A Game Characterization for Contrasimilarity

Benjamin Bisping Luisa Montanari

Technische Universität Berlin
Modelle und Theorie Verteilter Systeme

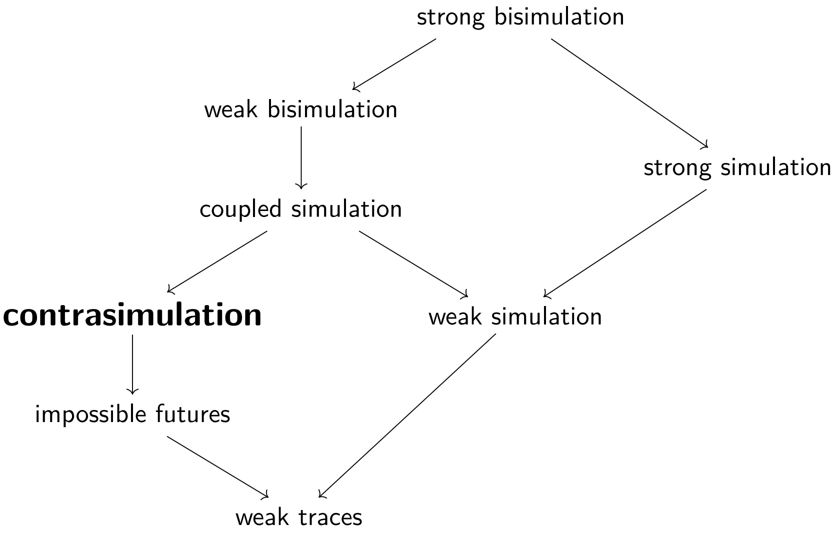
Workshop EXPRESS/SOS, August 2021

Introduction to Contrasimilarity

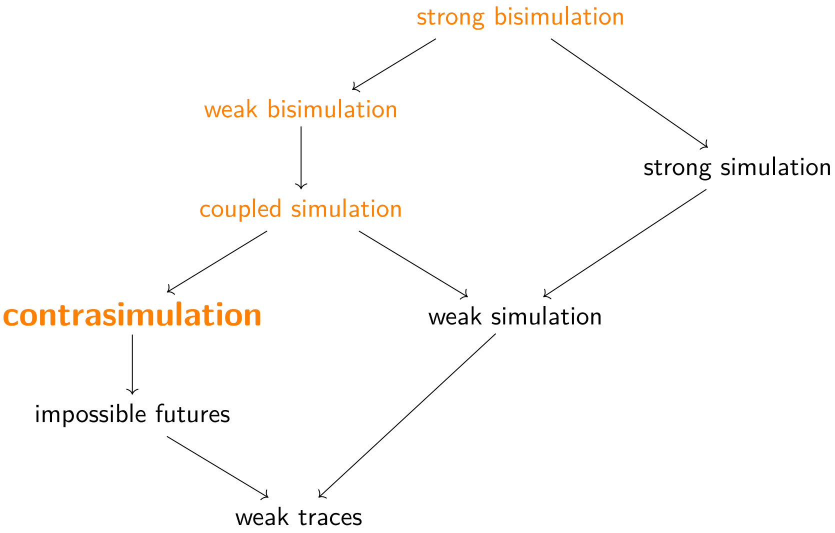
Contrasimilarity is a

- behavioral equivalence
- for systems with internal steps

Linear Time – Branching Time Spectrum



Linear Time – Branching Time Spectrum



Introduction

Contrasimilarity:

- behavioral equivalence
- internal behavior
- weakest abstraction of bisimulation

Our main contributions:

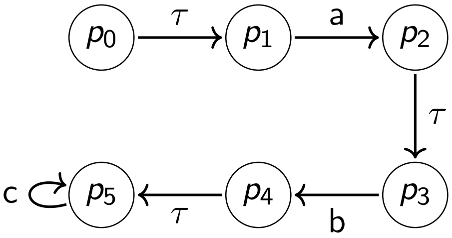
1. present the first game characterization of the contrasimulation preorder
2. prove its correctness with Isabelle/HOL

Preliminaries — Transition Systems

Definition (Labeled Transition System, LTS)

An LTS $(S, Act_\tau, \rightarrow)$ consists of

- a set of states S,
- a set $Act_\tau = Act \cup \{\tau\}$ of
  - visible actions Act and
  - an internal action τ, and
- a transition relation $\rightarrow : S \times Act_\tau \times S$.

