operator.
- We need a more global approach that takes into account how many disagreements there are.

Outline

On a global definition of the distance between processes

$$\overline{d} : Act \times Act \to \mathbb{N} \quad \longrightarrow \left\{ \begin{array}{l} \overline{d}(a,b) = 0 \Leftrightarrow a = b \\ \overline{d}(a,c) + \overline{d}(c,b) \leq \overline{d}(a,b) \end{array} \right\} \longrightarrow \quad d_{\overline{d}}(p,q) \leq m$$

An example: We simply consider the discrete distance between actions.

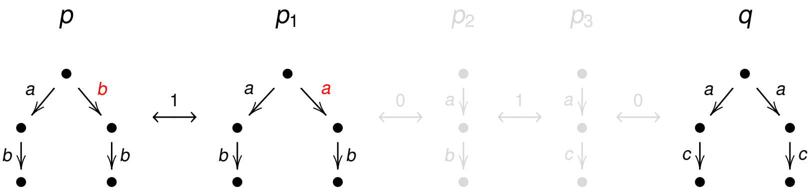


$$d_{\overline{d}}(p,q)\,?$$

On a global definition of the distance between processes

$$\overline{d} : Act \times Act \to \mathbb{N} \quad \longrightarrow \left\{ \begin{array}{l} \overline{d}(a,b) = 0 \Leftrightarrow a = b \\ \overline{d}(a,c) + \overline{d}(c,b) \leq \overline{d}(a,b) \end{array} \right\} \longrightarrow \quad d_{\overline{d}}(p,q) \leq m$$

An example: We simply consider the discrete distance between actions.

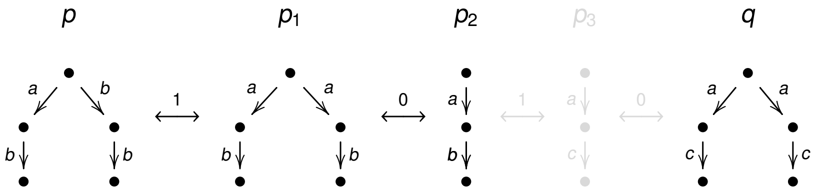


$$\overline{d}(b,a) = 1$$

On a global definition of the distance between processes

$$\overline{d} : Act \times Act \to \mathbb{N} \longrightarrow \left\{ \begin{array}{l} \overline{d}(a,b) = 0 \Leftrightarrow a = b \\ \overline{d}(a,c) + \overline{d}(c,b) \leq \overline{d}(a,b) \end{array} \right\} \longrightarrow d_{\overline{d}}(p,q) \leq m$$

An example: We simply consider the discrete distance between actions.

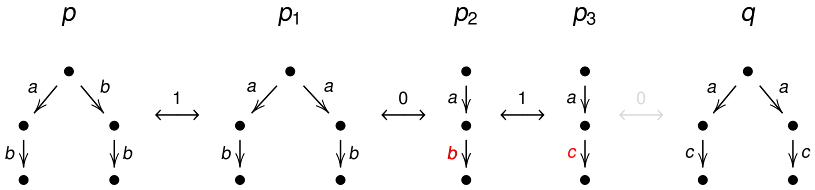


$$d(p_1, p_2) = 0 \text{ using that } p_1 \sim p_2$$

On a global definition of the distance between processes

$$\overline{d} : Act \times Act \to \mathbb{N} \longrightarrow \left\{ \begin{array}{l} \overline{d}(a,b) = 0 \Leftrightarrow a = b \\ \overline{d}(a,c) + \overline{d}(c,b) \leq \overline{d}(a,b) \end{array} \right\} \longrightarrow d_{\overline{d}}(p,q) \leq m$$

An example: We simply consider the discrete distance between actions.

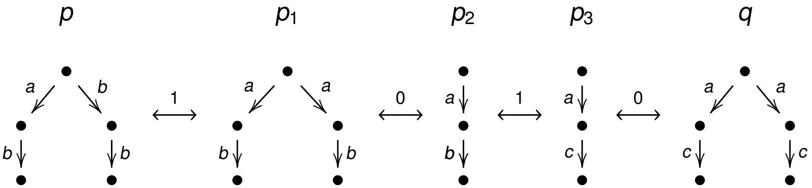


$$\overline{d}(b,c) = 1$$

On a global definition of the distance between processes

$$\overline{d} : Act \times Act \to \mathbb{N} \quad \longrightarrow \left\{ \begin{array}{l} \overline{d}(a,b) = 0 \Leftrightarrow a = b \\ \overline{d}(a,c) + \overline{d}(c,b) \leq \overline{d}(a,b) \end{array} \right\} \longrightarrow \quad d_{\overline{d}}(p,q) \leq m$$

An example: We simply consider the discrete distance between actions.

