Approximating Weak Bisimilarity of Basic Parallel Processes

Patrick Totzke[1] Piotrek Hofman[2]

[1]LFCS, University of Edinburgh

[2]University of Warsaw

September 7, 2012

Basic Parallel Processes (BPP)

Rewriting Rules

$$\{Q \xrightarrow{a} QQ,\ Q \xrightarrow{b} \varepsilon,\ Q \xrightarrow{c} Q\}$$

Basic Parallel Processes (BPP)

Rewriting Rules

$$\{Q \xrightarrow{a} QQ,\ Q \xrightarrow{b} \varepsilon,\ Q \xrightarrow{c} Q\}$$

Transition System

$$\begin{tikzcd}
\varepsilon & Q \arrow[l, "b"] \arrow[r, shift left, "a"] \arrow[loop above, "c"] & Q^2 \arrow[l, shift left, "b"] \arrow[r, shift left, "a"] \arrow[loop above, "c"] & Q^3 \arrow[l, shift left, "b"] \arrow[r, shift left, "a"] \arrow[loop above, "c"] & \cdots \arrow[l, shift left, "b"]
\end{tikzcd}$$

Basic Parallel Processes (BPP)

Rewriting Rules

$$\{Q \xrightarrow{a} QQ,\ Q \xrightarrow{b} \varepsilon,\ Q \xrightarrow{c} Q\}$$

Notation

$$\begin{tikzcd} \varepsilon & Q \arrow[l, "b"'] \arrow[loop above, "c"] \looparrowleft aQ \end{tikzcd}$$

Transition System

$$\begin{tikzcd} \varepsilon & Q \arrow[l, "b"] \arrow[loop above, "c"] \arrow[r, shift left, "a"] & Q^2 \arrow[l, shift left, "b"] \arrow[loop above, "c"] \arrow[r, shift left, "a"] & Q^3 \arrow[l, shift left, "b"] \arrow[loop above, "c"] \arrow[r, shift left, "a"] & \cdots \arrow[l, shift left, "b"] \end{tikzcd}$$

Basic Parallel Processes (BPP)

Rewriting Rules

Notation

$$\{Q \xrightarrow{a} QQ,\ Q \xrightarrow{b} \varepsilon,\ Q \xrightarrow{c} Q\}$$

$$\begin{tikzcd} \varepsilon & Q \arrow[l, "b"'] \arrow[loop above, "c"] \arrow[r, "\supset"] & aQ \end{tikzcd}$$

Transition System

$$\begin{tikzcd} \varepsilon & Q \arrow[l, "b"] \arrow[loop above, "c"] \arrow[r, shift left, "a"] & Q^2 \arrow[l, shift left, "b"] \arrow[loop above, "c"] \arrow[r, shift left, "a"] & Q^3 \arrow[l, shift left, "b"] \arrow[loop above, "c"] \arrow[r, shift left, "a"] & \cdots \arrow[l, shift left, "b"] \end{tikzcd}$$

Bisimulation Games

…are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

Bisimulation Games

...are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

In each round

α *vs.* β

1. Spoiler moves $\xrightarrow{a}$ from one process
2. Duplicator responds $\xrightarrow{a}$ from the other
3. the game continues from α' vs. β'.

Bisimulation Games

...are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

In each round

$$\begin{tikzcd} \alpha \arrow[d, "a"'] & \textit{vs.} & \beta \\ \alpha' & & \end{tikzcd}$$

1. Spoiler moves $\xrightarrow{a}$ from one process
2. Duplicator responds $\xrightarrow{a}$ from the other
3. the game continues from α' vs. β'.

Bisimulation Games

. . . are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

In each round

$$\begin{tikzcd} \alpha \arrow[d, "a"'] & \textit{vs.} & \beta \arrow[d, red, "\color{red}a"'] \\ \alpha' & & \beta' \end{tikzcd}$$

1. Spoiler moves $\xrightarrow{a}$ from one process
2. Duplicator responds $\xrightarrow{a}$ from the other
3. the game continues from α' vs. β'.

Bisimulation Games

. . . are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

In each round

$$\begin{array}{ccc} \alpha & vs. & \beta \\ \Big\downarrow a & & \Big\downarrow a \\ \alpha' & vs. & \beta' \end{array}$$

1. Spoiler moves $\xrightarrow{a}$ from one process
2. Duplicator responds $\xrightarrow{a}$ from the other
3. the game continues from α' vs. β'.