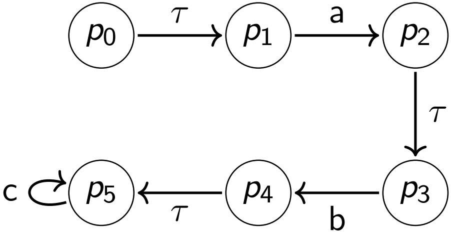


Preliminaries — Transition Systems

Definition (Transitions)

Let $\alpha \in Act_\tau, \vec{w} \in Act^*$. Write

- $p \xrightarrow{\alpha} p'$ iff $(p, \alpha, p') \in \rightarrow$,



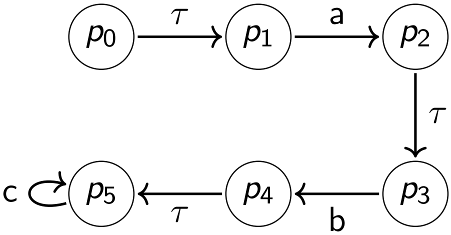
$p_1 \xrightarrow{a} p_2$

Preliminaries — Transition Systems

Definition (Transitions)

Let $\alpha \in Act_\tau, \vec{w} \in Act^*$. Write

- $p \xrightarrow{\alpha} p'$ iff $(p, \alpha, p') \in \rightarrow$,
- $p \Rightarrow p'$ iff $p \xrightarrow{\tau}{}^* p'$,



$p_0 \Rightarrow p_0$

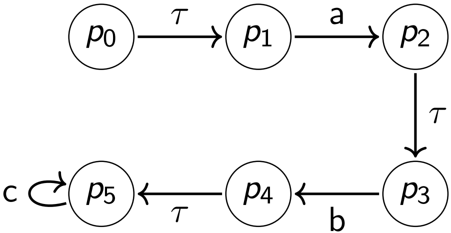
$p_0 \Rightarrow p_1$

Preliminaries — Transition Systems

Definition (Transitions)

Let $\alpha \in Act_\tau, \vec{w} \in Act^*$. Write

- $p \xrightarrow{\alpha} p'$ iff $(p, \alpha, p') \in \rightarrow$,
- $p \Rightarrow p'$ iff $p \xrightarrow{\tau}^* p'$,
- $p \overset{\hat{\alpha}}{\Longrightarrow} p'$ iff $p \Rightarrow \xrightarrow{\alpha} \Rightarrow p'$ for $\alpha \neq \tau$,



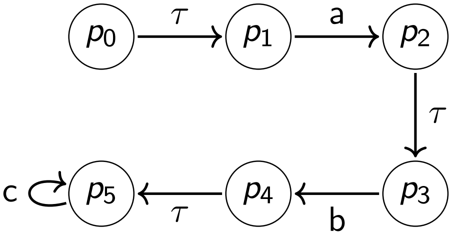
$p_0 \overset{\hat{a}}{\Longrightarrow} p_2$
$p_0 \overset{\hat{a}}{\Longrightarrow} p_3$

Preliminaries — Transition Systems

Definition (Transitions)

Let $\alpha \in Act_\tau, \vec{w} \in Act^*$. Write

- $p \xrightarrow{\alpha} p'$ iff $(p, \alpha, p') \in \rightarrow$,
- $p \Rightarrow p'$ iff $p \xrightarrow{\tau}{}^* p'$,
- $p \xRightarrow{\hat{\alpha}} p'$ iff $p \Rightarrow\xrightarrow{\alpha}\Rightarrow p'$ for $\alpha \neq \tau$,
- $p \xRightarrow{\vec{w}} p'$ iff