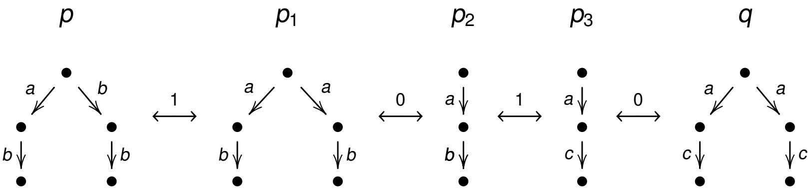


$d(p_3, q) = 0$ using that $p_3 \sim q$

On a global definition of the distance between processes

$$\overline{d} : Act \times Act \to \mathbb{N} \quad \longrightarrow \left\{ \begin{array}{l} \overline{d}(a,b) = 0 \Leftrightarrow a = b \\ \overline{d}(a,c) + \overline{d}(c,b) \le \overline{d}(a,b) \end{array} \right\} \longrightarrow \quad d_{\overline{d}}(p,q) \le m$$

An example: We simply consider the discrete distance between actions.



$$d_{\overline{d}}(p,q) \le 1 + 1 = 2$$

Outline

How it works?

- We could obtain “the” distance between two trees by considering the minimal value d for which we have $d_{\overline{d}}(p, q) \leq d$.
- But, unfortunately, this corresponds to a global study of the set of derivations that produces the bounds.
- It is easy to see that our distance would correspond to the shortest path in the graph whose
  - Nodes correspond to abstract trees (up to bisimulation).
  - Arcs connect trees that only differ in one arc.
  - Their cost is the distance between the two labels of the changed arc.

How it works?

- We consider processes up-to bisimulation.

Definition (Tree distance)

We say that an unordered tree p is at most at distance d from another tree q, w.r.t. the symmetric distance between actions $\overline{d}$, and then we write $d_{\overline{d}}(p,q) \leq d$, if and only if:

- (C1) $p = ap'$, $q = bp'$, and $d \geq \overline{d}(a,b)$, or
- (C2) $p = p' + r$, $q = q' + r$, and $d \geq d_{\overline{d}}(p',q')$, or
- (C3) $p = ap'$, $q = aq'$, and $d \geq d_{\overline{d}}(p',q')$, or
- (C4) There exist r, d' and d'' s.t. $d' \geq d_{\overline{d}}(p,r)$, $d'' \geq d_{\overline{d}}(r,q)$ and $d \geq d' + d''$.

How it works?

- We can present this definition by means of the following rules.

$$(1)\ \frac{}{p\ \ G_n\ \ p} \qquad (2)\ \frac{p\ G_n q}{ap\ \ G_{n+\bar{d}(b,a)}\ \ bq}$$

$$(3)\ \frac{p\ G_n\ p'}{p+q\ \ G_n\ \ p'+q} \qquad (4)\ \frac{p\ G_n\ q \quad q\ G_{n'}\ r}{p\ \ G_{n+n'}\ \ r}$$

PROPOSITION

For all $n \in \mathbb{N}$, we have $p\ G_n\ q$ if and only if $d_{\bar{d}}(p,q) \leq n$.

How it works?

- We can present this definition by means of the following rules.

$$(1)\ \frac{}{p\ \ G_n\ \ p} \qquad (2)\ \frac{p\ G_n q}{ap\ \ G_{n+\bar{d}(b,a)}\ \ bq}$$

$$(3)\ \frac{p\ G_n\ p'}{p+q\ \ G_n\ \ p'+q} \qquad (4)\ \frac{p\ G_n\ q \quad q\ G_{n'}\ r}{p\ \ G_{n+n'}\ \ r}$$

Proposition

For all $n \in \mathbb{N}$, we have $p\ G_n\ q$ if and only if $d_{\bar{d}}(p,q) \leq n$.

Outline

Simulation distance

Definition

Given two processes p and q, we say that the simulation distance from q to p is at most $m \in \mathbb{N}$, w.r.t. the symmetric distance between actions $\overline{d}$, and then we write $d^S_{\overline{d}}(p,q) \leq m$, if we can derive $p\ G^S_m\ q$ applying the following rules:

$$(1)\ \frac{p\ \sqsubseteq_S\ q}{p\ G^S_n\ q} \qquad (2)\ \frac{p\ G^S_n\ q}{ap\ G^S_{n+\overline{d}(b,a)}\ bq}$$

$$(3)\ \frac{p\ G^S_n\ p'}{p+q\ G^S_n\ p'+q} \qquad (4)\ \frac{p\ G^S_n\ q \quad q\ G^S_{n'}\ r}{p\ G^S_{n+n'}\ r}$$

We look for the “minimal changes” that we need to make in q to get a process q' which simulates p.

