Motivation	Tool presentation	In the end

PREG Axiomatizer – A Ground Bisimilarity Checker for GSOS with Predicates

Luca Aceto¹, Georgiana Caltais¹, Eugen-Ioan Goriac¹, Anna Ingólfsdóttir¹

¹Reykjavik University ICE-TCS, Iceland

Motivation	Tool presentation	In the end
•••••••		
Introduction		
Purpose		

Check for behavioral equivalences

- between processes specified using GSOS operators
- faster than by just applying the definition

Extend the expressiveness of the GSOS framework for giving semantics to operators

with predicates

Motivation	Tool presentation	In the end
•••••••		
Introduction		
Purpose		

Check for behavioral equivalences

- between processes specified using GSOS operators
- faster than by just applying the definition

Extend the expressiveness of the GSOS framework for giving semantics to operators

• with predicates

Motivation	Tool presentation	In the end
00000000	0000000	00
Introduction		
Pre . (lude + liminaries)		

• preg transition rules $(\mathcal{R}^{\mathcal{A}})$:

 $\{x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+\} \qquad \{P_{ij}x_i \mid i \in J^+\} \\ \{x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i\} \quad \{\neg Qx_i \mid i \in J^-, Q \in \mathcal{Q}_i\}$

$$f(x_1,\ldots,x_l) \xrightarrow{c} C[\vec{x},\vec{y}]$$

• preg predicate rules $(\mathcal{R}^{\mathcal{P}})$:

$$\begin{cases} x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+ \} & \{P_{ij}x_i \mid i \in J^+ \} \\ \{x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i\} & \{\neg Qx_i \mid i \in J^-, Q \in \mathcal{Q}_i\} \end{cases}$$

$$P(f(x_1, \dots, x_l))$$

preg system: $G = (\Sigma, \mathcal{R}^{\mathcal{A}} \cup \mathcal{R}^{\mathcal{P}})$

<ロト <回ト < 三ト

Motivation	Tool presentation	In the end
0000000		
Introduction		
Pre . (lude + liminaries)		

• preg transition rules $(\mathcal{R}^{\mathcal{A}})$:

 $\begin{cases} x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+ \} & \{ P_{ij} \times_i \mid i \in J^+ \} \\ \{ x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i \} & \{ \neg Q_{X_i} \mid i \in J^-, Q \in \mathcal{Q}_i \} \end{cases}$ $f(x_1, \dots, x_l) \xrightarrow{c} C[\vec{x}, \vec{y}]$

• preg predicate rules $(\mathcal{R}^{\mathcal{P}})$:

 $\{x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+\} \qquad \{P_{ij}x_i \mid i \in J^+\}$ $\{x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i\} \quad \{\neg Qx_i \mid i \in J^-, Q \in \mathcal{Q}_i\}$ $P(f(x_1, \dots, x_l))$

preg system: $G = (\Sigma, \mathcal{R}^{\mathcal{A}} \cup \mathcal{R}^{\mathcal{P}})$

- 4 同 ト 4 ヨ ト - 4 ヨ ト - -

Motivation	Tool presentation	In the end
0000000		
Introduction		
Pre . (lude + liminaries)		

• preg transition rules $(\mathcal{R}^{\mathcal{A}})$:

 $\begin{cases} x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+ \} & \{ P_{ij}x_i \mid i \in J^+ \} \\ \{ x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i \} & \{ \neg Qx_i \mid i \in J^-, Q \in \mathcal{Q}_i \} \end{cases}$ $f(x_1, \dots, x_l) \xrightarrow{c} C[\vec{x}, \vec{y}]$

• preg predicate rules $(\mathcal{R}^{\mathcal{P}})$:

 $\{ x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+ \}$ $\{ P_{ij}x_i \mid i \in J^+ \}$ $\{ x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i \}$ $\{ \neg Qx_i \mid i \in J^-, Q \in \mathcal{Q}_i \}$ $P(f(x_1, \dots, x_l))$

preg system: $G = (\Sigma, \mathcal{R}^{\mathcal{A}} \cup \mathcal{R}^{\mathcal{P}})$

《口》 《聞》 《臣》 《臣》 二臣

Motivation	Tool presentation	In the end
00000000		
Introduction		
Pre (lude + liminaries)		

• preg transition rules $(\mathcal{R}^{\mathcal{A}})$:

 $\begin{cases} x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+ \} & \{ P_{ij}x_i \mid i \in J^+ \} \\ \{ x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i \} & \{ \neg Qx_i \mid i \in J^-, Q \in \mathcal{Q}_i \} \\ \hline f(x_1, \dots, x_l) \xrightarrow{c} C[\vec{x}, \vec{y}] \end{cases}$

• preg predicate rules $(\mathcal{R}^{\mathcal{P}})$:

$$\begin{cases} x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+ \} & \{P_{ij}x_i \mid i \in J^+ \} \\ \{x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i\} & \{\neg Qx_i \mid i \in J^-, Q \in \mathcal{Q}_i\} \end{cases}$$

$$P(f(x_1, \dots, x_l))$$

preg system: $G = (\Sigma, \mathcal{R}^{\mathcal{A}} \cup \mathcal{R}^{\mathcal{P}})$

Motivation	Tool presentation	In the end
00000000		
Introduction		
Pre (lude $+$ l	iminaries)	

• preg transition rules $(\mathcal{R}^{\mathcal{A}})$:

$$\begin{cases} x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+ \} & \{ P_{ij}x_i \mid i \in J^+ \} \\ \{ x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i \} & \{ \neg Qx_i \mid i \in J^-, Q \in \mathcal{Q}_i \} \\ \hline f(x_1, \dots, x_l) \xrightarrow{c} C[\vec{x}, \vec{y}] \end{cases}$$

• preg predicate rules $(\mathcal{R}^{\mathcal{P}})$:

$$\begin{array}{c} \{x_i \xrightarrow{a_{ij}} y_{ij} \mid i \in I^+\} & \{P_{ij}x_i \mid i \in J^+\} \\ \{x_i \xrightarrow{b} \mid i \in I^-, b \in \mathcal{B}_i\} & \{\neg Qx_i \mid i \in J^-, Q \in \mathcal{Q}_i\} \\ \hline P(f(x_1, \dots, x_l)) \end{array}$$

preg system: $G = (\Sigma, \mathcal{R}^{\mathcal{A}} \cup \mathcal{R}^{\mathcal{P}})$

Motivation ○○●○○○○○○	Tool presentation	In the end
Case Study		
Finite trees	Parallel composition _ _	Immediate termination \downarrow

Semantics: $\begin{pmatrix} \frac{x}{a.x} \xrightarrow{a} x & \frac{x}{x+y} \xrightarrow{a} x' & \frac{y}{x+y} \xrightarrow{a} y' \\ \frac{x}{x+y} \xrightarrow{a} x' & \frac{y}{x+y} \xrightarrow{a} y' \\ \frac{x}{(x+y)} & \frac{y}{(x+y)} \xrightarrow{y} \\ \frac{x}{x \parallel y} \xrightarrow{a} x' \parallel y & \frac{y}{x \parallel y} \xrightarrow{a} x \parallel y' & \frac{x \downarrow y}{(x \parallel y)} \\ \end{pmatrix}$

個人 くほん くほん しほ

Motivation ○○●○○○○○○	Tool presentation	In the end
Case Study		
Finite trees	Parallel composition _ _	Immediate termination \downarrow

Semantics: $\begin{pmatrix} \frac{1}{a.x \xrightarrow{a} x} & \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} & \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'} \\ \frac{1}{\kappa_{\downarrow} \downarrow} & \frac{x \downarrow}{(x + y) \downarrow} & \frac{y \downarrow}{(x + y) \downarrow} \\ \frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} & \frac{y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'} & \frac{x \downarrow y \downarrow}{(x \parallel y) \downarrow} \end{pmatrix}$

個人 くほん くほん しほ

Motivation ○○●○○○○○○	Tool presentation	In the end
Case Study		
Finite trees	Parallel composition $_{ _}$	Immediate termination \downarrow

Semantics: $\begin{pmatrix} \frac{x \xrightarrow{a} x'}{a.x \xrightarrow{a} x} & \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} & \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'} \\ \frac{x \xrightarrow{a} x}{(x + y) \downarrow} & \frac{y \downarrow}{(x + y) \downarrow} \\ \frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} & \frac{y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'} & \frac{x \downarrow y \downarrow}{(x \parallel y) \downarrow} \end{pmatrix}$

