Efficient Algorithm for pre* on Interprecedural Parallel Flow Graph

Yutaka Nagashima
Technische Universität München

Efficient Algorithm for pre* on Interprecedural Parallel Flow Graph

Parallel programming is difficult!

We need a good technique of Software Engineering.

Efficient Algorithm for pre* on Interprecedural Parallel Flow Graph

Parallel programming is difficult!

We need a good technique of Software Engineering.

V&V Validation: Are you building the right thing?
Verification: Are you building it right?

$$\mathrm{pre}^*\,(L) = \{t \mid t \xrightarrow{*} t' \text{ for some } t' \in L\}$$

Efficient Algorithm for pre* on Interprecedural Parallel Flow Graph

Parallel programming is difficult!

We need a good technique of Software Engineering.

V&V Validation: Are you building the right thing?
Verification: Are you building it right?

$$\mathrm{pre}^*(L) = \{t \mid t \xrightarrow{*} t' \text{ for some } t' \in L\}$$

State-space explosion:
The size of transition system representations grows exponentially to the number of variables or to the number of components in a concurrent system.

Outline

- Introduction $\rightarrow$ Done!
- Program
- Parallel flow graph system
- PA-declaration Δ
- Construct input Automaton $\mathcal{A}$
- Epsilon-closed input automaton $\tilde{\mathcal{A}}$
- Declarative Part: Defining PA
- Operational Part:
  - Saturate PA: PA $\rightarrow$ SatPA
  - Reduce PA: SatPA $\rightarrow$ RedPA

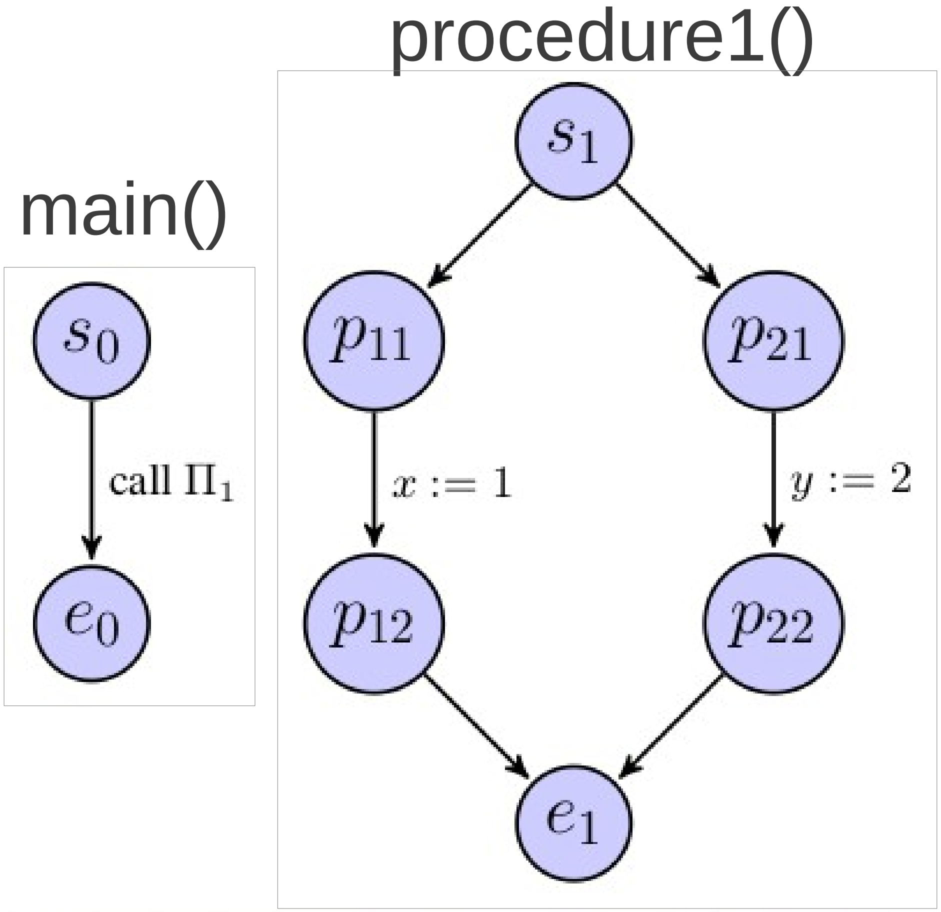
Parallel Flow Graph System (FGS)