Bisimulation Games

…are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

In each round

$$\begin{tikzcd} \alpha \arrow[d, "a"'] & \textit{vs.} & \beta \arrow[d, "a"'] \\ \alpha' & \textit{vs.} & \beta' \end{tikzcd}$$

1. Spoiler moves $\xrightarrow{a}$ from one process
2. Duplicator responds $\xrightarrow{a}$ from the other
3. the game continues from α' vs. β'.

Def: Bisimulation ($\sim$)

$\alpha \sim \beta$ iff Duplicator has a strategy to win the game from α vs. β.

Silent and Weak Steps

silent weak step

$$\xRightarrow{\tau} \;:=\; \xrightarrow{\tau}{}^{*}$$

visible weak step ($a \neq \tau$)

$$\xRightarrow{a} \;:=\; \xrightarrow{\tau}{}^{*} \xrightarrow{a} \xrightarrow{\tau}{}^{*}$$

Bisimulation Games

...are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

In each round

$$\begin{array}{ccc} \alpha & vs. & \beta \\ \Big\downarrow a & & \Big\downarrow a \\ \alpha' & vs. & \beta' \end{array}$$

1. Spoiler moves $\xrightarrow{a}$ from one process
2. Duplicator responds $\xrightarrow{a}$ from the other
3. the game continues from α' vs. β'.

Bisimulation $\sim$

$\alpha \sim \beta$ iff Duplicator has a strategy to win the game from α vs. β.

Weak Bisimulation Games

…are played in rounds between Spoiler and Duplicator. If a player cannot move the other wins. Infinite plays are won by Duplicator.

In each round

$$\begin{tikzcd} \alpha \arrow[d, "a"'] & vs. & \beta \arrow[d, "a"'] \\ \alpha' & vs. & \beta' \end{tikzcd}$$

1. Spoiler moves $\stackrel{a}{\Longrightarrow}$ from one process
2. Duplicator responds $\stackrel{a}{\Longrightarrow}$ from the other
3. the game continues from α' vs. β'.

Weak Bisimulation $\approx$

$\alpha \approx \beta$ iff Duplicator has a strategy to win the game from α vs. β.

Approximants

Idea

$\alpha \sim_i \beta$ iff Duplicator can survive i rounds of the game.

Approximants

Idea

$\alpha \sim_i \beta$ iff Duplicator can survive i rounds of the game.

$\sim_0 \supseteq \sim_1 \supseteq \sim_2 \cdots$

Approximants

Idea

$\alpha \sim_i \beta$ iff Duplicator can survive i rounds of the game.

$$\sim_0 \supseteq \sim_1 \supseteq \sim_2 \dots \supseteq \sim_\omega = \bigcap_{i \in \mathbb{N}} \sim_i$$

Approximants

Idea

$\alpha \sim_i \beta$ iff Duplicator can survive i rounds of the game.

For image-finite systems:

$$\sim_0 \supseteq \sim_1 \supseteq \sim_2 \ldots \supseteq \sim_\omega = \bigcap_{i \in \mathbb{N}} \sim_i = \sim$$

Approximants

Idea

$\alpha \sim_i \beta$ iff Duplicator can survive i rounds of the game.

For image-finite systems:

$$\sim_0 \supseteq \sim_1 \supseteq \sim_2 \dots \supseteq \sim_\omega = \bigcap_{i \in \mathbb{N}} \sim_i = \sim$$

In general:

$$\sim_0 \supseteq \sim_\omega \supseteq \sim_{\omega+\omega} \supseteq \sim_{\omega*\omega} \supseteq \sim_{\omega^\omega} \supseteq \sim$$

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[d, "a"] \arrow[loop right, "\tau Q"] \\
Z \arrow[r, "\tau"] \arrow[loop left, "\tau Q"] & \varepsilon \\
 & Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
$$

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[loop right, "\tau Q"] \arrow[d, "a"] \\
Z \arrow[loop left, "\tau Q"] \arrow[r, "\tau"] & \varepsilon \\
& Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}$$