▲圖 ▶ ▲ 臣 ▶ ▲ 臣 ▶ ― 臣

Motivation ○○●○○○○○○	Tool presentation	In the end
Case Study		
Finite trees	Parallel composition $_{ _}$	Immediate termination \downarrow

Semantics: $\begin{pmatrix} \frac{1}{a.x \xrightarrow{a} x} & \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} & \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'} \\ \frac{1}{\kappa_{\downarrow} \downarrow} & \frac{x \downarrow}{(x + y) \downarrow} & \frac{y \downarrow}{(x + y) \downarrow} \\ \frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} & \frac{y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'} & \frac{x \downarrow y \downarrow}{(x \parallel y) \downarrow} \end{pmatrix}$

Eugen-Ioan Goriac (Reykjavik Uiversity)

個人 くほん くほん 一足

Motivation ○○●○○○○○○	Tool presentation	In the end
Case Study		
Finite trees	Parallel composition $_{ _}$	Immediate termination \downarrow

Semantics: $\begin{pmatrix} \frac{x \xrightarrow{a} x'}{a.x \xrightarrow{a} x} & \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} & \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'} \\ \frac{x \downarrow}{\kappa_{\downarrow} \downarrow} & \frac{x \downarrow}{(x + y) \downarrow} & \frac{y \downarrow}{(x + y) \downarrow} \\ \frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} & \frac{y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'} & \frac{x \downarrow y \downarrow}{(x \parallel y) \downarrow} \end{pmatrix}$

Eugen-Ioan Goriac (Reykjavik Uiversity)

個人 くほん くほん 一足

Motivation ○○●○○○○○○	Tool presentation	In the end
Case Study		
Finite trees	Parallel composition _ _	Immediate termination \downarrow

Semantics: $\begin{pmatrix} \frac{x \xrightarrow{a} x'}{a.x \xrightarrow{a} x} & \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} & \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'} \\ \frac{x \xrightarrow{b} x}{\kappa_{\downarrow} \downarrow} & \frac{x \downarrow}{(x + y) \downarrow} & \frac{y \downarrow}{(x + y) \downarrow} \\ \frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} & \frac{y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'} & \frac{x \downarrow y \downarrow}{(x \parallel y) \downarrow} \end{pmatrix}$

Motivation ○○●○○○○○○	Tool presentation	In the end
Case Study		
Finite trees	Parallel composition _ _	Immediate termination \downarrow

Semantics: $\begin{pmatrix} \frac{x \xrightarrow{a} x'}{a.x \xrightarrow{a} x} & \frac{x \xrightarrow{a} x'}{x + y \xrightarrow{a} x'} & \frac{y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'} \\ \frac{x \xrightarrow{b} x}{\kappa_{\downarrow} \downarrow} & \frac{x \downarrow}{(x + y) \downarrow} & \frac{y \downarrow}{(x + y) \downarrow} \\ \frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y} & \frac{y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'} & \frac{x \downarrow y \downarrow}{(x \parallel y) \downarrow} \end{pmatrix}$

◎ ▶ ▲ 臣 ▶ ▲ 臣 ▶ ▲ 臣 ● � � � �

Motivation	Tool presentation	In the end
0000000		
Case Study		
Question		

ls

 $s = a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow}$ strongly bisimilar to $t = a.(a.b.b.\kappa_{\downarrow} + b.(a.b.\kappa_{\downarrow} + b.a.\kappa_{\downarrow})) + b.(a.(a.b.\kappa_{\downarrow} + b.a.\kappa_{\downarrow}) + b.a.a.\kappa_{\downarrow})$?

Answer { 1) the definition of strong bisimilarity by using { 2) an axiomatization modulo bisimilarity

個 と く ヨ と く ヨ と

Motivation	Tool presentation	In the end
0000000		
Case Study		
Question		

ls

 $s = a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow}$ strongly bisimilar to $t = a.(a.b.b.\kappa_{\downarrow} + b.(a.b.\kappa_{\downarrow} + b.a.\kappa_{\downarrow})) + b.(a.(a.b.\kappa_{\downarrow} + b.a.\kappa_{\downarrow}) + b.a.a.\kappa_{\downarrow})$?