main() { call procedure1;}

procedure1 $\{x := 1; \| \ y := 2; \}$

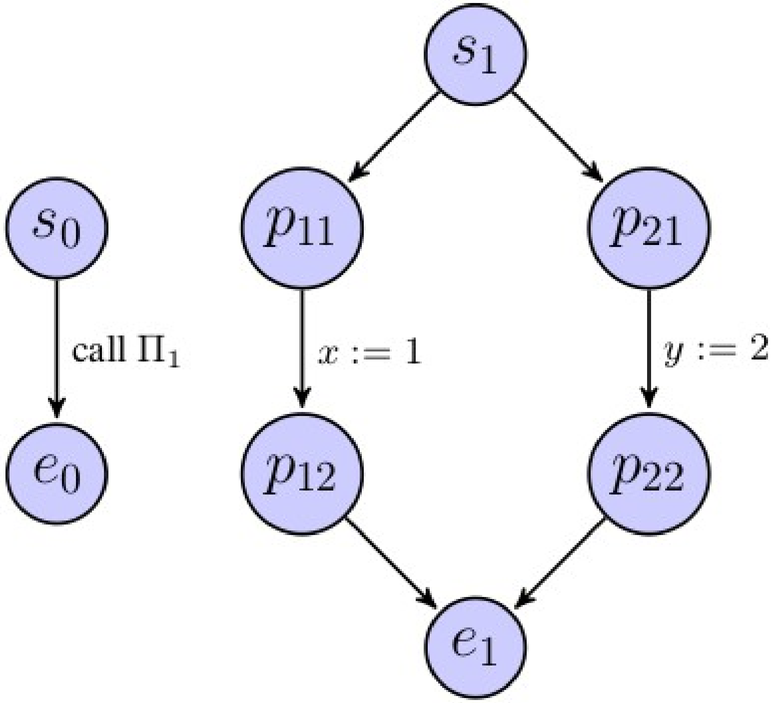
Parallel Flow Graph System (FGS)