X $\qquad\qquad$ Y

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$\begin{tikzcd} X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[loop right, "\tau Q"] \arrow[d, "a"] \\ Z \arrow[loop left, "\tau Q"] \arrow[r, "\tau"] & \varepsilon \\ & Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"'] \end{tikzcd}$$

$$\begin{tikzcd} X \arrow[d, Rightarrow, "a"] & & Y \\ Z & & \end{tikzcd}$$

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[loop right, "\tau Q"] \arrow[d, "a"] \\
Z \arrow[loop left, "\tau Q"] \arrow[r, "\tau"] & \varepsilon \\
 & Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
$$

$$
\begin{tikzcd}
X \arrow[d, Rightarrow, "a"] \\
Z
\end{tikzcd}
\qquad
\begin{tikzcd}
Y \arrow[d, Rightarrow, "a"] \\
Q^n
\end{tikzcd}
$$

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[loop right, "\tau Q"] \arrow[d, "a"] \\
Z \arrow[loop left, "\tau Q"] \arrow[r, "\tau"] & \varepsilon \\
 & Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[d, Rightarrow, "a"] \\
Z \arrow[d, Rightarrow, "b"] \\
Z
\end{tikzcd}
\qquad
\begin{tikzcd}
Y \arrow[d, Rightarrow, "a"] \\
Q^n
\end{tikzcd}
$$

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[loop right, "\tau Q"] \arrow[d, "a"] \\
Z \arrow[r, "\tau"] \arrow[loop left, "\tau Q"] & \varepsilon \\
& Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[d, Rightarrow, "a"] \\
Z \arrow[d, Rightarrow, "b"] \\
Z
\end{tikzcd}
\qquad
\begin{tikzcd}
Y \arrow[d, Rightarrow, "a"] \\
Q^n \arrow[d, Rightarrow, "b"] \\
Q^{n-1}
\end{tikzcd}
$$

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[loop right, "\tau Q"] \arrow[d, "a"] \\
Z \arrow[loop left, "\tau Q"] \arrow[r, "\tau"] & \varepsilon \\
& Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
$$

$$
\begin{tikzcd}
X \arrow[d, Rightarrow, "a"] \\
Z \arrow[d, Rightarrow, "b"] \\
Z \arrow[d, dotted, "b^{n-1}"] \\
Z
\end{tikzcd}
\qquad
\begin{tikzcd}
Y \arrow[d, Rightarrow, "a"] \\
Q^n \arrow[d, Rightarrow, "b"] \\
Q^{n-1}
\end{tikzcd}
$$

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[d, "a"] \arrow[loop right, "\tau Q"] \\
Z \arrow[r, "\tau"] \arrow[loop left, "\tau Q"] & \varepsilon \\
 & Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
\qquad
\begin{tikzcd}
X \arrow[d, Rightarrow, "a"] \\
Z \arrow[d, Rightarrow, "b"] \\
Z \arrow[d, dotted, "b^{n-1}"] \\
Z
\end{tikzcd}
\qquad
\begin{tikzcd}
Y \arrow[d, Rightarrow, "a"] \\
Q^n \arrow[d, Rightarrow, "b"] \\
Q^{n-1} \arrow[d, dotted, "b^{n-1}"] \\
\varepsilon
\end{tikzcd}
$$

A Guessing Game

> Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[d, "a"] \arrow[loop right, "\tau Q"] \\
Z \arrow[r, "\tau"] \arrow[loop left, "\tau Q"] & \varepsilon \\
 & Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
$$

$$
\begin{tikzcd}
X \arrow[d, Rightarrow, "a"] & & Y \arrow[d, Rightarrow, "a"] \\
Z \arrow[d, Rightarrow, "b"] & & Q^n \arrow[d, Rightarrow, "b"] \\
Z \arrow[d, dotted, "b^{n-1}"] & & Q^{n-1} \arrow[d, dotted, "b^{n-1}"] \\
Z & \approx_0 & \varepsilon
\end{tikzcd}
$$

A Guessing Game

> Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[d, "a"] \arrow[loop right, "\tau Q"] \\
Z \arrow[r, "\tau"] \arrow[loop left, "\tau Q"] & \varepsilon \\
& Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
$$

$$
\begin{tikzcd}
X \arrow[d, Rightarrow, "a"] & & Y \arrow[d, Rightarrow, "a"] \\
Z \arrow[d, Rightarrow, "b"] & & Q^n \arrow[d, Rightarrow, "b"] \\
Z \arrow[d, dotted, Rightarrow, "b^{n-1}"] & \approx_{n-1} & Q^{n-1} \arrow[d, dotted, Rightarrow, "b^{n-1}"] \\
Z & \approx_0 & \varepsilon
\end{tikzcd}
$$