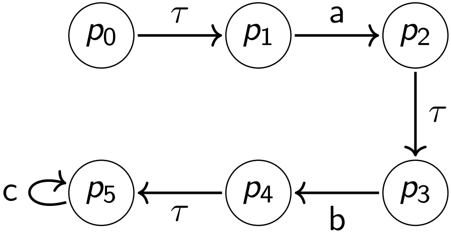


Preliminaries — Transition Systems

Definition (Transitions)

Let $\alpha \in Act_\tau, \vec{w} \in Act^*$. Write

- $p \xrightarrow{\alpha} p'$ iff $(p, \alpha, p') \in \rightarrow$,
- $p \Rightarrow p'$ iff $p \xrightarrow{\tau}^* p'$,
- $p \xRightarrow{\hat{\alpha}} p'$ iff $p \Rightarrow \xrightarrow{\alpha} \Rightarrow p'$ for $\alpha \neq \tau$,
- $p \xRightarrow{\vec{w}} p'$ iff
  - $\vec{w} = \varepsilon$ and $p \Rightarrow p'$ or



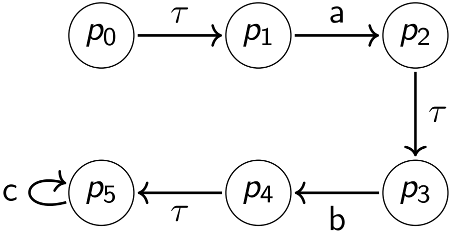
$p_0 \xRightarrow{\varepsilon} p_0$

Preliminaries — Transition Systems

Definition (Transitions)

Let $\alpha \in Act_\tau, \vec{w} \in Act^*$. Write

- $p \xrightarrow{\alpha} p'$ iff $(p, \alpha, p') \in \rightarrow$,
- $p \Rightarrow p'$ iff $p \xrightarrow{\tau}^* p'$,
- $p \xRightarrow{\hat{\alpha}} p'$ iff $p \Rightarrow\xrightarrow{\alpha}\Rightarrow p'$ for $\alpha \neq \tau$,
- $p \xRightarrow{\vec{w}} p'$ iff
  - $\vec{w} = \varepsilon$ and $p \Rightarrow p'$ or
  - $\vec{w} = w_0 w_1 \dots w_n$ and $p \xRightarrow{\hat{w}_0}\xRightarrow{\hat{w}_1} \dots \xRightarrow{\hat{w}_n} p'$.



$$p_0 \xRightarrow{abc} p_5$$

Contrasimulation

Definition (Contrasimulation)

A contrasimulation is a relation R where, for all $(p, q) \in R$ with $\vec{w} \in Act^*$ and $p \stackrel{\vec{w}}{\Longrightarrow} p'$, there is a q' with $q \stackrel{\vec{w}}{\Longrightarrow} q'$ and $(q', p') \in R$.

Contrasimulation

Definition (Contrasimulation)

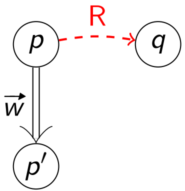
A contrasimulation is a relation R where, for all $(p, q) \in R$ with $\vec{w} \in Act^*$ and $p \stackrel{\vec{w}}{\Longrightarrow} p'$, there is a q' with $q \stackrel{\vec{w}}{\Longrightarrow} q'$ and $(q', p') \in R$.

$$\begin{tikzcd} p \arrow[r, dashed, red, "R"] & q \end{tikzcd}$$

Contrasimulation

Definition (Contrasimulation)

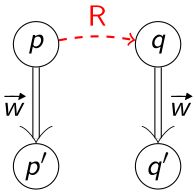
A contrasimulation is a relation R where, for all $(p, q) \in R$ with $\vec{w} \in Act^*$ and $p \overset{\vec{w}}{\Longrightarrow} p'$, there is a q' with $q \overset{\vec{w}}{\Longrightarrow} q'$ and $(q', p') \in R$.