main() { call procedure1;}
procedure1 $\{x := 1; \| \; y := 2; \}$



From parallel FGS to PA-declaration Δ

Process algebra:
action-labelled transitions between states denoted by PA-terms

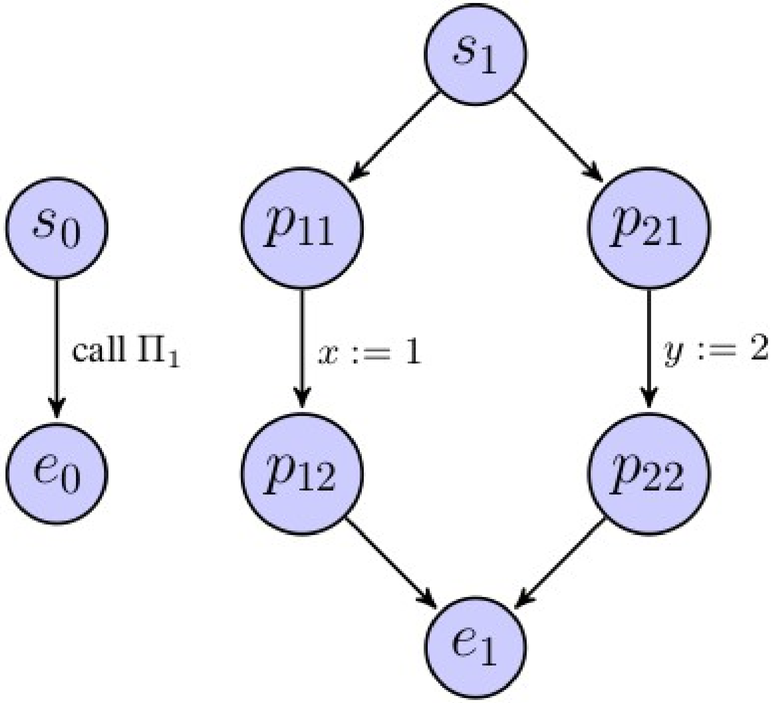


From parallel FGS to PA-declaration Δ

$$X \xrightarrow{a} t$$

X: process constant
a: action
t: PA-term

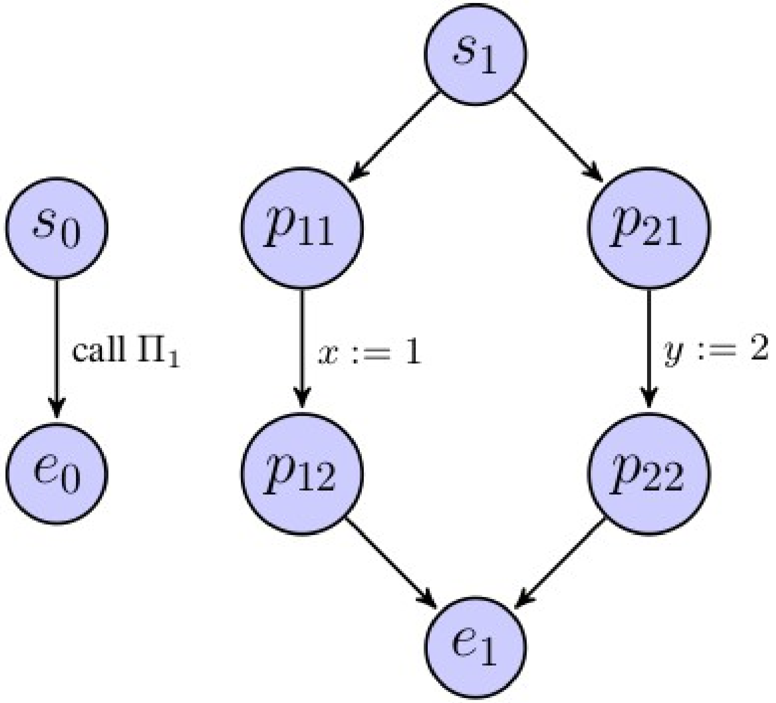
Process algebra:
action-labelled transitions between states denoted by PA-terms



From parallel FGS to PA-declaration Δ

$$X \xrightarrow{a} t$$

X: process constant
a: action
t: PA-term



Process algebra:
action-labelled transitions between states denoted by PA-terms

PA-term (T_A)

$$t = X \mid t_\varepsilon \mid (t_1 \bullet t_2) \mid (t_1 \parallel t_2)$$

epsilon term (IsNil)

$$t_\varepsilon = \varepsilon \mid (t_{\varepsilon_1} \bullet t_{\varepsilon_2}) \mid (t_{\varepsilon_1} \parallel t_{\varepsilon_2})$$

empty process: successful termination of process

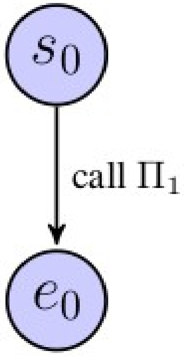
From parallel FGS to PA-declaration Δ

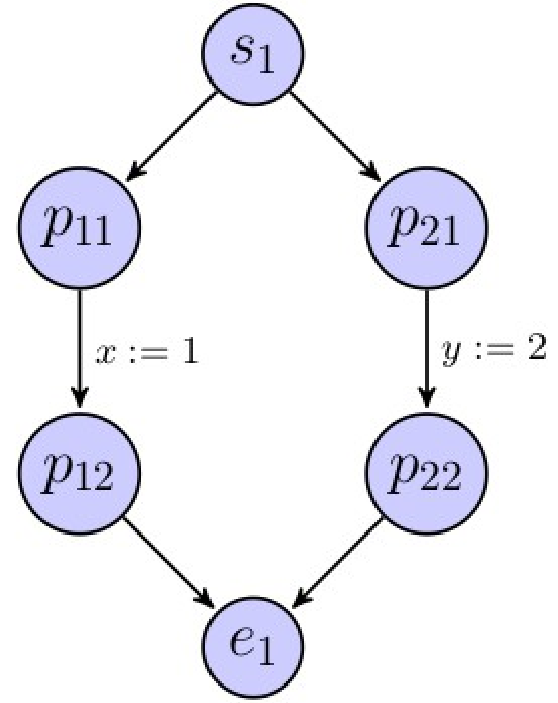
in parallel FGS	in PA-declaration
for $n \longrightarrow m$	$N \longrightarrow M$
for $n \xrightarrow{v:=t} m$	$N \xrightarrow{v:=t} M$
for $n \xrightarrow{call\Pi_i(T)} m$	$N \longrightarrow START_i \bullet M$
for end node of procedure Π_i	$END_i \longrightarrow \varepsilon$
for $n \longrightarrow \{m_1, m_2\}$	$N \longrightarrow K \bullet M$, $K \longrightarrow \{M_1 \parallel M_2\}$
for $\{m_1', m_2'\} \longrightarrow m$	$M_1' \longrightarrow \varepsilon$, $M_2' \longrightarrow \varepsilon$

