A Generic Approach to the Static Analysis of concurrent recursive programs

Tayssir Touili

LIAFA - Paris 7

Joint work with Ahmed Bouajjani and Javier Esparza

Analysis of Recursive Programs: Pushdown Automata

```
function P(n:integer){
    if (n=0 ou n=1)
        {return 1}
    else{
          x:= Q(n)
          y:= R(x)
          if (x=y)
              {return P(n-1)}
          else
              {return P(n-2)}
           }
}
```

$P \rightarrow Q \cdot R \cdot P$

Infinite data types are abstracted into finite types

Stack: structure of the sequentiel calls of the different procedures

Analysis of Recursive Programs: Pushdown Automata

```
function P(n:integer){
   if (n=0 ou n=1)
      {return 1}
   else{
        x:= Q(n)
        y:= R(x)
        if (x=y)
           {return P(n-1)}
        else
           {return P(n-2)}
         }
}
```

P$\rightarrow$Q·R·P

Is some point T reachable?

Infinite data types are abstracted into finite types

Stack: structure of the sequentiel calls of the different procedures

Analysis of Recursive Programs: Pushdown Automata

```
function P(n:integer){
    if (n=0 ou n=1)
        {return 1}
    else{
        x:= Q(n)
        y:= R(x)
        if (x=y)
            {return P(n-1)}
        else
            {return P(n-2)}
        }
}
```

P$\longrightarrow$Q·R·P

Is some point T reachable?

$\Downarrow$

Reachability Analysis of Pushdown Automata

Infinite data types are abstracted into finite types

Stack: structure of the sequentiel calls of the different procedures

Analysis of Recursive Programs: Pushdown Automata

```
function P(n:integer){
   if (n=0 ou n=1)
      {return 1}
   else{
        x:= Q(n)
        y:= R(x)
        if (x=y)
           {return P(n-1)}
        else
           {return P(n-2)}
         }
}
```

P$\longrightarrow$Q·R·P

Is some point T reachable?

$\Downarrow$

Reachability Analysis of Pushdown Automata

[Büchi'64],[Caucal'92], [Bouajjani,Esparza,Maler'97], [Finkel,Willems,Wolper'97]...

Infinite data types are abstracted into finite types

Stack: structure of the sequentiel calls of the different procedures

What about Concurrency?

Synchronization $\Rightarrow$ Undecidability (synchronization of pushdown systems)

$\Rightarrow$ Approximate techniques

Existing works:

- Not very much!
- Approximate the effect of both sequentiality and concurrency
- No trade-off between precision and cost

Our work

- Extend the techniques on pushdown systems to the parallel case
- Approximate only the effect of synchronization
- Propose a general method which allows to perform analysis of different precision and cost

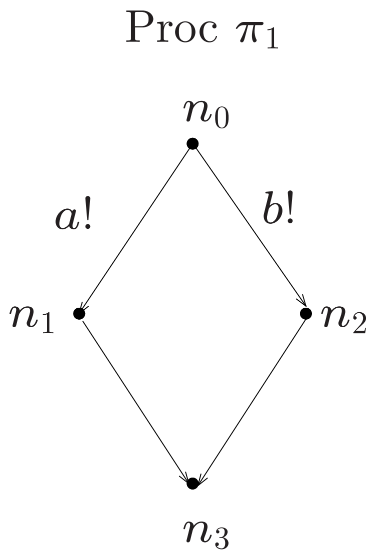
Concurrent Programs

- Concurrent program=n sequential components running in parallel
- Component=recursive program
- Communication through rendezvous
- undirectionnal point-to-point channels
- Communication through synchronization signals: $ch!$ and $ch?$

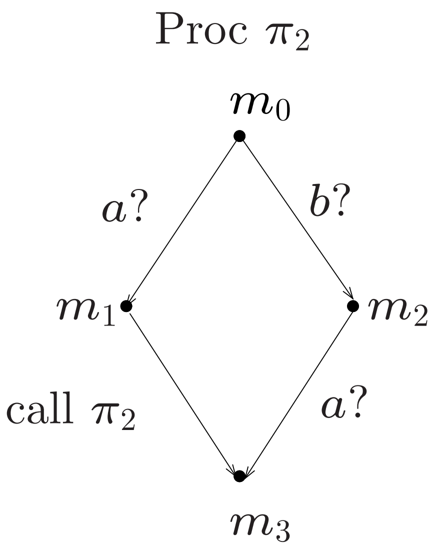
Flow Graphs

- Procedure=flow graph (graphical representation)
- Nodes in the flow graph=control points in the procedure
- Edges=statements
- Statements= assignements, calls to other procedures of the same sequential program, or communications with another sequential program