Contrasimulation

Definition (Contrasimulation)

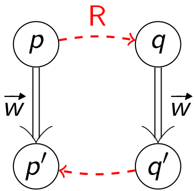
A contrasimulation is a relation R where, for all $(p, q) \in R$ with $\vec{w} \in Act^*$ and $p \stackrel{\vec{w}}{\Longrightarrow} p'$, there is a q' with $q \stackrel{\vec{w}}{\Longrightarrow} q'$ and $(q', p') \in R$.



Contrasimulation

Definition (Contrasimulation)

A contrasimulation is a relation R where, for all $(p, q) \in R$ with $\vec{w} \in Act^*$ and $p \overset{\vec{w}}{\Longrightarrow} p'$, there is a q' with $q \overset{\vec{w}}{\Longrightarrow} q'$ and $(q', p') \in R$.



Contrasimulation

Definition (Contrasimulation)

A contrasimulation is a relation R where, for all $(p, q) \in R$ with $\vec{w} \in Act^*$ and $p \overset{\vec{w}}{\Longrightarrow} p'$, there is a q' with $q \overset{\vec{w}}{\Longrightarrow} q'$ and $(q', p') \in R$.

$$\begin{tikzcd} p \arrow[r, dashed, "R"] \arrow[d, Rightarrow, "\vec{w}"'] & q \arrow[d, Rightarrow, "\vec{w}"] \\ p' & q' \arrow[l, dashed] \end{tikzcd}$$

Write $p \preceq_C q$ if
there is a contrasimulation R with $(p, q) \in R$.

Game characterizations

Behavioral equivalences can be characterized with games!

Two opposing players:

- attacker wants to **disprove** $p \preceq_C q$,
- defender wants to **maintain** $p \preceq_C q$.

Preliminaries — Games

Definition (Games)

A game $\mathcal{G}[g_0] = (G, G_d, \rightarrowtail, g_0)$ consists of

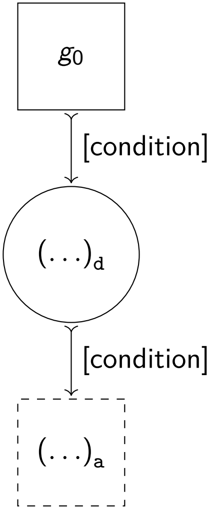
- game positions G, partitioned into
  - defender positions $G_d \subseteq G$
  - attacker positions $G_a := G \setminus G_d$,
- game moves $\rightarrowtail \subseteq G \times G$, and
- an initial position $g_0 \in G$.

Preliminaries — Games

Definition (Games)

A game $\mathcal{G}[g_0] = (G, G_d, \rightarrowtail, g_0)$ consists of

- game positions G, partitioned into
  - defender positions $G_d \subseteq G$
  - attacker positions $G_a := G \setminus G_d$,
- game moves $\rightarrowtail \subseteq G \times G$, and
- an initial position $g_0 \in G$.



Preliminaries — Games

Definition (Plays)

Plays are (in)finite paths $g_0 g_1 \ldots \in G^\infty$ with $g_i \rightarrowtail g_{i+1}$ of $\mathcal{G}[g_0]$.

Definition (Play wins)

1. The defender wins infinite plays.
2. If a finite play $g_0 \ldots g_n \not\rightarrowtail$ is stuck, the stuck player loses.

Preliminaries — Games

Definition (Defender strategies)

A defender strategy f is a mapping from initial play fragments to next moves:
$f \subseteq \{(g_0 \dots g_n, g_{n+1}) \mid g_n \in G_d \wedge g_n \rightarrowtail g_{n+1}\}$.

Definition (Strategy consistency)

A play g is **consistent** with a defender strategy f iff, for each move $g_i \rightarrowtail g_{i+1}$ with $g_i \in G_d$, we have $g_{i+1} = f(g_0 \dots g_i)$.

Definition (Game wins)

The defender **wins** game $\mathcal{G}[g_0]$ iff they win all plays consistent with a strategy f.