$$X \xrightarrow{a} t$$

X: process constant
a: action
t: PA-term

Process algebra:
action-labelled transitions between states denoted by PA-terms





PA-term (TA)

$$t = X \mid t_\varepsilon \mid (t_1 \bullet t_2) \mid (t_1 \parallel t_2)$$

epsilon term (IsNil)

$$t_\varepsilon = \varepsilon \mid (t_{\varepsilon_1} \bullet t_{\varepsilon_2}) \mid (t_{\varepsilon_1} \parallel t_{\varepsilon_2})$$

empty process: successful termination of process

From PA-declaration to input automaton $\mathcal{A}$

Possible execution of the program

$$
\begin{aligned}
& S_0 \\
& \rightarrow S_1 \bullet E_0 \\
& \rightarrow (K \bullet E_1) \bullet E_0 \\
& \rightarrow ((P_{11}||(P_{21}) \bullet E_1) \bullet E_0 \\
& \xrightarrow{y:=2} ((P_{11}||P_{22}) \bullet E_1) \bullet E_0 \\
& \xrightarrow{x:=1} ((P_{12}||P_{22}) \bullet E_1) \bullet E_0 \\
& \rightarrow ((P_{12}||\varepsilon) \bullet E_1) \bullet E_0 \\
& \rightarrow \boxed{((\varepsilon||\varepsilon) \bullet E_1) \bullet E_0} \\
& \rightarrow ((\varepsilon||\varepsilon) \bullet \varepsilon) \bullet E_0 \\
& \rightarrow ((\varepsilon||\varepsilon) \bullet \varepsilon) \bullet \varepsilon
\end{aligned}
$$

From PA-declaration to input automaton $\mathcal{A}$

Possible execution of the program

$$
\begin{aligned}
& S_0 \\
& \to S_1 \bullet E_0 \\
& \to (K \bullet E_1) \bullet E_0 \\
& \to ((P_{11} || (P_{21}) \bullet E_1) \bullet E_0 \\
& \xrightarrow{y:=2} ((P_{11} || P_{22}) \bullet E_1) \bullet E_0 \\
& \xrightarrow{x:=1} ((P_{12} || P_{22}) \bullet E_1) \bullet E_0 \\
& \to ((P_{12} || \varepsilon) \bullet E_1) \bullet E_0 \\
& \to \boxed{((\varepsilon || \varepsilon) \bullet E_1) \bullet E_0} \\
& \to ((\varepsilon || \varepsilon) \bullet \varepsilon) \bullet E_0 \\
& \to ((\varepsilon || \varepsilon) \bullet \varepsilon) \bullet \varepsilon
\end{aligned}
$$

Epsilon-closed language

$$t \in L \iff t_\varepsilon \bullet t \in L \iff t_\varepsilon \parallel t \in L \iff t \parallel t_\varepsilon \in L$$

Epsilon-closure

$$\tilde{L} := \bigcap \{M \supseteq L | M \text{ is } \varepsilon\text{-closed}\}$$