An Example



Component 1



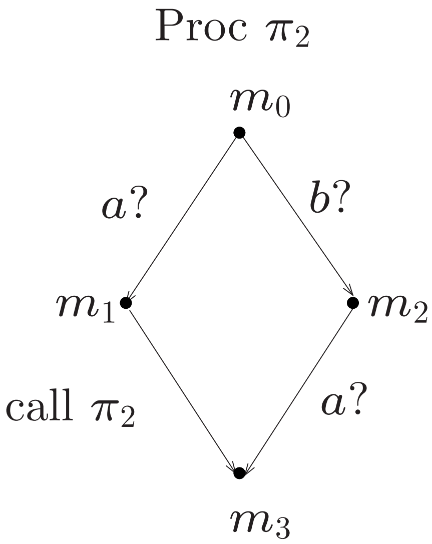
Component 2

Sequential Program=Pushdown System

Pushdown System $\mathcal{P} = (P, Act, \Gamma, c_0, \Delta)$:

- P is a finite set of control locations
- Act is a finite set of actions
- Γ is a finite stack alphabet
- c_0 is the initial configuration: a pair $\langle p, v \rangle$, $p \in P$, $v \in \Gamma^*$
- Δ is a finite set of rules of the form $\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', w \rangle$, $p, p' \in P, a \in Act, \gamma \in \Gamma, w \in \Gamma^*$
- Transition relation: $\langle p, \gamma u \rangle \xrightarrow{a} \langle p', wu \rangle$

From a Sequential Program to a Pushdown System



$$
\begin{array}{lcl}
r_1 & : & \langle p, m_0 \rangle \overset{a}{\hookrightarrow} \langle p, m_1 \rangle \\
r_2 & : & \langle p, m_0 \rangle \overset{b}{\hookrightarrow} \langle p, m_2 \rangle \\
r_3 & : & \langle p, m_1 \rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3 \rangle \\
r_4 & : & \langle p, m_2 \rangle \overset{a}{\hookrightarrow} \langle p, m_3 \rangle \\
r_5 & : & \langle p, m_3 \rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon \rangle
\end{array}
$$

Global variables: states of P

Local variables+nodes of the flow graph: symbols of Γ

Concurrent Program=Communicating Pushdown System

- Communicating Pushdown System: $(\mathcal{P}_1, \ldots, \mathcal{P}_n)$, $\mathcal{P}_i$ a pushdown system
- n concurrent sequential components= CPS$(\mathcal{P}_1, \ldots, \mathcal{P}_n)$
- A global configuration $g = (c_1, \ldots, c_n)$, c_i: a configuration of $\mathcal{P}_i$
- Transition relation: Synchronizing the pushdown systems

$$\begin{tikzcd}[column sep=small] g \arrow[d, "\tau"] & = (c_1, \ldots, c_{i-1}, & c_i \arrow[d, "\tau"] & , c_{i+1} \ldots, c_n) \\ g' & = (c_1, \ldots, c_{i-1}, & c'_i & , c_{i+1} \ldots, c_n) \end{tikzcd}$$

$$\begin{tikzcd}[column sep=small] g \arrow[d, "a"] & = (c_1, \ldots, & c_i \arrow[d, "a"] & , \ldots, & c_j \arrow[d, "a"] & , \ldots, c_n) \\ g' & = (c_1, \ldots, & c'_i & , \ldots, & c'_j & , \ldots, c_n) \end{tikzcd}$$

Reachability Analysis of CPDSs

Problem: $(c_1^0, c_2^0) \xrightarrow{*} (c_1, c_2)$ in $(\mathcal{P}_1, \mathcal{P}_2)$?

$$\Updownarrow$$

$$L(c_1^0, c_1) \cap L(c_2^0, c_2) \neq \emptyset$$

$L(c_i^0, c_i)$: paths of $\mathcal{P}_i$ leading from c_i^0 to c_i
: Context-free language

Undecidable!

$$\Downarrow$$

No exact solution

Our Approach: Abstraction

Problem: $L(c_1^0, c_1) \cap L(c_2^0, c_2) \stackrel{?}{=} \emptyset$

Idea: Compute abstractions A_i of $\big(L(c_i^0, c_i)\big)$:

- $L(c_i^0, c_i) \subseteq A_i$
- $A_1 \cap A_2 \stackrel{?}{\neq} \emptyset$ decidable

$\Rightarrow$ $A_1 \cap A_2 = \emptyset$ then $L(c_1^0, c_1) \cap L(c_2^0, c_2) = \emptyset$

- A_i finitely represented by an abstract object d_i

Our Approach: Abstraction

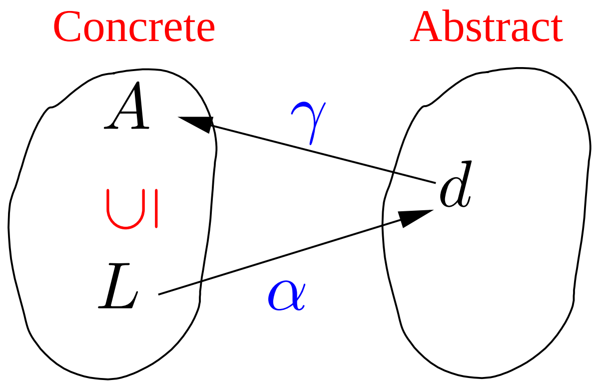
Problem: $L(c_1^0, c_1) \cap L(c_2^0, c_2) \stackrel{?}{=} \emptyset$

Idea: Compute abstractions A_i of $\big(L(c_i^0, c_i)\big)$:

- $L(c_i^0, c_i) \subseteq A_i$
- $A_1 \cap A_2 \stackrel{?}{\neq} \emptyset$ decidable

$\Rightarrow A_1 \cap A_2 = \emptyset$ then $L(c_1^0, c_1) \cap L(c_2^0, c_2) = \emptyset$

- A_i finitely represented by an abstract object d_i



(α, γ): a Galois connection

Abstract interpretation

First Abstraction

$\alpha_1(L)$ is the pair $[F, R]$, $F, R \subseteq Act$:

- F (for forbidden) is the set of actions that do not occur in any word in L
- R (for required) is the set of actions that occur in all words of L