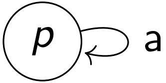
Attempt 1:
The Basic Contrasimulation Game

The Basic Contrasimulation Game

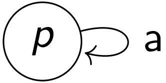
$$\begin{tikzcd} p \arrow[r, dashed, red, "R"] \arrow[d, Rightarrow, "\vec{w}"'] & q \arrow[d, Rightarrow, "\vec{w}"] \\ \forall \; p' & q' \; \exists \arrow[l, dashed, red] \end{tikzcd}$$

$$(p, q)_{\mathtt{a}} \xrightarrow{[p \xRightarrow{\vec{w}} p']} (\vec{w}, p', q)_{\mathtt{d}} \xrightarrow{[q \xRightarrow{\vec{w}} q']} (q', p')_{\mathtt{a}}$$

Problems of the Basic Game



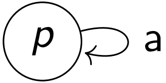
Problems of the Basic Game



Induces infinitely many words:

- ε
- a
- aa
- aaa
- ...

Problems of the Basic Game



Induces infinitely many words:

- ε
- a
- aa
- aaa
- ...

$\Longrightarrow$ **Infinitely many game positions!**

How to eliminate words from the game moves?

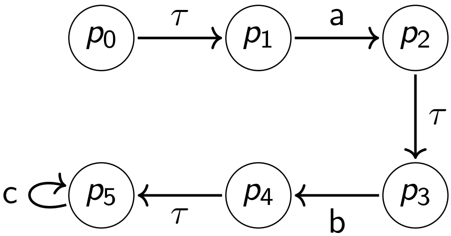
Attempt 2:
The Contrasimulation Set Game

Preliminaries — Transition Systems

Definition (Transitions)

Let $\alpha \in Act_\tau, \vec{w} \in Act^*$. Write

- $p \xrightarrow{\alpha} p'$ iff $(p, \alpha, p') \in \rightarrow$,
- $p \Rightarrow p'$ iff $p \xrightarrow{\tau}^* p'$,
- $p \xRightarrow{\hat{\alpha}} p'$ iff $p \Rightarrow \xrightarrow{\alpha} \Rightarrow p'$ for $\alpha \neq \tau$,
- $p \xRightarrow{\vec{w}} p'$ iff
  - $\vec{w} = \varepsilon$ and $p \Rightarrow p'$ or
  - $\vec{w} = w_0 w_1 \dots w_n$ and $p \xRightarrow{\hat{w}_0} \xRightarrow{\hat{w}_1} \dots \xRightarrow{\hat{w}_n} p'$.
- **NEW:** $\mathbf{p} \overset{\alpha}{\Longrightarrow\!\!\triangleright} \mathbf{p}'$ **iff** $\mathbf{p} \Rightarrow \xrightarrow{\alpha} \mathbf{p}'$.



$p_0 \overset{a}{\Longrightarrow\!\!\triangleright} p_2$

$p_0 \not\overset{a}{\Longrightarrow\!\!\triangleright} p_3$

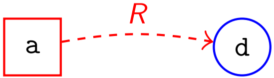
Contrasimulation Set Game

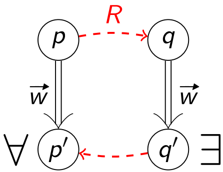
Idea:

- Split words $\vec{w}$ into single actions $w_0, w_1 \ldots$.
- Split attacker word challenge into:
  - `Simulation` phase and
  - `Swap` request
- Let defender move over **sets of states**

Set Game — Intuition

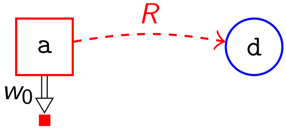
Attacker wants to challenge $\vec{w} = w_0 w_1 \dots w_n$.

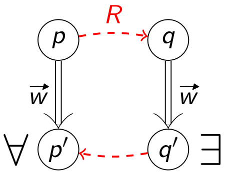




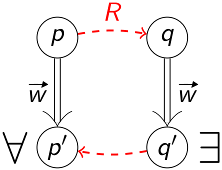
Set Game — Intuition

Attacker wants to challenge $\vec{w} = w_0 w_1 \dots w_n$.