Answer { 1) the definition of strong bisimilarity by using { 2) an axiomatization modulo bisimilarity

1) By the definition of strong bisimilarity

Definition (Bisimilarity "⇔")

A symmetric relation R is a bisimulation iff:

- if $s \ R \ t, \ a \in \mathcal{A}$ and $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ and $s' \ R \ t'$;
- if $s \ R \ t$, and $s \downarrow$ then $t \downarrow$.

Terms s and t are bisimilar $(s \Leftrightarrow t)$ iff s R t and R is a bisimulation.

Assume _||_ is associative, commutative, with κ_{\downarrow} as the identity.

Does
$$a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \stackrel{a}{\rightarrow} s'$$
 hold ?

Instantiate
$$\frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y}$$
 as $\frac{a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \xrightarrow{a} s^{ii}}{(a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow}) \parallel b.\kappa_{\downarrow} \xrightarrow{a} s^{ii} \parallel b.\kappa_{\downarrow}}$.

(本間) (本語) (本語)

1) By the definition of strong bisimilarity

Definition (Bisimilarity "⇔")

A symmetric relation R is a bisimulation iff:

- if $s \ R \ t, \ a \in \mathcal{A}$ and $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ and $s' \ R \ t'$;
- if $s \ R \ t$, and $s \downarrow$ then $t \downarrow$.

Terms s and t are bisimilar $(s \Leftrightarrow t)$ iff s R t and R is a bisimulation.

Assume _||_ is associative, commutative, with κ_{\downarrow} as the identity.

Does $a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \stackrel{a}{\rightarrow} s'$ hold ?

Instantiate $\frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y}$ as $\frac{a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \xrightarrow{a} s^{ii}$? $(a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow}) \parallel b.\kappa_{\downarrow} \xrightarrow{a} s^{ii} \parallel b.\kappa_{\downarrow}$.

1) By the definition of strong bisimilarity

Definition (Bisimilarity "⇔")

A symmetric relation R is a bisimulation iff:

- if $s \ R \ t, \ a \in \mathcal{A}$ and $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ and $s' \ R \ t'$;
- if $s \ R \ t$, and $s \downarrow$ then $t \downarrow$.

Terms s and t are bisimilar $(s \Leftrightarrow t)$ iff s R t and R is a bisimulation.

Assume _||_ is associative, commutative, with κ_{\downarrow} as the identity.

So, does
$$a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \xrightarrow{a} s^{ii}$$
 hold ?

Instantiate
$$\frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y}$$
 as $\frac{a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \xrightarrow{a} s^{iii}}{(a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow}) \parallel b.\kappa_{\downarrow} \xrightarrow{a} s^{iii} \parallel b.\kappa_{\downarrow}}$

- 4 回 2 - 4 回 2 - 4 回 2 - 4

イロン イ部ン イヨン イヨン 三日

Case Study

1) By the definition of strong bisimilarity

Definition (Bisimilarity "⇔")

A symmetric relation R is a bisimulation iff:

- if $s \ R \ t, \ a \in \mathcal{A}$ and $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ and $s' \ R \ t'$;
- if $s \ R \ t$, and $s \downarrow$ then $t \downarrow$.

Terms s and t are bisimilar $(s \Leftrightarrow t)$ iff s R t and R is a bisimulation.

Assume _||_ is associative, commutative, with κ_{\downarrow} as the identity.

So, does $a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \xrightarrow{a} s^{ii}$ hold ?

Instantiate
$$\frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y}$$
 as $\frac{a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \xrightarrow{a} s^{iii}$?
 $(a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow}) \parallel b.\kappa_{\downarrow} \xrightarrow{a} s^{iii} \parallel b.\kappa_{\downarrow}$

1) By the definition of strong bisimilarity

Definition (Bisimilarity "⇔")

A symmetric relation R is a bisimulation iff:

- if $s \ R \ t, \ a \in \mathcal{A}$ and $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ and $s' \ R \ t'$;
- if $s \ R \ t$, and $s \downarrow$ then $t \downarrow$.

Terms s and t are bisimilar $(s \Leftrightarrow t)$ iff s R t and R is a bisimulation.

Assume _||_ is associative, commutative, with κ_{\downarrow} as the identity.

So, does $a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \xrightarrow{a} s^{iii}$ hold ?