A Guessing Game

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[loop right, "\tau Q"] \arrow[d, "a"] \\
Z \arrow[r, "\tau"] \arrow[loop left, "\tau Q"] & \varepsilon \\
& Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
$$

$$
\begin{tikzcd}
X \arrow[d, Rightarrow, "a"] & & Y \arrow[d, Rightarrow, "a"] \\
Z \arrow[d, Rightarrow, "b"] & \approx_n & Q^n \arrow[d, Rightarrow, "b"] \\
Z \arrow[d, Rightarrow, dotted, "b^{n-1}"] & \approx_{n-1} & Q^{n-1} \arrow[d, Rightarrow, dotted, "b^{n-1}"] \\
Z & \approx_0 & \varepsilon
\end{tikzcd}
$$

A Guessing Game

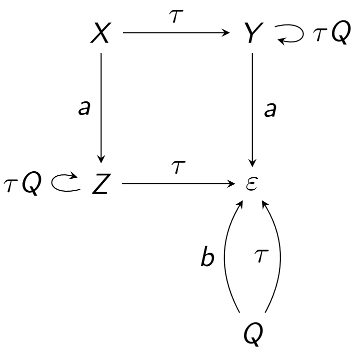
Pick a number; stepwise decrease by some positive integer; lose when you go below 0.

$$\begin{tikzcd}
 & X \arrow[r, "\tau"] \arrow[d, "a"'] & Y \arrow[loop right, "\tau Q"] \arrow[d, "a"] \\
 & Z \arrow[loop left, "\tau Q"] \arrow[r, "\tau"] & \varepsilon \\
 & & Q \arrow[u, bend left, "b"] \arrow[u, bend right, "\tau"']
\end{tikzcd}$$

$$\begin{tikzcd}
X \arrow[d, Rightarrow, "a"] & \approx_{n+1} & Y \arrow[d, Rightarrow, "a"] \\
Z \arrow[d, Rightarrow, "b"] & \approx_{n} & Q^n \arrow[d, Rightarrow, "b"] \\
Z \arrow[d, Rightarrow, dotted, "b^{n-1}"] & \approx_{n-1} & Q^{n-1} \arrow[d, Rightarrow, dotted, "b^{n-1}"] \\
Z & \approx_{0} & \varepsilon
\end{tikzcd}$$

A Guessing Game ($X \approx_\omega Y \not\approx X$)

Pick a number; stepwise decrease by some positive integer; lose when you go below 0.



$$
\begin{array}{ccc}
X & \approx_\omega & Y \\
\Big\Downarrow a & & \Big\Downarrow a \\
Z & \approx_n & Q^n \\
\Big\Downarrow b & & \Big\Downarrow b \\
Z & \approx_{n-1} & Q^{n-1} \\
\vdots\ b^{n-1} & & \vdots\ b^{n-1} \\
Z & \approx_0 & \varepsilon
\end{array}
$$

Using Approximants to decide Weak Bisimilarity

Semi-deciding $\approx$

. . . can be done (Esparza '97).

Using Approximants to decide Weak Bisimilarity

Semi-deciding $\approx$

...can be done (Esparza '97).

Semi-deciding $\not\approx$

For every finite approximant level i check if $(\alpha, \beta) \notin \approx_i$

Using Approximants to decide Weak Bisimilarity

Semi-deciding $\approx$

...can be done (Esparza '97).

Semi-deciding $\not\approx$

For every finite approximant level i check if $(\alpha, \beta) \notin \approx_i$

1. needs $\approx = \bigcap_{i \in \mathbb{N}} \approx_i$

Using Approximants to decide Weak Bisimilarity

Semi-deciding $\approx$

. . . can be done (Esparza '97).

Semi-deciding $\not\approx$

For every finite approximant level i check if $(\alpha, \beta) \notin \approx_i$

1. needs $\approx = \bigcap_{i \in \mathbb{N}} \approx_i$
2. needs decidable $\approx_i$

Using Approximants to decide Weak Bisimilarity

Semi-deciding $\approx$

...can be done (Esparza '97).