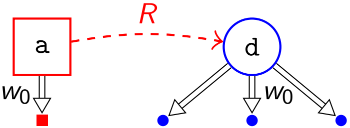




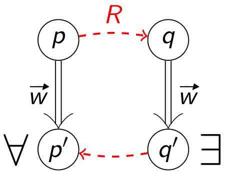
Set Game — Intuition



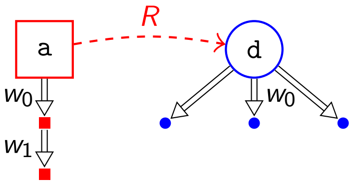
Attacker wants to challenge $\vec{w} = w_0 w_1 \dots w_n$.



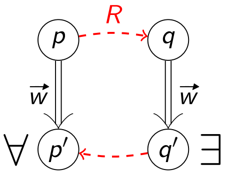
Set Game — Intuition



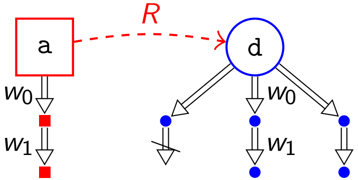
Attacker wants to challenge $\vec{w} = w_0 w_1 \dots w_n$.



Set Game — Intuition

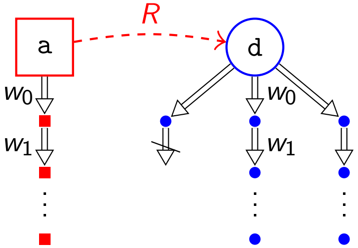


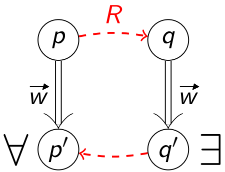
Attacker wants to challenge $\vec{w} = w_0 w_1 \dots w_n$.



Set Game — Intuition

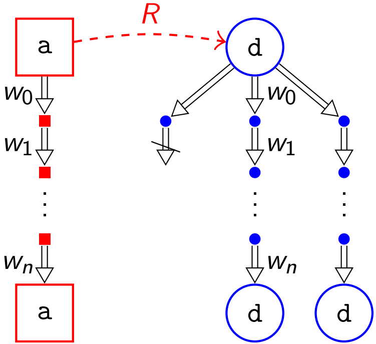
Attacker wants to challenge $\vec{w} = w_0 w_1 \dots w_n$.

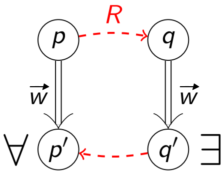




Set Game — Intuition

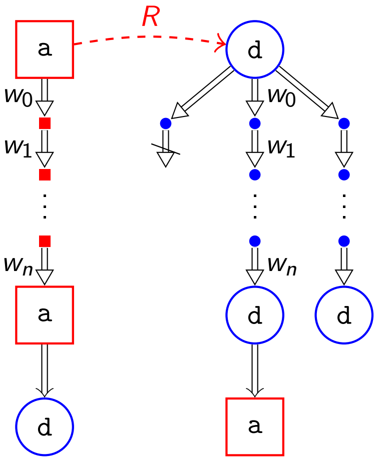
Attacker wants to challenge $\vec{w} = w_0 w_1 \dots w_n$.

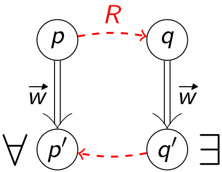




Set Game — Intuition

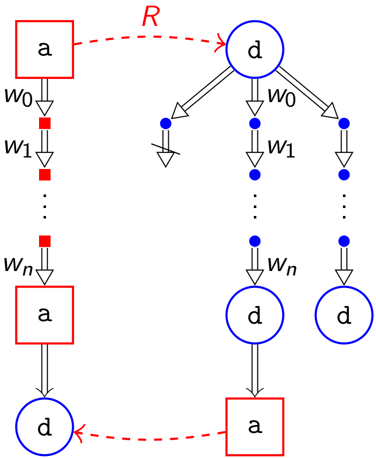
Attacker wants to challenge $\vec{w} = w_0 w_1 \dots w_n$.