Example: $Act = \{a, b, c\}$, $\alpha_1(ab^*) = [\{c\}, \{a\}]$

$\gamma_1([F, R])$= all words containing no letter of F and all letters of R

First Abstraction: Example

Procedure π_1

Procedure π_2

(n_1, m_2) not reachable: a required to reach n_1, but forbidden to reach m_2

Second Abstraction

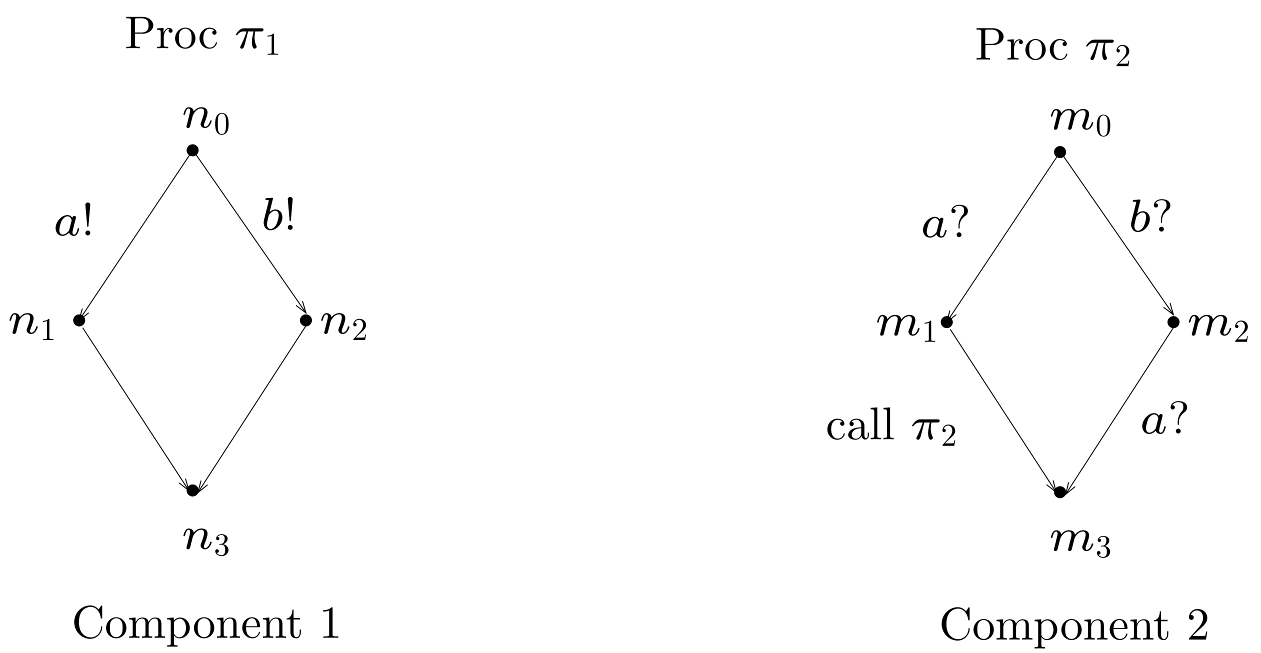
$\alpha_2(L) = \{al(w) \mid w \in L\}$

$al(w)$ is the set of letters that occur in w

Example: $\alpha_2(ab^*) = \{\{a\}, \{a, b\}\}$

$\gamma_2(\{al_1, \ldots, al_n\})=$ all words over the alphabets $al_1, \ldots, al_n$

Second Abstraction: Example



(n_3, m_3) not reachable:

$\alpha_2(L(n_0, n_3)) = \{\{a\}, \{b\}\}$, $\alpha_2(L(m_0, m_3)) = \{\{a, b\}\}$

Third Abstraction

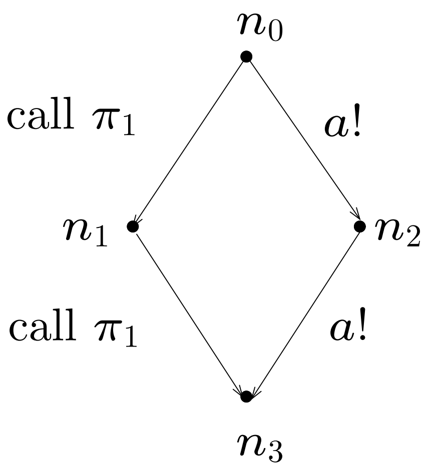
$\alpha_3(L) = \{Parikh(w) \mid w \in L\}$

$Parikh(w)$ is the Parikh image of w

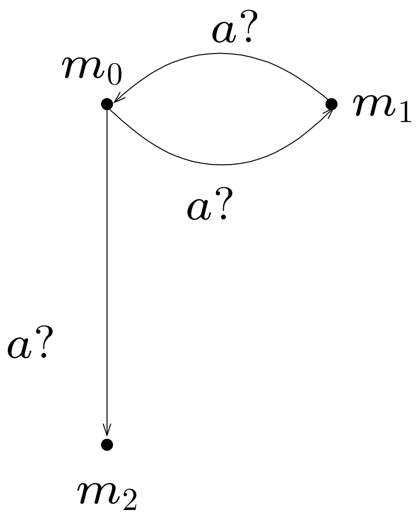
Example: $\alpha_3(ab^*) = \{(1, n) \mid n \geq 0\}$

$\gamma_3(L)$= all the permutations of the words of L

Third Abstraction: Example



Procedure π_1



Procedure π_2

(n_3, m_2) not reachable:

To reach n_3: a occurs an even number of times

To reach m_2: a occurs an odd number of times

The common feature?

- Commutativity: If $w \in A$, then $permutations(w) \subseteq A$

- Commutative Abstraction: $\gamma(\alpha(L))$ is a commutative language

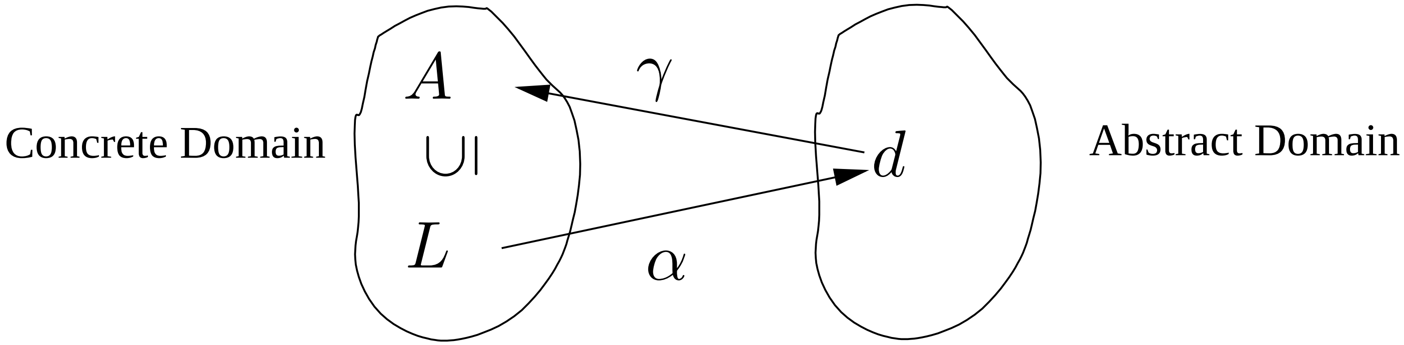
$\Rightarrow$ α forgets the order in which actions occur