From PA-declaration to input automaton $\mathcal{A}$

Possible execution of the program

$$
\begin{aligned}
& S_0 \\
\rightarrow\ & S_1 \bullet E_0 \\
\rightarrow\ & (K \bullet E_1) \bullet E_0 \\
\rightarrow\ & ((P_{11} || (P_{21}) \bullet E_1) \bullet E_0 \\
\xrightarrow{y:=2}\ & ((P_{11} || P_{22}) \bullet E_1) \bullet E_0 \\
\xrightarrow{x:=1}\ & ((P_{12} || P_{22}) \bullet E_1) \bullet E_0 \\
\rightarrow\ & ((P_{12} || \varepsilon) \bullet E_1) \bullet E_0 \\
\rightarrow\ & \boxed{((\varepsilon || \varepsilon) \bullet E_1) \bullet E_0} \\
\rightarrow\ & ((\varepsilon || \varepsilon) \bullet \varepsilon) \bullet E_0 \\
\rightarrow\ & ((\varepsilon || \varepsilon) \bullet \varepsilon) \bullet \varepsilon
\end{aligned}
$$

$\mathcal{A}$

$$
\begin{aligned}
q_i(\chi) &\Leftarrow true \\
q_i(x \bullet y) &\Leftarrow q_j(x) \wedge q_k(y) \\
q_i(x \bullet y) &\Leftarrow q_i(x) \\
q_i(x \parallel y) &\Leftarrow q_j(x) \wedge q_k(y) \\
q_i(x \parallel y) &\Leftarrow q_i(x) \\
q_i(x \parallel y) &\Leftarrow q_i(y)
\end{aligned}
$$

Epsilon-closed language

$$t \in L \iff t_\varepsilon \bullet t \in L \iff t_\varepsilon \parallel t \in L \iff t \parallel t_\varepsilon \in L$$

Epsilon-closure

$$\tilde{L} := \bigcap \{ M \supseteq L | M \text{ is } \varepsilon\text{-closed} \}$$

Epsilon-closed input automata

$$\mathcal{A} \longrightarrow \tilde{\mathcal{A}}$$

$$\mathcal{A}$$

$$q_i(\chi) \Leftarrow true$$

$$q_i(x \bullet y) \Leftarrow q_j(x) \wedge q_k(y)$$

$$q_i(x \bullet y) \Leftarrow q_i(x)$$

$$q_i(x \parallel y) \Leftarrow q_j(x) \wedge q_k(y)$$

$$q_i(x \parallel y) \Leftarrow q_i(x)$$

$$q_i(x \parallel y) \Leftarrow q_i(y)$$

Epsilon-closed input automata

$$\mathcal{A} \longrightarrow \tilde{\mathcal{A}}$$

$\mathcal{A}$

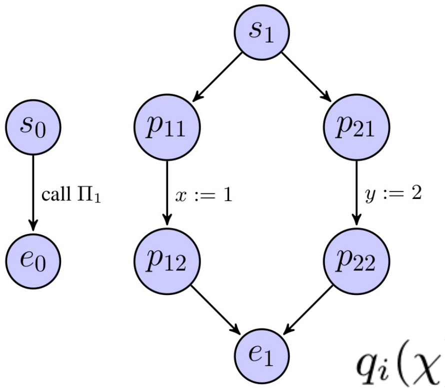
$$q_i(\chi) \Leftarrow true$$
$$q_i(x \bullet y) \Leftarrow q_j(x) \wedge q_k(y)$$
$$q_i(x \bullet y) \Leftarrow q_i(x)$$
$$q_i(x \parallel y) \Leftarrow q_j(x) \wedge q_k(y)$$
$$q_i(x \parallel y) \Leftarrow q_i(x)$$
$$q_i(x \parallel y) \Leftarrow q_i(y)$$

$\tilde{\mathcal{A}}$

$$q_\varepsilon(\varepsilon) \Leftarrow true$$
$$q_i(x \bullet y) \Leftarrow q_\varepsilon(x) \wedge q_i(y)$$
$$q_i(x \parallel y) \Leftarrow q_\varepsilon(x) \wedge q_i(y)$$
$$q_i(x \parallel y) \Leftarrow q_i(x) \wedge q_\varepsilon(y)$$

Where are we...?