Semi-deciding $\not\approx$

For every finite approximant level i check if $(\alpha, \beta) \notin \approx_i$

1. needs $\approx = \bigcap_{i\in\mathbb{N}} \approx_i$
2. needs decidable $\approx_i$ ✓

Using Approximants to decide Weak Bisimilarity

Semi-deciding $\approx$

…can be done (Esparza '97).

Semi-deciding $\not\approx$

For every finite approximant level i check if $(\alpha, \beta) \notin \approx_i$

1. needs $\approx = \bigcap_{i \in \mathbb{N}} \approx_i$ ✗
2. needs decidable $\approx_i$ ✓

A Conjecture

Conjecture (Jančar, Hirshfeld)

For any BPP process description, $\approx = \approx_{\omega+\omega}$

A Conjecture

Conjecture (Jančar, Hirshfeld)

For any BPP process description, $\approx = \approx_{\omega+\omega}$

- If this holds *and we could decide* $\approx_\omega$ then we could semi-decide $\not\approx$.

A Conjecture

Conjecture (Jančar, Hirshfeld)

For any BPP process description, $\approx = \approx_{\omega+\omega}$

- If this holds *and we could decide* $\approx_\omega$ then we could semi-decide $\not\approx$.
- To falsify we look for inequivalent processes $\alpha \approx_{\omega+\omega} \beta$.

Non-convergence of $\approx_i$ at level $\omega + \omega$

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "c"'] & Y \arrow[loop right, "\tau A"] \arrow[d, "c"] \\
Z \arrow[r, "\tau"] \arrow[loop left, "\tau A"] & \varepsilon \\
 & A \arrow[u, bend left, "a"] \arrow[u, bend right, "\tau"']
\end{tikzcd}
$$

$$X \approx_\omega Y \not\approx X$$

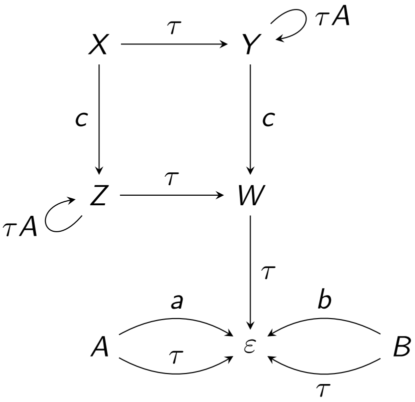
Non-convergence of $\approx_i$ at level $\omega + \omega$

$$
\begin{tikzcd}
X \arrow[r, "\tau"] \arrow[d, "c"'] & Y \arrow[loop right, "\tau A"] \arrow[d, "c"] \\
Z \arrow[loop left, "\tau A"] \arrow[r, "\tau"] & W \arrow[d, "\tau"] \\
A \arrow[r, bend left, "a"] \arrow[r, bend right, "\tau"'] & \varepsilon
\end{tikzcd}
$$

$$X \approx_\omega Y \not\approx X$$

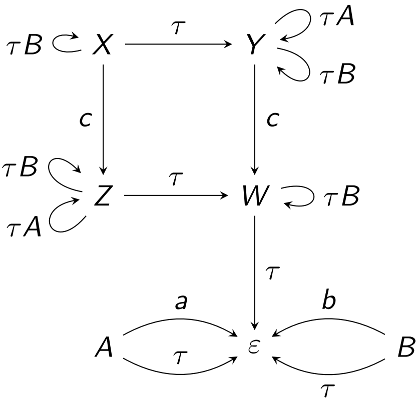
$$W \approx \varepsilon$$

Non-convergence of $\approx_i$ at level $\omega + \omega$



$$X \approx_\omega Y \not\approx X$$
$$W \approx \varepsilon$$

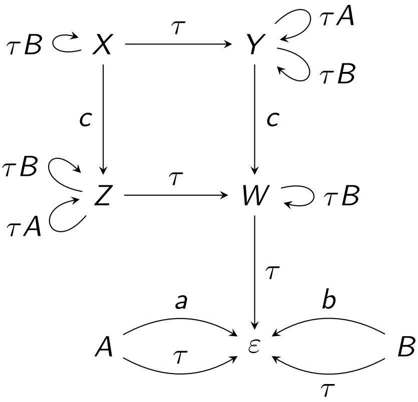
Non-convergence of $\approx_i$ at level $\omega + \omega$