- We provide a generic algorithm for computing

$$\alpha\big(L(I, F)\big)$$

when α: commutative abstraction

Our Formal Framework: Commutative Abstractions



2^{Act^*}	- - - - - - - - - - -	K
$\subseteq$	- - - - - - - - - - -	$\leq$
$\cup$	- - - - - - - - - - -	$\oplus$: associative and commutative
$\cap$	- - - - - - - - - - -	$\sqcap$
$\emptyset$	- - - - - - - - - - -	$\bar{0}$: neutral for $\oplus$
τ	- - - - - - - - - - -	$\bar{1}$: neutral for $\odot$
$\cdot$	- - - - - - - - - - -	$\odot$: associative and commutative
$*$	- - - - - - - - - - -	$\star : a^{\star} = \bigoplus_{n \geq 0} a^n$

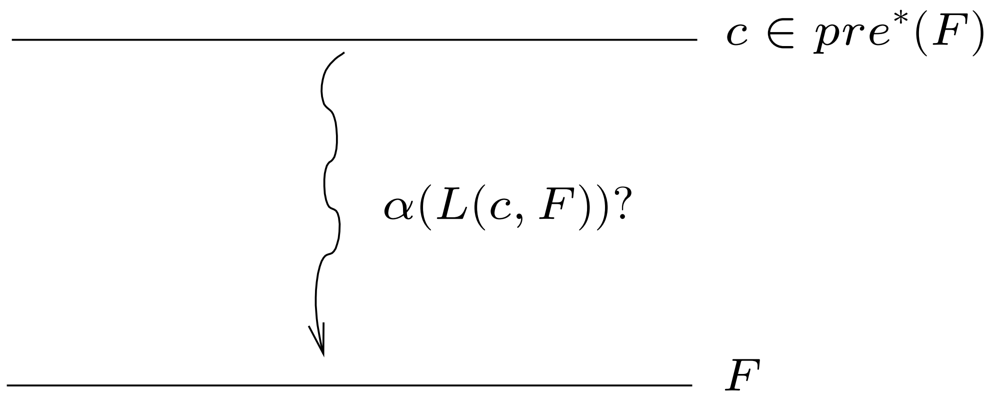
$(K, \oplus, \odot, \bar{0}, \bar{1})$: idempotent semiring $\quad$ $(K, \oplus, \odot, \star, \bar{0}, \bar{1})$: Kleene Algebra

Instances of Commutative Abstractions: Parikh Images

- $\leq = \subseteq$
- $\mathcal{L}_1 \oplus \mathcal{L}_2 = \mathcal{L}_1 \cup \mathcal{L}_2$
- $\mathcal{L}_1 \odot \mathcal{L}_2 = \{x + y \mid x \in \mathcal{L}_1, y \in \mathcal{L}_2\}$
- $\bar{1} = (0, \ldots, 0)$
- $\sqcap = \cap$

Computing $\alpha(L(I,F))$: The approach

- Compute $pre^*(F)$: Finite automaton $\mathcal{A}$
- Tag the transitions of $\mathcal{A}$ $\Rightarrow$ Compute $\alpha(L(c,F))$ for each $c \in pre^*(F)$