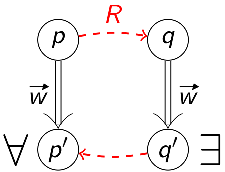




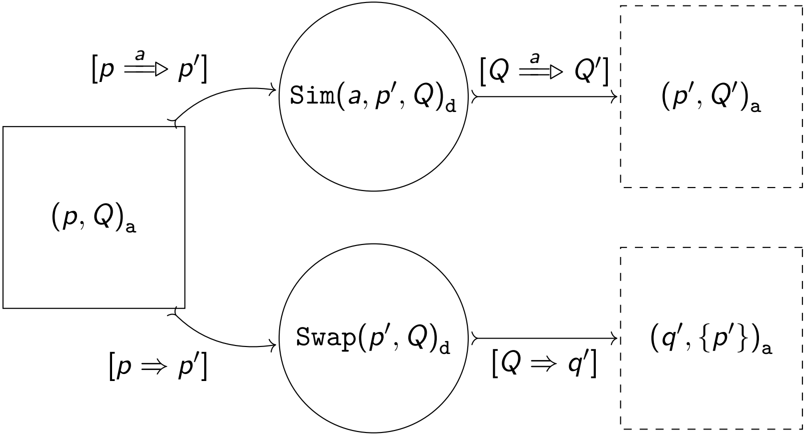
Set Game — Intuition

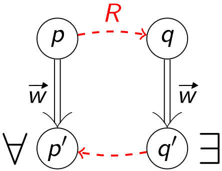
Attacker wants to challenge $\vec{w} = w_0 w_1 \dots w_n$.





Set Game — Model





Correctness

Theorem

The defender wins $\mathcal{G}_C[(p, \{q\})_{\mathtt{a}}]$ *if and only if* $p \preceq_C q$.

Soundness — Proof Sketch

Lemma

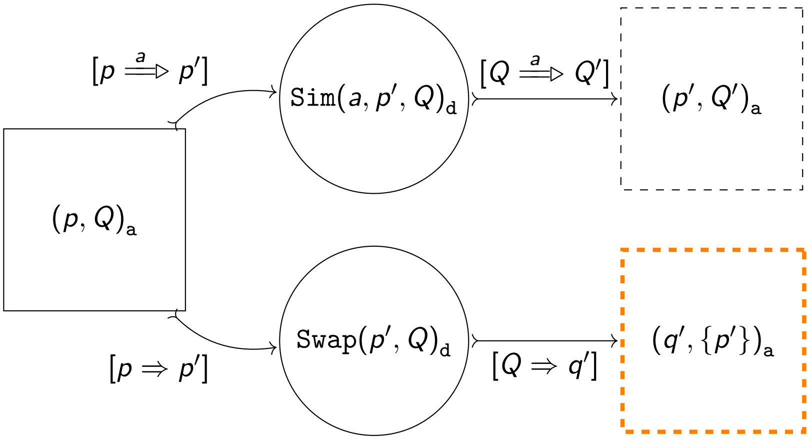
If the defender wins $\mathcal{G}_C[(p, \{q\})_\mathtt{a}]$, then $p \preceq_C q$.

Proof.

1. Construct a relation R containing
   - the initial states (p, q) and
   - states of all possible $(\ldots)_\mathtt{a}$-positions following a $\mathtt{Swap}(\ldots)_\mathtt{d}$-position.

$\square$

Soundness — Proof Sketch



Soundness — Proof Sketch

Lemma

If the defender wins $\mathcal{G}_C[(p, \{q\})_{\mathtt{a}}]$, then $p \preceq_C q$.

Proof.

1. Construct a relation R containing
   - the initial states (p, q) and
   - states of all possible $(\ldots)_{\mathtt{a}}$-positions following a $\mathtt{Swap}(\ldots)_{\mathtt{d}}$-position.
2. Prove that R is a contrasimulation.

□

Completeness — Proof Sketch

Lemma

If $p \preceq_C q$, then the defender wins $\mathcal{G}_C[(p, \{q\})_\mathtt{a}]$.

Proof.