The rest of the semantics

A (General) Definition

Given a semantics $\mathcal{L}$, defined by a preorder $\sqsubseteq_{\mathcal{L}}$, we say that a process q is at global distance at most $m \in \mathbb{N}$ of being better than some other p, w.r.t. the semantics $\mathcal{L}$ and the distance between actions $\overline{d}$, and then we write $gd_{\overline{d}}^{\mathcal{L}}(p,q) \leq n$, if we can infer $p\ G_n^{\mathcal{L}}\ q$, by applying the following rules:

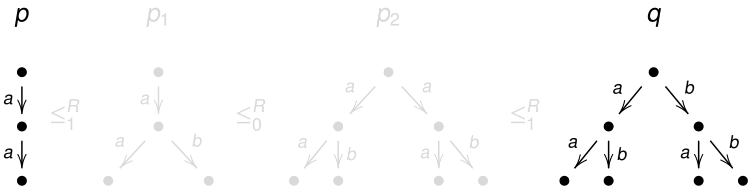
$$(1)\ \frac{p \sqsubseteq_{\mathcal{L}} q}{p\ G_n^{\mathcal{L}}\ q} \qquad (2)\ \frac{p\ G_n^{\mathcal{L}}\ q}{ap\ G_{n+\overline{d}(b,a)}^{\mathcal{L}}\ bq}$$

$$(3)\ \frac{p\ G_n^{\mathcal{L}}\ p'}{p+q\ G_n^{\mathcal{L}}\ p'+q} \qquad (4)\ \frac{p\ G_n^{\mathcal{L}}\ q \quad q\ G_{n'}^{\mathcal{L}}\ r}{p\ G_{n+n'}^{\mathcal{L}}\ r}$$

Proposition

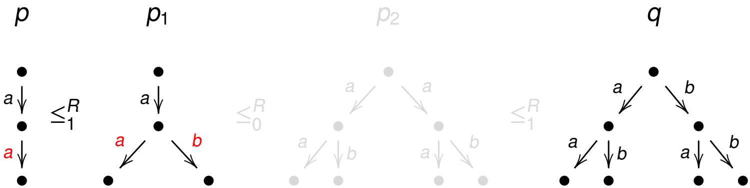
Whenever we have two semantics $\mathcal{L}_1$ and $\mathcal{L}_2$ and the first is finer than the latter ($\sqsubseteq_{\mathcal{L}_1} \subseteq \sqsubseteq_{\mathcal{L}_2}$), then we have $gd_{\overline{d}}^{\mathcal{L}_1}(p,q) \leq n \Rightarrow gd_{\overline{d}}^{\mathcal{L}_2}(p,q) \leq n$, for all processes p, q and any value $n \in \mathbb{N}$.

An interesting example: The case of Readiness



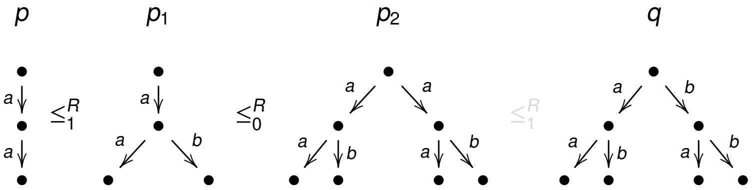
$$d(p, q) \leq ?$$

An interesting example: The case of Readiness



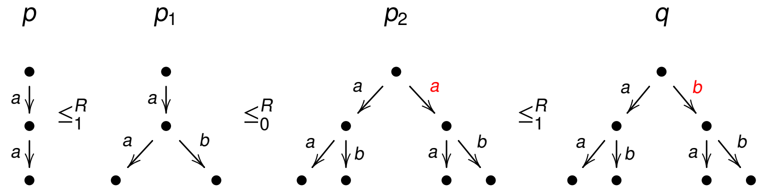
$$\dfrac{a\ G_0\ a \qquad \dfrac{\dfrac{a \sqsubseteq_R a+a}{a\ G_0\ a+a}(1) \quad \dfrac{a\ G_1\ b}{a+a\ G_1\ a+b}(3)}{a\ G_{0+1}\ a+b}(4)}{\dfrac{a\ G_{0+1}\ a+b}{aa\ G_{1+\bar{d}}\ a(a+b)}(2)}(4)$$