$$X \approx_\omega Y \not\approx X$$

$$W \approx \varepsilon$$

Non-convergence of $\approx_i$ at level $\omega + \omega$



$$X \approx_\omega Y \not\approx X$$

$$W \approx \varepsilon$$

$$XB^m \approx_\omega YB^n \not\approx XB^m$$

Non-convergence of $\approx_i$ at level $\omega + \omega$

$$XB^m \approx_\omega YB^n \not\approx XB^m$$

X Y

Non-convergence of $\approx_i$ at level $\omega + \omega$

$$
\begin{tikzcd}
X_2 \arrow[r, "\tau"] \arrow[d, "c"'] & Y_2 \arrow[loop above, "\tau B"] \arrow[d, "c"'] & \\
Z_2 \arrow[r, "\tau"] \arrow[loop below left, "\tau B"] & W_2 \arrow[r, "\tau"'] & \varepsilon
\end{tikzcd}
$$

X $\quad$ Y

$$X_2 \approx_\omega Y_2 \not\approx X_2$$

$$XB^m \approx_\omega YB^n \not\approx XB^m$$

Non-convergence of $\approx_i$ at level $\omega + \omega$

$$
\begin{tikzcd}
X_2 \arrow[r, "\tau"] \arrow[d, "c"'] & Y_2 \arrow[d, "c"] \arrow[loop above, "\tau B"] & \\
Z_2 \arrow[r, "\tau"] \arrow[loop below, "\tau B"] & W_2 \arrow[r, "b"'] & S_2 \arrow[loop above, "\tau B"] \\
X & Y &
\end{tikzcd}
$$

$$X_2 \approx Y_2 \approx X_2$$

$$XB^m \approx_\omega YB^n \not\approx XB^m$$

Non-convergence of $\approx_i$ at level $\omega + \omega$

$$
\begin{tikzcd}
X_2 \arrow[r, "\tau"] \arrow[d, "c"'] & Y_2 \arrow[loop above, "\tau B"] \arrow[d, "c"'] & \\
Z_2 \arrow[loop left, "\tau B"] \arrow[r, "\tau"] & W_2 \arrow[r, "b"'] \arrow[dl, "d"'] & S_2 \arrow[loop above, "\tau B"] \arrow[dl, "d"'] \\
X & Y &
\end{tikzcd}
$$

$$X_2 \approx_\omega Y_2 \not\approx X_2$$
$$XB^m \approx_\omega YB^n \not\approx XB^m$$

Non-convergence of $\approx_i$ at level $\omega + \omega$

$$
\begin{tikzcd}
X_2 \arrow[r, "\tau"] \arrow[d, "c"'] & Y_2 \arrow[loop above, "\tau B"] \arrow[d, "c"'] & \\
Z_2 \arrow[loop left, "\tau B"] \arrow[r, "\tau"] & W_2 \arrow[r, "b"'] \arrow[dl, "d"'] & S_2 \arrow[loop above, "\tau B"] \arrow[dl, "d"'] \\
X & Y &
\end{tikzcd}
$$

$$X_2 \approx_\omega Y_2 \not\approx X_2$$

$$XB^m \approx_\omega YB^n \not\approx XB^m$$

$$Z_2 \approx_{\omega+1} S_2 \not\approx Z_2$$

Non-convergence of $\approx_i$ at level $\omega + \omega$

$$
\begin{tikzcd}
X_2 \arrow[r, "\tau"] \arrow[d, "c"'] & Y_2 \arrow[d, "c"'] \arrow[loop right, "\tau B"] & \\
Z_2 \arrow[r, "\tau"] \arrow[loop left, "\tau B"] & W_2 \arrow[r, "b"'] \arrow[dl, "d"'] & S_2 \arrow[dl, "d"'] \arrow[loop right, "\tau B"] \\
X & Y &
\end{tikzcd}
$$