1. Construct a defender strategy f_C from $\preceq_C$.
2. $\preceq_C$ always ensures existence of a next move for the defender.
3. The defender is never stuck using f_C.
4. The defender wins every play using f_C.

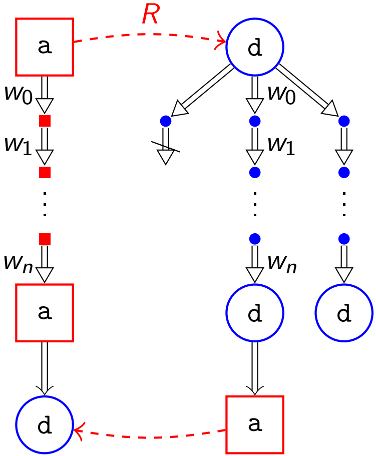
□

Why Sets?

Subset construction induces an **exponential** game size!

Why Sets?

Subset construction induces an **exponential** game size!



Conclusion

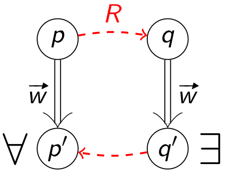
Contributions:

- an overview of the bisimulation-like properties of contrasimilarity
- a game for the contrasimulation preorder
- a proof of its correctness in Isabelle/HOL

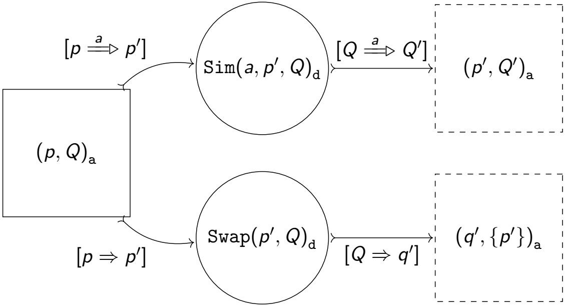
Perspective:

- use of the game in contrasimulation equivalence checking
- further use of Isabelle theory in verification contexts
    - available at `https://github.com/luisamontanari/ContrasimGame`

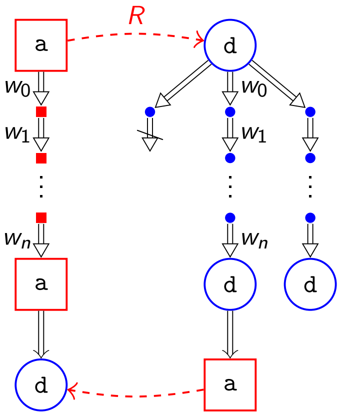
Thank you for your attention!



Schematic Definition Contrasimulation

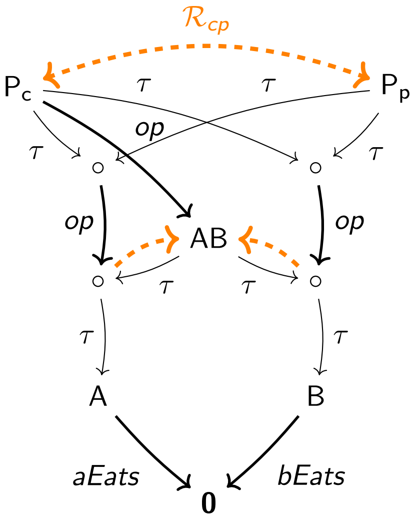


Schematic Model Set Game

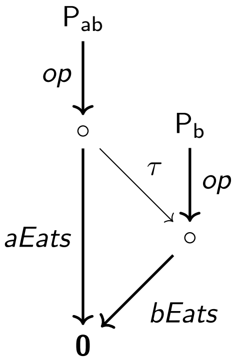


Relating the Set Game
to the Contrasimulation

A contrasimilar process



Fooling One-Step Contrasimulation



Additional Equalities

Contrasimilarity satisfies

- all laws satisfied by weak bisimulation
- **CS** : $\tau.(\tau.X + Y) = \tau.X + Y$ (shared with coupled similarity)
- **C** : $a.(\tau.X + \tau.Y) = a.X + a.Y$.