An interesting example: The case of Readiness



$$\frac{\dfrac{a\, G_0\, a \qquad \dfrac{\dfrac{a \sqsubseteq_R a+a}{a\, G_0\, a+a}(1) \quad \dfrac{a\, G_1\, b}{a+a\, G_1\, p+b}(3)}{a\, G_{0+1}\, a+b}(4)}{a\, G_{0+1}\, a+b}(4)}{p\, G_{1+\bar{d}}\, a(a+b)}(2) \qquad \frac{a(a+b) \sqsubseteq_R a(a+b)+a(a+b)}{a(a+b)\, G_0\, a(a+b)+a(a+b)}(1)$$

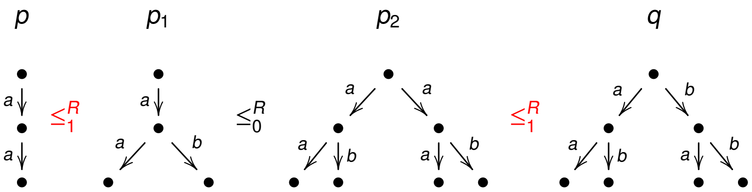
An interesting example: The case of Readiness



$$\dfrac{\dfrac{a\,G_0\,a \qquad \dfrac{\dfrac{a \sqsubseteq_R a+a}{a\,G_0\,a+a}\,(1) \quad \dfrac{a\,G_1\,b}{a+a\,G_1\,p+b}\,(3)}{a\,G_{0+1}\,a+b}\,(4)}{\dfrac{a\,G_{0+1}\,a+b}{p\,G_{1+\overline{d}}\,a(a+b)}\,(2)}\,(4) \qquad \dfrac{a(a+b) \sqsubseteq_R a(a+b)+a(a+b)}{a(a+b)\,G_0\,a(a+b)+a(a+b)}\,(1)}{}$$

$$\dfrac{\dfrac{\dfrac{a+b \sqsubseteq_R a+b}{a+b\,G_0\,a+b}\,(1)}{a(a+b)\,G_{0+\overline{d}(a,b)}\,b(a+b)}\,(2)}{a(a+b)+a(a+b)\,G_1\,q}\,(3)$$

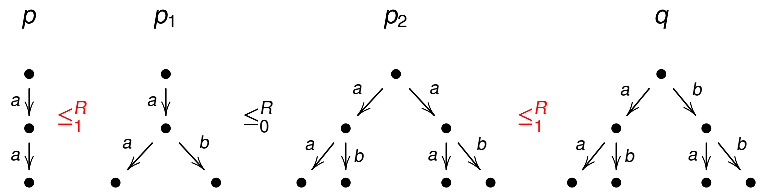
An interesting example: The case of Readiness



$$\dfrac{\dfrac{\dfrac{\dfrac{a\,G_0\,a \qquad \dfrac{\dfrac{a \sqsubseteq_R a+a}{a\,G_0\,a+a}\,(1) \quad \dfrac{a\,G_1\,b}{a+a\,G_1\,p+b}\,(3)}{a\,G_{0+1}\,a+b}\,(4)}{a\,G_{0+1}\,a+b}\,(4)}{p\,G_{1+\bar{d}}\,a(a+b)}\,(2) \qquad \dfrac{a(a+b) \sqsubseteq_R a(a+b)+a(a+b)}{a(a+b)\,G_0\,a(a+b)+a(a+b)}\,(1)}{p\,G_{1+0}\,a(a+b)+a(a+b)}\,(4) \qquad \dfrac{\dfrac{\dfrac{a+b \sqsubseteq_R a+b}{a+b\,G_0\,a+b}\,(1)}{a(a+b)\,G_{0+\bar{d}(a,b)}\,b(a+b)}\,(2)}{a(a+b)+a(a+b)\,G_1\,q}\,(3)}{p\,G_{1+1}\,q}\,(4)$$