$$X_2 \approx_\omega Y_2 \not\approx X_2$$

$$XB^m \approx_\omega YB^n \not\approx XB^m$$

$$Z_2 \approx_{\omega+1} S_2 \not\approx Z_2$$

$$Z_2 \approx_{\omega+2} W_2 \not\approx Z_2$$

Non-convergence of $\approx_i$ at level $\omega + \omega$

$$
\begin{tikzcd}
X_2 \arrow[r, "\tau"] \arrow[d, "c"'] & Y_2 \arrow[d, "c"'] \arrow[loop right, "\tau B"] & \\
Z_2 \arrow[r, "\tau"] \arrow[loop left, "\tau B"] & W_2 \arrow[r, "b"'] \arrow[dl, "d"'] & S_2 \arrow[dl, "d"'] \arrow[loop right, "\tau B"] \\
X & Y &
\end{tikzcd}
$$

$$
\begin{aligned}
X_2 \approx_{\omega+\omega} Y_2 &\not\approx X_2 \\
XB^m \approx_{\omega} YB^n &\not\approx XB^m \\
Z_2 \approx_{\omega+1} S_2 &\not\approx Z_2 \\
Z_2 \approx_{\omega+2} W_2 &\not\approx Z_2
\end{aligned}
$$

Non-convergence of $\approx_i$ at level $\omega + \omega$

$$\begin{tikzcd}
X_2 \arrow[r, "\tau"] \arrow[d, "c"'] & Y_2 \arrow[loop above, "\tau B"] \arrow[d, "c"'] & \\
Z_2 \arrow[loop below, "\tau B"] \arrow[r, "\tau"] & W_2 \arrow[r, "b"'] \arrow[dl, "d"'] & S_2 \arrow[loop above, "\tau B"] \arrow[dl, "d"'] \\
X & Y &
\end{tikzcd}$$

$$X_2 \approx_{\omega+\omega} Y_2 \not\approx X_2$$
$$XB^m \approx_\omega YB^n \not\approx XB^m$$
$$Z_2 \approx_{\omega+1} S_2 \not\approx Z_2$$
$$Z_2 \approx_{\omega+2} W_2 \not\approx Z_2$$

Stacking this k times yields processes $X_k \approx_{\omega * k} Y_k \not\approx X_k$.

Norms

The *norm* of process α:

the length of the shortest word $w = a_0 a_1 \cdots a_n$ such that $\alpha \xRightarrow{a_0} \xRightarrow{a_1} \cdots \xRightarrow{a_n} \varepsilon$ and ∞ if no such words exists.

Norms

The *norm* of process α:

the length of the shortest word $w = a_0 a_1 \cdots a_n$ such that $\alpha \xRightarrow{a_0} \xRightarrow{a_1} \cdots \xRightarrow{a_n} \varepsilon$ and ∞ if no such words exists.

Properties

- Norms can be easily computed
- Norm equality is an invariant for $\approx$

Faster Approximants

Can we define faster converging approximants?

Faster Approximants

Can we define faster converging approximants?

Example: Word Approximants

$$\begin{tikzcd} \alpha \arrow[d, Rightarrow, "word"'] & \approx_{i+1} & \beta \arrow[d, Rightarrow, "word"'] \\ \alpha' & \approx_i & \beta' \end{tikzcd}$$

Faster Approximants

Can we define faster converging approximants?

Example: Word Approximants

$$\begin{tikzcd}
\alpha \arrow[d, "word"'] \arrow[r, phantom, "\approx_{i+1}"] & \beta \arrow[d, "word"'] \\
\alpha' \arrow[r, phantom, "\approx_i"] & \beta'
\end{tikzcd}$$

Example: Add Invariant P

Let $\approx_i^P$ enforce that Duplicator must preserve P. Then for every i,

$$\approx_i \supseteq \approx_i^P$$

Jitka's Class

Jitka's Class (Stribrna '98: $\approx = \approx_{\omega+\omega}$)

- one visible action symbol
- no zero-norm variables (and no trivial deadlocks)

Jitka's Class

Jitka's Class (Stribrna '98: $\approx = \approx_{\omega+\omega}$)

- one visible action symbol
- no zero-norm variables (and no trivial deadlocks)

Properties

- If $norm(\alpha) = norm(\beta) = \infty$ then $\alpha \approx \beta$
- $Succ_n(\alpha) = \{\alpha' : \alpha \stackrel{a}{\Longrightarrow} \alpha' \wedge |\alpha'| = n < \infty\}$ is finite

Jitka's Class

Jitka's Class (Stribrna '98: $\approx = \approx_{\omega+\omega}$)

- one visible action symbol
- no zero-norm variables (and no trivial deadlocks)

Properties

- If $norm(\alpha) = norm(\beta) = \infty$ then $\alpha \approx \beta$
- $Succ_n(\alpha) = \{\alpha' : \alpha \stackrel{a}{\Longrightarrow} \alpha' \wedge |\alpha'| = n < \infty\}$ is finite

Approximants with Norm test

Check norm equality in each round. Then $\approx_\omega^N = \approx$.