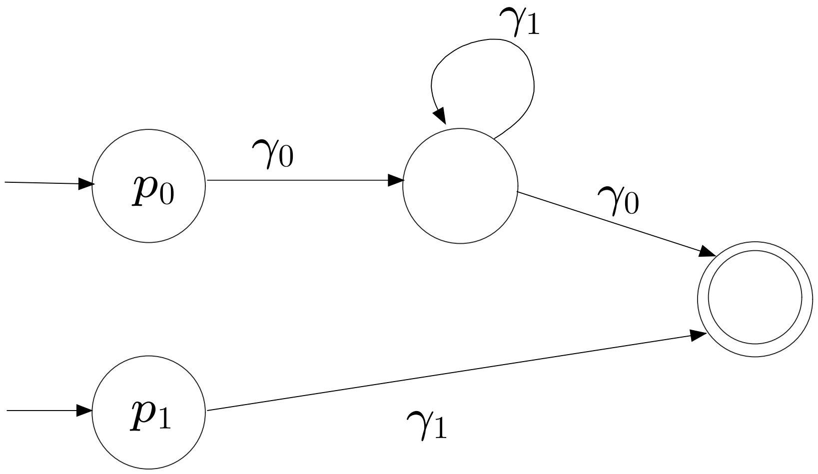
- Return $\alpha(L(I,F))$

Multi-automata

[Bouajjani, Esparza, Maler'97]

Represent regular sets of configurations of PDS

Example: The set $\langle p_0, \gamma_0\gamma_1^*\gamma_0\rangle \cup \langle p_1, \gamma_1\rangle$ is represented by:



First step: $pre^*(F)$

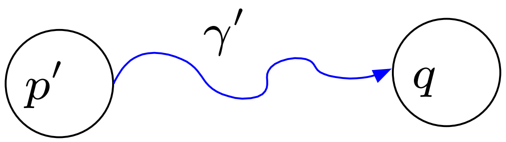
Theorem [Bouajjani, Esparza, Maler'97]

F regular $\Rightarrow pre^*(F)$ can be effectively representable by a multi-automaton

Technique: saturation algorithm: $\mathcal{A}_0 \subset \mathcal{A}_1 \subset \cdots$

$$\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', \gamma' \rangle$$

$\mathcal{A}_i$



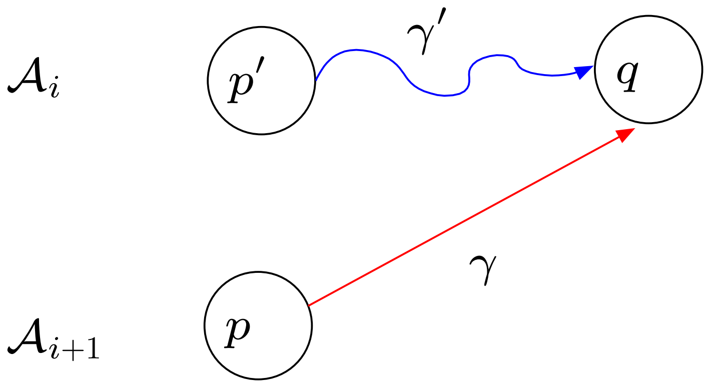
First step: $pre^*(F)$

Theorem [Bouajjani, Esparza, Maler'97]

F regular $\Rightarrow pre^*(F)$ can be effectively representable by a multi-automaton

Technique: saturation algorithm $\mathcal{A}_0 \subset \mathcal{A}_1 \subset \cdots$

$$\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', \gamma' \rangle$$



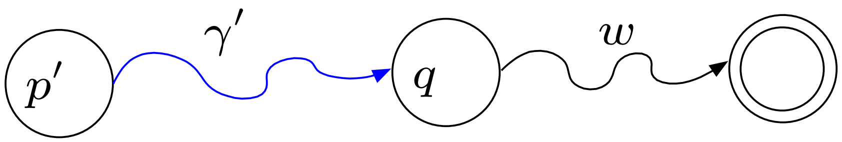
First step: $pre^*(F)$

Theorem [Bouajjani, Esparza, Maler'97]

F regular $\Rightarrow$ $pre^*(F)$ can be effectively representable by a multi-automaton