$d(p,q) \leq 1+1 = 2$ and $d(p,q) \not\leq 1$

An interesting example: The case of Readiness

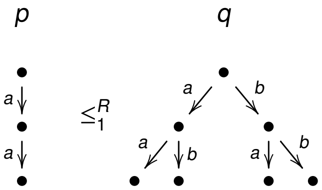


$$\dfrac{\dfrac{\dfrac{a\,G_0\,a \qquad \dfrac{\dfrac{a \sqsubseteq_R a+a}{a\,G_0\,a+a}(1) \quad \dfrac{a\,G_1\,b}{a+a\,G_1\,p+b}(3)}{a\,G_{0+1}\,a+b}(4)}{a\,G_{0+1}\,a+b}(4)}{p\,G_{1+\bar{d}}\,a(a+b)}(2) \qquad \dfrac{a(a+b) \sqsubseteq_R a(a+b)+a(a+b)}{a(a+b)\,G_0\,a(a+b)+a(a+b)}(1)}{\color{red}{p\,G_{1+0}\,a(a+b)+a(a+b)}}\color{red}{(4)} \qquad \dfrac{\dfrac{\dfrac{a+b \sqsubseteq_R a+b}{a+b\,G_0\,a+b}(1)}{a(a+b)\,G_{0+\bar{d}(a,b)}\,b(a+b)}(2)}{a(a+b)+a(a+b)\,G_1\,q}(3)}{\color{red}{p\,G_{1+1}\,q}}\color{red}{(4)}$$

$$d(p,q) \leq 1+1=2 \quad \text{and} \quad d(p,q) \not\leq 1$$

An interesting example: The case of Readiness

However using the classical definition...



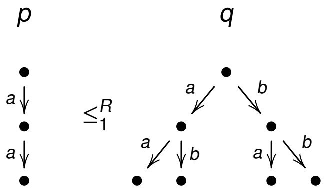
In fact,

$$T(p) = T(q) \Rightarrow \begin{cases} 0 & p \leq_R q \\ 1 & p \not\leq_R q \end{cases}$$

which gives us the same information that the Readiness semantics!!.

An interesting example: The case of Readiness

However using the classical definition...



In fact,

$$T(p) = T(q) \Rightarrow \begin{cases} 0 & p \leq_R q \\ 1 & p \not\leq_R q \end{cases}$$

which gives us the same information that the Readiness semantics!!.

The classic game distances in our frame

A simple variation of the system of rules defining our distances, produces a characterization of the “classical” simulation distance (and also traces distances).

$$(1)\ \frac{p \sqsubseteq_S q}{p\ H_n^S\ q} \qquad (2)\ \frac{p\ H_n^S\ q}{ap\ H^S_{n+\bar{d}(b,a)}\ bq}$$

$$(3')\ \frac{p\ H_n^S\ p' \quad q\ H_{n'}^S\ q'}{p+q\ H^S_{max\{n,n'\}}\ p'+q'} \qquad (4)\ \frac{p\ H_n^S\ q \quad q\ H_{n'}^S\ r}{p\ H^S_{n+n'}\ r}$$

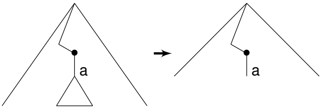
Outline

Outline

Take into account

- We consider the distance between processes based on bisimulation.

- $\pi_k(t)$ is the k-th projection of t, i.e., we preserve those nodes at the k first levels of t.

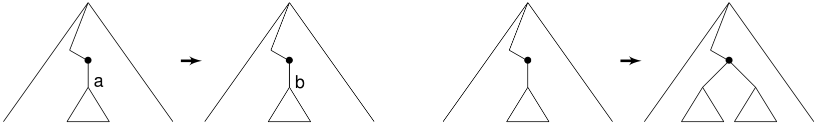


- We restrict ourselves to finitary trees in order to guarantee continuity of bisimilarity:

$$(\forall k \in \mathbb{N}\ \pi_k(t) \sim \pi_k(t')) \Rightarrow t \sim t'.$$

Outline

Our coinductive distance



Definition

Given a domain of actions $(\mathbb{A}, \mathbf{d})$, a discount factor $\alpha \in (0, 1]$ and a family $\mathcal{D} \subseteq \mathit{FyTrees}(\mathbb{A}) \times \mathit{FyTrees}(\mathbb{A}) \times \mathbb{R}^+$, we define the family of relations $\equiv_d^{\mathcal{D},\alpha}$, by:

1. For all $d \geq 0$ we have (i) $(\sum_{j \in J} a_j t_j) + at + at \equiv_d^{\mathcal{D},\alpha} (\sum_{j \in J} a_j t_j) + at$, and (ii) $(\sum_{j \in J} a_j t_j) + at \equiv_d^{\mathcal{D},\alpha} (\sum_{j \in J} a_j t_j) + at + at$.
2. $(\sum_{j \in J} a_j t_j) + at \equiv_{\mathbf{d}(a,b)}^{\mathcal{D},\alpha} (\sum_{j \in J} a_j t_j) + bt$.
3. For all $(t, t', d) \in \mathcal{D}$ we have $(\sum_{j \in J} a_j t_j) + at \equiv_{\alpha d}^{\mathcal{D},\alpha} (\sum_{j \in J} a_j t_j) + at'$.