Word Approximants ($\approx_i^W$)

1. are undecidable already at level 1 for normed BPP

Word Approximants ($\approx_i^W$)

1. are undecidable already at level 1 for normed BPP ✗

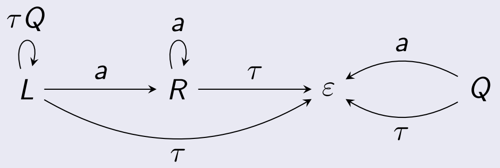
Word Approximants ($\approx_i^W$)

1. are undecidable already at level 1 for normed BPP ✗
2. do not converge at level ω

Word Approximants ($\approx_i^W$)

1. are undecidable already at level 1 for normed BPP ✗
2. do not converge at level ω ✗

Idea: Preserve equality on "contains R" to force short steps

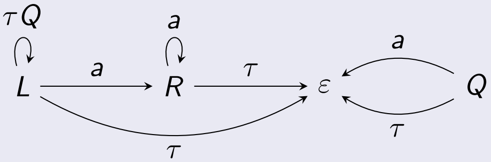


Play L^{n+1} vs. L^n, where n is Duplicator's choice

Word Approximants ($\approx_i^W$)

1. are undecidable already at level 1 for normed BPP ✗
2. do not converge at level ω ✗

Idea: Preserve equality on "contains R" to force short steps



Play L^{n+1} vs. L^n, where n is Duplicator's choice

This is useful!

Any successful approach will define a computable $\not\approx_\omega^W \subsetneq D \subseteq \not\approx$.

Hopeless invariants

Norm Equality

If α and β has different norms then $\approx_2^W$ distinguishes them.

Hopeless invariants

Norm Equality

If α and β has different norms then $\approx_2^W$ distinguishes them.

Ability to perform an infinite sequence a^ω

Captured by $\approx_1^W$.

Hopeless invariants

Norm Equality

If α and β has different norms then $\approx_2^W$ distinguishes them.

Ability to perform an infinite sequence a^ω

Captured by $\approx_1^W$.

Distance to disable action a (and higher level DD-functions)

(Distance to)^i disable a is captured by $\approx_{i+1}^W$.

Conclusion

- Hirschfeld's Conjecture ($\approx = \approx_{\omega+\omega}$) is false

Conclusion

- Hirschfeld's Conjecture ($\approx = \approx_{\omega+\omega}$) is false
- New proofs for decidability of subclasses

Conclusion

- Hirschfeld's Conjecture ($\approx = \approx_{\omega+\omega}$) is false
- New proofs for decidability of subclasses
- Lower bound of $\omega + \omega$ for convergence of "Word Approximants"

Conclusion

- Hirschfeld's Conjecture ($\approx = \approx_{\omega+\omega}$) is false
- New proofs for decidability of subclasses
- Lower bound of $\omega + \omega$ for convergence of "Word Approximants"
- Found a distinguishing property? Make sure it's not captured by $\approx_\omega^W$!

Bibliography

- Javier Esparza.
  Petri nets, commutative context-free grammars, and basic parallel processes.
  Fundam. Inform., 31(1):13–25, 1997.

- Will Harwood, Faron Moller, and Anton Setzer.
  Weak bisimulation approximants.
  In *CSL*, pages 365–379, 2006.

- Robin Milner.
  Communication and concurrency.
  PHI Series in computer science. Prentice Hall, 1989.

- Colin Stirling.
  Decidability of weak bisimilarity for a subset of basic parallel processes.
  In *FoSSaCS*, pages 379–393, 2001.

- Jitka Stríbrná.
  Decidability and complexity of equivalences for simple process algebras.
  PhD thesis, School of Informatics, University of Edinburgh, 1998.