Instantiate
$$\frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y}$$
 as $\frac{a \cdot \kappa_{\downarrow} \xrightarrow{a} s^{iv}}{a \cdot \kappa_{\downarrow} \parallel a \cdot \kappa_{\downarrow} \xrightarrow{a} s^{iv} \parallel a \cdot \kappa_{\downarrow}}$

イロン イ部ン イヨン イヨン 三日

1) By the definition of strong bisimilarity

Definition (Bisimilarity "⇔")

A symmetric relation R is a bisimulation iff:

- if $s \ R \ t, a \in \mathcal{A}$ and $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ and $s' \ R \ t'$;
- if $s \ R \ t$, and $s \downarrow$ then $t \downarrow$.

Terms s and t are bisimilar $(s \Leftrightarrow t)$ iff s R t and R is a bisimulation.

Assume _||_ is associative, commutative, with κ_{\downarrow} as the identity.

So, does $a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \xrightarrow{a} s^{iii}$ hold ?

Instantiate
$$\frac{x \xrightarrow{a} x'}{x \parallel y \xrightarrow{a} x' \parallel y}$$
 as $\frac{a.\kappa_{\downarrow} \xrightarrow{a} s^{iv}$?
 $a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \xrightarrow{a} s^{iv} \parallel a.\kappa_{\downarrow}$.

(日本) (日本) (日本)

1) By the definition of strong bisimilarity

Definition (Bisimilarity "⇔")

A symmetric relation R is a bisimulation iff:

- if $s \ R \ t, a \in \mathcal{A}$ and $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ and $s' \ R \ t'$;
- if $s \ R \ t$, and $s \downarrow$ then $t \downarrow$.

Terms s and t are bisimilar $(s \Leftrightarrow t)$ iff s R t and R is a bisimulation.

Assume _||_ is associative, commutative, with κ_{\downarrow} as the identity.

So, does $a.\kappa_{\downarrow} \xrightarrow{a} s^{i\nu}$ hold ?

Instantiate
$$\frac{a}{a.x \xrightarrow{a} x}$$
 as $\frac{a}{a.\kappa_{\downarrow} \xrightarrow{a} \kappa_{\downarrow}}$.

イロト イポト イヨト イヨト

1) By the definition of strong bisimilarity

Definition (Bisimilarity "⇔")

A symmetric relation R is a bisimulation iff:

- if $s \ R \ t, a \in \mathcal{A}$ and $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ and $s' \ R \ t'$;
- if $s \ R \ t$, and $s \downarrow$ then $t \downarrow$.

Terms s and t are bisimilar $(s \Leftrightarrow t)$ iff s R t and R is a bisimulation.

Assume _||_ is associative, commutative, with κ_{\downarrow} as the identity.

So, does $a.\kappa_{\downarrow} \xrightarrow{a} s^{i\nu}$ hold ?

Instantiate
$$\xrightarrow{a} x \xrightarrow{a} x$$
 as $\overrightarrow{a.\kappa_{\downarrow}} \xrightarrow{a} \kappa_{\downarrow}$. \checkmark

イロン イ部ン イヨン イヨン 三日

1) By the definition of strong bisimilarity

Definition (Bisimilarity "⇔")

A symmetric relation R is a bisimulation iff:

- if $s \ R \ t, a \in \mathcal{A}$ and $s \xrightarrow{a} s'$ then $t \xrightarrow{a} t'$ and $s' \ R \ t'$;
- if $s \ R \ t$, and $s \downarrow$ then $t \downarrow$.

Terms s and t are bisimilar $(s \Leftrightarrow t)$ iff s R t and R is a bisimulation.

Assume _||_ is associative, commutative, with κ_{\downarrow} as the identity.

Therefore, at the end of the day, it holds that:

$$a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \stackrel{a}{\rightarrow} s' = a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow}$$

- 4 同 2 4 三 2 4 三 2 4

Motivation

Tool presentation

In the end $\circ\circ$

Case Study

1) By the definition of strong bisimilarity

$$s = a.\kappa_{\downarrow} \parallel a.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow} \parallel b.\kappa_{\downarrow}$$

$$\mathsf{t} = \mathsf{a}.(\mathsf{a}.b.b.\kappa_{\downarrow} + b.(\mathsf{a}.b.\kappa_{\downarrow} + b.\mathbf{a}.\kappa_{\downarrow})) + b.(\mathsf{a}.(\mathsf{a}.b.\kappa_{\downarrow} + b.\mathbf{a}.\kappa_{\downarrow}) + b.\mathbf{a}.\mathbf{a}.\kappa_{\downarrow})$$