Our coinductive distance

We introduce the coinductive proof obligations imposed to the families of triples in the previous slide.

Definition

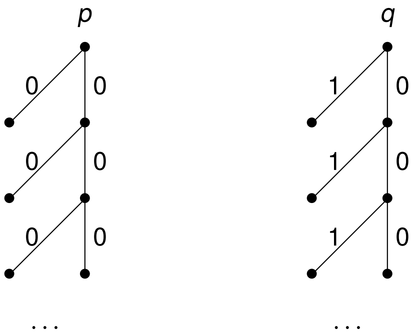
Given a domain of actions $(\mathbb{A}, \mathbf{d})$ and a discount factor $\alpha \in (0, 1]$, we say that a family $\mathcal{D}$ is an α-coinductive collection of distances (α-ccd) between finitary trees, if for all $(t, t', d) \in \mathcal{D}$ there exists a *finite coinductive transformation sequence*

$$\mathcal{C} := t = t^0 \equiv_{d_1}^{\mathcal{D},\alpha} t^1 \equiv_{d_2}^{\mathcal{D},\alpha} \ldots \equiv_{d_n}^{\mathcal{D},\alpha} t^n = t'$$

with $d \geq \sum_{j=1}^{n} d_j$. Then, when there exists an α-ccd $\mathcal{D}$ with $(t, t', d) \in \mathcal{D}$, we will write $t \equiv_d^\alpha t'$, and say that tree t is at most at distance d from tree t' wrt α.

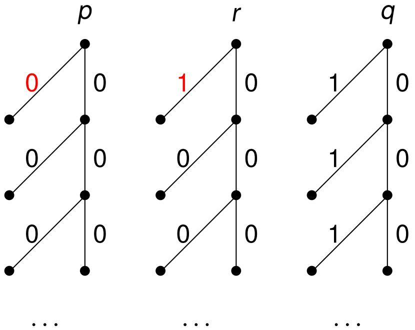
An example

Given $\alpha = 1/2$ for the family $\mathcal{D} = \{(p, q, 2)\}$, we have that $p \equiv_2^{\mathcal{D},1/2} q$.



An example

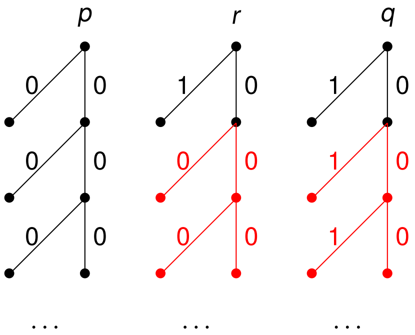
Given $\alpha = 1/2$ for the family $\mathcal{D} = \{(p, q, 2)\}$, we have that $p \equiv_2^{\mathcal{D},1/2} q$.



We apply rule 2 to prove that $(0p') + 0\bullet \equiv_{\mathbf{d}(0,1)}^{\mathcal{D},1/2} (0r') + 1\bullet$

An example

Given $\alpha = 1/2$ for the family $\mathcal{D} = \{(p, q, 2)\}$, we have that $p \equiv_2^{\mathcal{D},1/2} q$.



We apply rule 3 to prove that $(1\bullet) + 0r' \equiv_1^{\mathcal{D},1/2} (1\bullet) + 0q'$

An example

- The finite coinductive sequence

$$\mathcal{C} := p \equiv_1^{\mathcal{D},1/2} r \equiv_1^{\mathcal{D},1/2} q$$

  proves us that $\mathcal{D}$ is indeed an $1/2$-coinductive collection of distance, so that $p \equiv_2^{1/2} q$.

- The coinductive procedure "aggregates" the summands that produce the bound for the distance 2 in a single step.

- We do not need to sum infinite series: coinduction does it for free!.

- Our coinductive definition is equivalent to the operational one for finite trees.

An example

- The finite coinductive sequence

$$\mathcal{C} := p \equiv_1^{\mathcal{D},1/2} r \equiv_1^{\mathcal{D},1/2} q$$

  proves us that $\mathcal{D}$ is indeed an $1/2$-coinductive collection of distance, so that $p \equiv_2^{1/2} q$.

- The coinductive procedure “aggregates” the summands that produce the bound for the distance 2 in a single step.

- We do not need to sum infinite series: coinduction does it for free!.

- Our coinductive definition is equivalent to the operational one for finite trees.