main() { call procedure1;}
procedure1 $\{x := 1; \parallel y := 2; \}$



$$q_i(\chi) \Leftarrow true$$
$$q_i(x \bullet y) \Leftarrow q_j(x) \wedge q_k(y)$$
$$q_i(x \bullet y) \Leftarrow q_i(x)$$
$$q_i(x \parallel y) \Leftarrow q_j(x) \wedge q_k(y)$$
$$q_i(x \parallel y) \Leftarrow q_i(x)$$
$$q_i(x \parallel y) \Leftarrow q_i(y)$$

$$\mathcal{A}$$

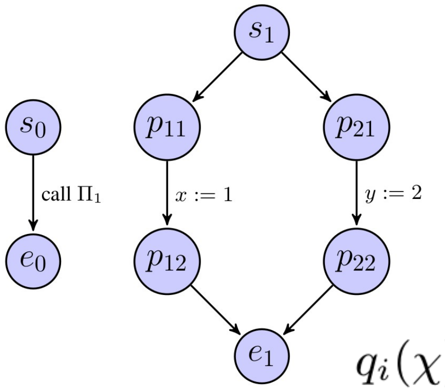
$$q_\varepsilon(\varepsilon) \Leftarrow true$$
$$q_i(x \bullet y) \Leftarrow q_\varepsilon(x) \wedge q_i(y)$$
$$q_i(x \parallel y) \Leftarrow q_\varepsilon(x) \wedge q_i(y)$$
$$q_i(x \parallel y) \Leftarrow q_i(x) \wedge q_\varepsilon(y)$$

Here!

$$\tilde{\mathcal{A}}$$

Where are we...?

main() { call procedure1;}
procedure1 $\{x := 1; \parallel y := 2; \}$



$$q_i(\chi) \Leftarrow true$$
$$q_i(x \bullet y) \Leftarrow q_j(x) \wedge q_k(y)$$
$$q_i(x \bullet y) \Leftarrow q_i(x)$$
$$q_i(x \parallel y) \Leftarrow q_j(x) \wedge q_k(y)$$
$$q_i(x \parallel y) \Leftarrow q_i(x)$$
$$q_i(x \parallel y) \Leftarrow q_i(y)$$

RedPA

An automaton that accepts pre*(L)

pre* $(L) = \{t | t \xrightarrow{*} t' \text{ for some } t' \in L\}$

$\mathcal{A}$

SatPA

$$q_\varepsilon(\varepsilon) \Leftarrow true$$
$$q_i(x \bullet y) \Leftarrow q_\varepsilon(x) \wedge q_i(y)$$
$$q_i(x \parallel y) \Leftarrow q_\varepsilon(x) \wedge q_i(y)$$
$$q_i(x \parallel y) \Leftarrow q_i(x) \wedge q_\varepsilon(y)$$

Here!

PA

$\tilde{\mathcal{A}}$

The declarative part: define P_A

$$t \in \mathrm{pre}^*(L) \iff P_A \models p_0(t)$$

$$P_A = \text{clauses of } \tilde{\mathcal{A}} \cup \text{clauses in the table}$$

The declarative part: define PA

$$t \in \mathrm{pre}^*(L) \iff P_A \models p_0(t)$$

$$P_A = \text{clauses of } \tilde{\mathcal{A}} \cup \text{clauses in the table}$$

$p_i(\chi) \Leftarrow q_i(\chi)$	
	for each $\chi \in \{\text{process constants of} \Delta\} \cup \{\varepsilon\}$
$p_i(X) \Leftarrow q_i(X)$	
	for each $(X \xrightarrow{a} t) \in \Delta$
$p_i(x_1 \bullet x_2) \Leftarrow p_j(x_1) \wedge q_k(x_2)$	
	for each $(q_i(x \bullet y) \Leftarrow q_j(x) \wedge q_k(y)) \in \tilde{\mathcal{A}}$
$p_i(x_1 \bullet x_2) \Leftarrow p_i(x_1)$	
	for each $(q_i(x \bullet y) \Leftarrow q_i(x)) \in \tilde{\mathcal{A}}$
$p_i(x_1 \bullet x_2) \Leftarrow p_\varepsilon(x_1) \wedge p_i(x_2)$	
$p_i(x_1 \parallel x_2) \Leftarrow p_j(x_1) \wedge q_k(x_2)$	
	for each $(q_i(x \parallel y) \Leftarrow q_j(x) \wedge q_k(y)) \in \tilde{\mathcal{A}}$
$p_i(x_1 \parallel x_2) \Leftarrow p_i(x_1)$	
	for each $(q_i(x \parallel y) \Leftarrow q_i(x)) \in \tilde{\mathcal{A}}$
$p_i(x_1 \parallel x_2) \Leftarrow p_i(x_2)$	
	for each $(q_i(x \parallel y) \Leftarrow q_i(y)) \in \tilde{\mathcal{A}}$