 $s \Leftrightarrow t$

- 4 回 2 - 4 □ 2 - 4 □

3

Motivation	Tool presentation	In the end
000000000		
Case Study		

1) By the definition of strong bisimilarity

Demo

A⊒ ▶ < ∃

Eugen-Ioan Goriac (Reykjavik Uiversity) CALCO'11 PREG Axiomatizer

Μ	otiv	ati	on
	000	000	0.0

Tool presentation

In the end

Case Study

2) By an axiomatization modulo bisimilarity

$$\begin{array}{l} x + x &= x \\ x + y &= y + x \\ (x + y) + z &= x + (y + z) \\ x + \delta &= x \\ \end{array} \\ x \parallel y &= x \parallel^1 y + x \parallel^2 y + x \parallel^3 y \\ x \parallel^1 (y + z) &= x \parallel^1 y + x \parallel^1 z \\ (x + y) \parallel^1 z &= x \parallel^1 z + y \parallel^1 z \\ (x + y) \parallel^2 z &= x \parallel^2 z + y \parallel^2 z \\ x \parallel^3 (y + z) &= x \parallel^3 y + x \parallel^3 z \\ k_{\downarrow} \parallel^1 k_{\downarrow} &= k_{\downarrow} \\ a.x' \parallel^2 y &= a.(x' \parallel^2 y) \\ x \parallel^3 a.y' &= a.(x \parallel^3 y') \\ x \parallel^{1/2/3} y &= \delta, \text{ otherwise} \end{array}$$

Using this axiomatization seems to be less intuitive, however, it is

- much faster, and
- derived for free.

A (1) > A (1) > A

Ν	Л	0	ł	V	a	ł	0	n	
	~	~		~	~	~	-		1

Tool presentation

In the end

Case Study

2) By an axiomatization modulo bisimilarity

$$\begin{array}{l} x + x &= x \\ x + y &= y + x \\ (x + y) + z &= x + (y + z) \\ x + \delta &= x \\ \end{array} \\ x \parallel y &= x \parallel^1 y + x \parallel^2 y + x \parallel^3 y \\ x \parallel^1 (y + z) &= x \parallel^1 y + x \parallel^1 z \\ (x + y) \parallel^1 z &= x \parallel^1 z + y \parallel^1 z \\ (x + y) \parallel^2 z &= x \parallel^2 z + y \parallel^2 z \\ x \parallel^3 (y + z) &= x \parallel^3 y + x \parallel^3 z \\ k_{\downarrow} \parallel^1 k_{\downarrow} &= k_{\downarrow} \\ a.x' \parallel^2 y &= a.(x' \parallel^2 y) \\ x \parallel^3 a.y' &= a.(x \parallel^3 y') \\ x \parallel^{1/2/3} y &= \delta, \text{ otherwise} \end{array}$$

Using this axiomatization seems to be less intuitive, however, it is

▲帰▶ ▲臣▶ ▲臣▶

- much faster, and
- derived for free.

Motivation	Tool presentation	In the end
00000000		
Case Study		
2) By an axiomatization	modulo bisimilarity	

Demo

・ロト ・回ト ・ヨト ・ヨト

æ

Eugen-Ioan Goriac (Reykjavik Uiversity) CALCO'11 PREG Axiomatizer

Motivation	Tool presentation ●○○○○○○	In the end
Introduction		
PREG Axiomatizer		

- the first public tool for automatically deriving sound and ground-complete axiomatizations modulo bisimilarity for GSOS-like languages (to our knowledge)
- downloadable from http://goriac.info/tools/preg-axiomatizer/
- implemented using
 - Maude for the theory (\sim 2000 lines)
 - Python for the graphic user interface (${\sim}300$ lines)