Outline

On the continuity of the global bisimulation distance

PROPOSITION (Sound wrt finite approximation)

For any α-ccd $\mathcal{D}$, the projected family

$$\pi(\mathcal{D}) = \{(\pi_n(t), \pi_n(t'), d) \mid (t, t', d) \in \mathcal{D},\ n \in \mathbb{N}\}$$

is an α-ccd that proves $t \equiv_d^\alpha t' \Rightarrow \forall n \in \mathbb{N}\ \pi_n(t) \equiv_d^\alpha \pi_n(t')$.

PROPOSITION (Completeness wrt finite approximation?)

$t \equiv_d^\alpha t' \Leftarrow \forall n \in \mathbb{N}\ \pi_n(t) \equiv_d^\alpha \pi_n(t')$.

Idea: As far as we have a collection of "uniform" operational sequences $\mathcal{S}^n := \pi_n(t) \rightsquigarrow_{\alpha,d} \pi_n(t')$, we could "overlap" all of them getting an infinite transformation. By "folding" this transformation we obtain the coinductive sequence $\mathcal{C}$, proving $t \equiv_d^\alpha t'$.

Outline

Our algebraic framework

Equational deduction system

1. Monotonicity: $d \leq d_1$ and $p =_d p' \Rightarrow p =_{d_1} p'$
2. Partial reflexivity: $p =_0 p$.
3. Symmetry: $p =_d p' \Rightarrow p' =_d p$.
4. Triangular transitivity: $p =_d p', p' =_{d'} p'' \Rightarrow p =_{d+d'} p''$.
5. Instantiation: $p =_d p' \Rightarrow p + q =_d p' + q \wedge ap =_d ap'$ (plain) [$ap =_{df} ap'$ (discounted)].
6. Prefix axiom: $ap =_d bp \; \forall d \geq d(a, b)$.

Our algebraic framework

Equational deduction system

1. Monotonicity: $d \leq d_1$ and $p =_d p' \Rightarrow p =_{d_1} p'$
2. Partial reflexivity: $p =_0 p$.
3. Symmetry: $p =_d p' \Rightarrow p' =_d p$.
4. Triangular transitivity: $p =_d p', p' =_{d'} p'' \Rightarrow p =_{d+d'} p''$.
5. Instantiation: $p =_d p' \Rightarrow p + q =_d p' + q \wedge ap =_d ap'$ (plain) [$ap =_{df} ap'$ (discounted)].
6. Prefix axiom: $ap =_d bp \; \forall d \geq d(a, b)$.

$$p \equiv p$$

Our algebraic framework

Equational deduction system

1. Monotonicity: $d \leq d_1$ and $p =_d p' \Rightarrow p =_{d_1} p'$
2. Partial reflexivity: $p =_0 p$.
3. Symmetry: $p =_d p' \Rightarrow p' =_d p$.
4. Triangular transitivity: $p =_d p', p' =_{d'} p'' \Rightarrow p =_{d+d'} p''$.
5. Instantiation: $p =_d p' \Rightarrow p + q =_d p' + q \wedge ap =_d ap'$ (plain) [$ap =_{df} ap'$ (discounted)].
6. Prefix axiom: $ap =_d bp \; \forall d \geq d(a, b)$.

$$p \equiv p' \Rightarrow p' \equiv p$$

Our algebraic framework

Equational deduction system

1. Monotonicity: $d \leq d_1$ and $p =_d p' \Rightarrow p =_{d_1} p'$
2. Partial reflexivity: $p =_0 p$.
3. Symmetry: $p =_d p' \Rightarrow p' =_d p$.
4. Triangular transitivity: $p =_d p', p' =_{d'} p'' \Rightarrow p =_{d+d'} p''$.
5. Instantiation: $p =_d p' \Rightarrow p + q =_d p' + q \wedge ap =_d ap'$ (plain) [$ap =_{df} ap'$ (discounted)].
6. Prefix axiom: $ap =_d bp \; \forall d \geq d(a, b)$.

$$p \equiv p', p' \equiv p'' \Rightarrow p \equiv p''$$

Our algebraic framework

Equational deduction system

1. Monotonicity: $d \leq d_1$ and $p =_d p' \Rightarrow p =_{d_1} p'$
2. Partial reflexivity: $p =_0 p$.
3. Symmetry: $p =_d p' \Rightarrow p' =_d p$.
4. Triangular transitivity: $p =_d p', p' =_{d'} p'' \Rightarrow p =_{d+d'} p''$.
5. Instantiation: $p =_d p' \Rightarrow p + q =_d p' + q \wedge ap =_d ap'$ (plain) [$ap =_{df} ap'$ (discounted)].
6. Prefix axiom: $ap =_d bp \; \forall d \geq d(a, b)$.