Theory

$$\mathrm{pre}^*(L_{q_i}) = \{t \in T_{PA} \mid P_A \models p_i(t)\}$$

Proposition:
The sets pre*(Lqi) are the smallest sets such that the following holds:

1. If $\chi \in L_{q_i}$, then $\chi \in \mathrm{pre}^*(L_{q_i})$.
2. If $((X \xrightarrow{a} t) \in \Delta)$ and $(t \in \mathrm{pre}^*(L_{q_i}))$, then $X \in \mathrm{pre}^*(L_{q_i})$.
3. If $((q_i(x \bullet y) \Leftarrow q_j(x) \wedge q_k(x)) \in \tilde{\mathcal{A}})$ and $(t_1 \in \mathrm{pre}^*(L_{q_j}))$ and $(t_2 \in L_{q_k})$, then $t_1 \bullet t_2 \in \mathrm{pre}^*(L_{q_i})$.
4. If $((q_i(x \bullet y) \Leftarrow q_i(x)) \in \tilde{\mathcal{A}})$ and $(t_1 \in \mathrm{pre}^*(L_{q_i}))$, then $(t_1 \bullet t_2) \in \mathrm{pre}^*(L_{q_i})$ for $t_2 \in T_{PA}$.
5. If $(t_1 \in \mathrm{pre}^*(\mathrm{IsNil}))$ and $(t_2 \in \mathrm{pre}^*(L_{q_i}))$, then $(t_1 \bullet t_2) \in \mathrm{pre}^*(L_{q_i})$.
6. If $((q_i(x \parallel y) \Leftarrow q_j(x) \wedge q_k(x)) \in \tilde{\mathcal{A}})$ and $(t_1 \in \mathrm{pre}^*(L_{q_j}))$ and $(t_2 \in \mathrm{pre}^*(L_{q_k}))$, then $(t_1 \parallel t_2) \in \mathrm{pre}^*(L_{q_i})$.
7. If $((q_i(x \parallel y) \Leftarrow q_i(x) \in \tilde{\mathcal{A}})$ and $(t_1 \in \mathrm{pre}^*(L_{q_i}))$, then $(t_1 \parallel t_2) \in \mathrm{pre}^*(L_{q_i})$ for $t_2 \in T_{PA}$.
8. If $((q_i(x \parallel y) \Leftarrow q_i(y) \in \tilde{\mathcal{A}})$ and $(t_2 \in \mathrm{pre}^*(L_{q_i}))$, then $(t_1 \parallel t_2) \in \mathrm{pre}^*(L_{q_i})$ for $t_1 \in T_{PA}$.

The operational part: $P_A \to SatP_A \to RedP_A$

$$P_A$$

$$\downarrow$$

$$SatP_A = P_A \cup \{p(\chi) \mid P_A \models p(\chi)\}$$

$$\downarrow$$

$$RedP_A = \text{reduction clauses of } SatP_A$$

Selected references

- „Efficient Algorithms for pre* and post* on Interprocedural Parallel Flow Graphs“ Javier Esparza and Andreas Podelski

Questions?

Back up transition rules from SOS

$$\Delta \quad \frac{(X \xrightarrow{a} t) \in \Delta}{X \xrightarrow{a} t}$$

$$\text{seqential1} \quad \frac{t_1 \xrightarrow{a} t_1'}{t_1 \bullet t_2 \xrightarrow{a} t_1' \bullet t_2}$$

$$\text{seqential2} \quad \frac{(t_2 \xrightarrow{a} t_2') \wedge (t_1 \in \text{IsNil})}{(t_1 \bullet t_2 \xrightarrow{a} t_1 \bullet t_2')}$$

$$\text{parallel1} \quad \frac{(t_1 \xrightarrow{a} t_1')}{(t_1 \parallel t_2 \xrightarrow{a} t_1' \parallel t_2)}$$

$$\text{parallel2} \quad \frac{(t_2 \xrightarrow{a} t_2')}{(t_1 \parallel t_2 \xrightarrow{a} t_1 \parallel t_2')}$$

Q&A