Motivation		Tool presentation	In the end
		000000	
Other case stu	dies		
: a	nd while_do_		

$$\frac{x \xrightarrow{a} x'}{x; y \xrightarrow{a} x'; y} : \begin{array}{c} x - (a) -> X, \\ ===\\ x; y \xrightarrow{a} x'; y \end{array} : \begin{array}{c} x - (a) -> X, \\ ===\\ x; y \xrightarrow{a} y'; y \end{array} : \begin{array}{c} x (a) -> X, \\ ===\\ x; y \xrightarrow{a} y' \end{array} : \begin{array}{c} x; y \xrightarrow{a} y' \\ x; y \xrightarrow{a} y' \end{array} : \begin{array}{c} P(X), Y - (a) -> Y, \\ ===\\ x; y \xrightarrow{a} y' \end{array} : \begin{array}{c} x; y \xrightarrow{a} y' \\ x; y \xrightarrow{a} y' \end{array} : \begin{array}{c} P(X), Y - (a) -> Y, \\ ===\\ y; y \xrightarrow{a} y' \end{array} : \begin{array}{c} x; y \xrightarrow{a} y' \\ ===\\ p(x) = x \\ (while \ X \ do \ y) \downarrow \end{array} : \begin{array}{c} P(X) \\ ===\\ p(while \ X \ do \ Y) \end{array} : \begin{array}{c} x \xrightarrow{a} y' \\ ===\\ y; while \ X \ do \ y) \xrightarrow{a} y; while \ x' \ do \ y \end{array} : \begin{array}{c} x \xrightarrow{a} y' \\ (while \ X \ do \ Y) - (a) -> Y; \end{array}$$

The following holds:

 $a.(a.a.\kappa_{\downarrow}; b.(a.a.\kappa_{\downarrow}; b.a.a.\kappa_{\downarrow})) \Leftrightarrow \text{while } a.b.b.\kappa_{\downarrow} \text{ do } a.a.\kappa_{\downarrow}$.

_ , _

・ 回 と ・ ヨ と ・ ヨ と …

æ

Motivatio	on 000		Tool presentation ○○●○○○○	In the end
Other cas	se studies			
	and	while do		

Demo

æ

Eugen-Ioan Goriac (Reykjavik Uiversity) CALCO'11 PREG Axiomatizer

Motivation	Tool presentation	In the end
	000000	
Other case studies		

$$\frac{x \downarrow y \downarrow}{(x \parallel y) \downarrow} : \qquad P(X), P(Y) = P(X \mid Y)$$

$$\frac{x \stackrel{act}{\rightarrow} x'}{(x \parallel y) \downarrow} : \qquad P(X \mid |Y)$$

$$\frac{x \stackrel{act}{\rightarrow} x'}{x \parallel y \stackrel{act}{\rightarrow} x' \parallel y} : \qquad X - (act) \rightarrow X' = X'$$

$$\frac{y \stackrel{act}{\rightarrow} x' \parallel y}{x \parallel y \stackrel{act}{\rightarrow} x' \parallel y'} : \qquad X \mid |Y - (act) \rightarrow X' \mid |Y$$

$$\frac{x \stackrel{p!d}{\rightarrow} x' y \stackrel{p?d}{\rightarrow} y'}{x \parallel y \stackrel{p\#d}{\rightarrow} x' \parallel y'} : \qquad X - (p!d) \rightarrow X' \mid |Y'$$

$$\frac{x \stackrel{p?d}{\rightarrow} x' y \stackrel{p!d}{\rightarrow} y'}{x \parallel y \stackrel{p\#d}{\rightarrow} x' \parallel y'} : \qquad X - (p!d) \rightarrow X' \mid |Y'$$

Motivation	Tool presentation	In the end
	0000000	
Other case studies		
_ _		



- A, B, C are the communicating processes,
- *ia*, *ab*, *ac*, *co* are the ports, and
- the actions of sending and receiving the datum *d* over the port *p* are denoted by, respectively, *p*!*d* and *p*?*d*.

The whole protocol is specified as the term

 $T = ia?d.(ab!d.\kappa_{\downarrow} \parallel ac!d.\kappa_{\downarrow}) \parallel ab?d.\kappa_{\downarrow} \parallel ac?d.co!d.\kappa_{\downarrow}.$

In order to enforce the communication over the ports *ab* and *ac*, one uses the encapsulation operator:

$$T' = \partial_{\{p!d,p?d \mid p \in \{ab,ac\}\},\emptyset}(T).$$

Motivation	Tool presentation	In the end
	0000000	
Other case studies		
11		



- A, B, C are the communicating processes,
- *ia*, *ab*, *ac*, *co* are the ports, and
- the actions of sending and receiving the datum *d* over the port *p* are denoted by, respectively, *p*!*d* and *p*?*d*.