$$p \equiv p' \Rightarrow p + q \equiv p' + q \wedge ap \equiv ap'$$

Our algebraic framework

Equational deduction system

1. Monotonicity: $d \leq d_1$ and $p =_d p' \Rightarrow p =_{d_1} p'$
2. Partial reflexivity: $p =_0 p$.
3. Symmetry: $p =_d p' \Rightarrow p' =_d p$.
4. Triangular transitivity: $p =_d p', p' =_{d'} p'' \Rightarrow p =_{d+d'} p''$.
5. Instantiation: $p =_d p' \Rightarrow p + q =_d p' + q \wedge ap =_d ap'$ (plain) [$ap =_{df} ap'$ (discounted)].
6. Prefix axiom: $ap =_d bp \; \forall d \geq d(a,b)$.

And also we need...

How it works?

- We have the equational deduction system.
- For each semantics in the spectrum we use the corresponding set of axioms transmuted into $=_0$.
- For instance, consider the case of bisimulation:

$$(B1)\ x + y =_0 y + x \qquad (B2)\ x + x =_0 x$$

$$(B3)\ x + (y + z) =_0 (x + y) + z \qquad (B4)\ z + \mathbf{0} =_0 z$$

How it works?

Theorem

For each semantics $\mathcal{S}$ we have defined an operational distance which measures "how far away" is a process p of being equivalent to q in this semantics.

This operational presentation is equivalent to the algebraic presentation.

Theorem (What about classical definition?)

An algebraic presentation of the classical (bi)simulation distance (also the trace distance) is obtained changing the instantiation rule:

$$p =_d p' \wedge q =_{d'} q' \Rightarrow p + q =_{max\{d,d'\}} p' + q' \wedge ap =_d ap'.$$

Generalization

A (General) Definition

Let $\mathcal{D}$ be an adequate domain for distance values (e.g. $\mathbb{N}$, $\mathbb{R}^+$, $\mathbb{Q}^+$) and Σ a (classic) signature. A $(\mathcal{D}, \Sigma)$-algebra is a Σ-algebra $\langle \mathcal{A}, \Sigma_{\mathcal{A}} \rangle$ and a collection of relations $\langle \equiv_d, d \in \mathcal{D} \rangle$, $\equiv_d \subseteq \mathcal{D} \times \mathcal{D}$, such that:

1. $a \equiv_0 a$ for all $a \in \mathcal{A}$.
2. $a \equiv_d b \Leftrightarrow b \equiv_d a$ for all $a, b \in \mathcal{A}$ and $d \in \mathcal{D}$.
3. $d \leq d'$, $a \equiv_d b \Rightarrow a \equiv_{d'} b$ for all $a, b \in \mathcal{A}$ and $d, d' \in \mathcal{D}$.
4. $(a \equiv_d b \wedge b \equiv_{d'} c) \Rightarrow a \equiv_{d+d'} c$ for all $a, b, c \in \mathcal{A}$ and $d, d' \in \mathcal{D}$.
5. $f \in \Sigma$, $ar(f) = k$, $a_i \equiv_{d_i} b_i$ for all $i \in 1..k \Rightarrow f(\overline{a}) \equiv_{\varphi_f(\overline{d})} f(\overline{b})$.

Outline

Some extensions

- When comparing p and q the distances presented reflect how many changes we need to make in q in order to get a process that really simulates p.
- It could be the case that q, already offers some actions that would perefctly do the work without needing any change.

Definition

An asymmetric quasi-distance in a set of actions Act is a function $d : Act \times Act \to \mathbb{N}$ which satisfies $d(a,a) = 0\ \forall a \in Act$, and the triangular inequality $d(a,b) + d(b,c) \geq d(a,c)\ \forall a,b,c \in Act$. We will say that $d(a,b)$ expresses "how far away" is action a of covering the expectations to have a b.

- Unexpected termination could be solved by adding a fixed payment f, taking $d(p,\mathbf{0}) \leq f$ and $d(\mathbf{0},p) \leq f$, $\forall p \neq \mathbf{0}$. Or also aggregating the payments needed to cover any action in p.

Conclusions and Future work

- We have presented the distance for all the semantics in an operational, denotational (FORTE 2012), algebraic way (WADT 2012), and recently in FORTE 2014 we have studied the distances between infinite (but finitary!) processes.
- Near future
  - We are (hopefuly!) close to conclude the proof of continuity.
  - We are working in a definition of *approximated testing*.
- Further future
  - The plain distance between trees, no considering idempotency, captures redundancy. We could use these ideas to define *approximated fault tolerance*.
  - What happens when we allow negative values at the distances? Amortized simulation.

THANKS!