The whole protocol is specified as the term

 $T = ia?d.(ab!d.\kappa_{\downarrow} \parallel ac!d.\kappa_{\downarrow}) \parallel ab?d.\kappa_{\downarrow} \parallel ac?d.co!d.\kappa_{\downarrow}.$

In order to enforce the communication over the ports *ab* and *ac*, one uses the encapsulation operator:

 $T' = \partial_{\{p!d,p?d \mid p \in \{ab,ac\}\},\emptyset}(T).$

_ 11 _

Image: A math a math

Motivation	Tool presentation	In the end
	0000000	
Other case studies		
11		



- A, B, C are the communicating processes,
- *ia*, *ab*, *ac*, *co* are the ports, and
- the actions of sending and receiving the datum *d* over the port *p* are denoted by, respectively, *p*!*d* and *p*?*d*.

The whole protocol is specified as the term

$${\mathcal T}={\it ia?d.}({\it ab!d.}\kappa_{\downarrow}\parallel{\it ac!d.}\kappa_{\downarrow})\parallel{\it ab?d.}\kappa_{\downarrow}\parallel{\it ac?d.co!d.}\kappa_{\downarrow}.$$

In order to enforce the communication over the ports ab and ac, one uses the encapsulation operator:

$$T' = \partial_{\{p!d,p?d \mid p \in \{ab,ac\}\},\emptyset}(T).$$

II -

Motivation	Tool presentation ○○○○○●○	In the end
Other case studies		
_ _		

Demo

< ロ > < 回 > < 回 > < 回 > < 回 > <

æ

Eugen-Ioan Goriac (Reykjavik Uiversity) CALCO'11 PREG Axiomatizer

Motivation	Tool presentation	In the end
	000000	
Other case studies		

The *reentrant server* operation
$$!_{-}$$
 is defined by $\frac{x \xrightarrow{a} x'}{|x \xrightarrow{a} x' \parallel |x}$.

In this case a pair of infinite rewriting axioms is derived:

$$!x = !'(x, x)$$

 $!'(a.x', x) = a.(x' || !x).$

: _

This problem occurs only in the case of operations for which a positive variable appears in the target.

Motivation

PREG Axiomatizer:

- works for operations given in a restricted format, extending the finite trees with predicates system
 - however, it covers most of the operators in the literature
- generates confluent axiomatizations, but only weakly normalizing
 - however, there is a class of systems (linear and syntactically well-founded) for which it is strongly normalizing
- PREG Axiomatizer handles:
 - format checking,
 - implicit predicates for trees (*a.t* terminates if *t* terminates).

A⊒ ▶ < ∃

otivation	Tool presentation	In the end
0000000		00

PREG Axiomatizer:

- works for operations given in a restricted format, extending the finite trees with predicates system
 - however, it covers most of the operators in the literature
- generates confluent axiomatizations, but only weakly normalizing
 - however, there is a class of systems (linear and syntactically well-founded) for which it is strongly normalizing
- PREG Axiomatizer handles:
 - format checking,
 - implicit predicates for trees (*a.t* terminates if *t* terminates).

otivation	Tool presentation	In the end
		•0

PREG Axiomatizer:

- works for operations given in a restricted format, extending the finite trees with predicates system
 - however, it covers most of the operators in the literature
- generates confluent axiomatizations, but only weakly normalizing
 - however, there is a class of systems (linear and syntactically well-founded) for which it is strongly normalizing

PREG Axiomatizer handles:

- format checking,
- implicit predicates for trees (*a.t* terminates if *t* terminates).

otivation	Tool presentation	In the end
		••

PREG Axiomatizer:

- works for operations given in a restricted format, extending the finite trees with predicates system
 - however, it covers most of the operators in the literature
- generates confluent axiomatizations, but only weakly normalizing
 - however, there is a class of systems (linear and syntactically well-founded) for which it is strongly normalizing
- PREG Axiomatizer handles:
 - format checking,
 - implicit predicates for trees (*a.t* terminates if *t* terminates).

Motivation	Tool presentation	In the end
		00

Future work

Ways to extend and improve the prototype:

- integration with external provers and checkers,
- format checking (operator properties),
- recursively defined terms, open terms,
- universal predicates,
- detect infinite rewriting axiomatizations,
- better user interface,
- . . .