Technique: saturation algorithm $\mathcal{A}_0 \subset \mathcal{A}_1 \subset \cdots$

$$\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', \gamma' \rangle$$



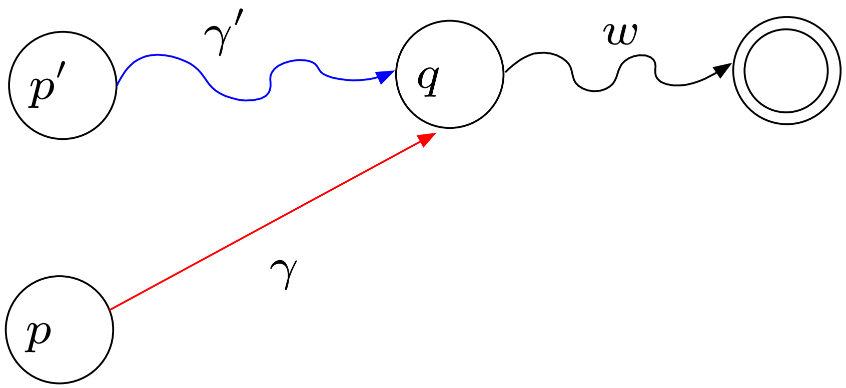
First step: $pre^*(F)$

Theorem [Bouajjani, Esparza, Maler'97]

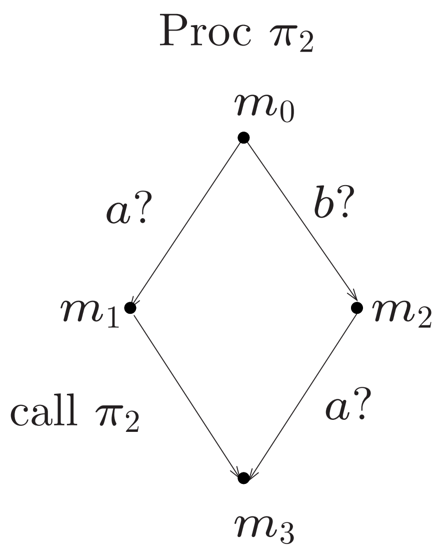
F regular $\Rightarrow pre^*(F)$ can be effectively representable by a multi-automaton

Technique: saturation algorithm $\mathcal{A}_0 \subset \mathcal{A}_1 \subset \cdots$

$$\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', \gamma' \rangle$$



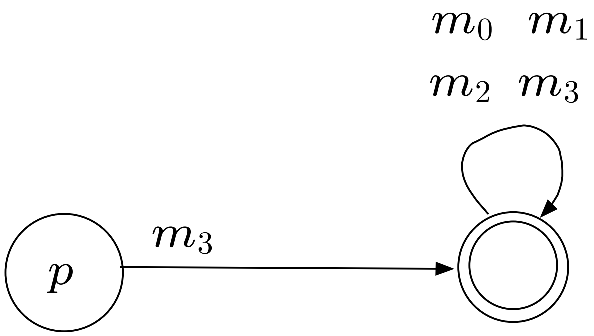
Example



$$
\begin{array}{lcl}
r_1 & : & \langle p, m_0\rangle \overset{a}{\hookrightarrow} \langle p, m_1\rangle \\
r_2 & : & \langle p, m_0\rangle \overset{b}{\hookrightarrow} \langle p, m_2\rangle \\
r_3 & : & \langle p, m_1\rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3\rangle \\
r_4 & : & \langle p, m_2\rangle \overset{a}{\hookrightarrow} \langle p, m_3\rangle \\
r_5 & : & \langle p, m_3\rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon\rangle
\end{array}
$$

Example

$$
\begin{aligned}
r_1 &: \langle p, m_0\rangle \xhookrightarrow{a} \langle p, m_1\rangle\\
r_2 &: \langle p, m_0\rangle \xhookrightarrow{b} \langle p, m_2\rangle\\
r_3 &: \langle p, m_1\rangle \xhookrightarrow{\tau} \langle p, m_0 m_3\rangle\\
r_4 &: \langle p, m_2\rangle \xhookrightarrow{a} \langle p, m_3\rangle\\
r_5 &: \langle p, m_3\rangle \xhookrightarrow{\tau} \langle p, \epsilon\rangle
\end{aligned}
$$



Example

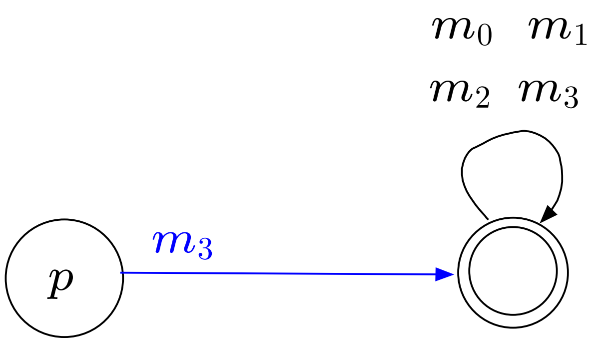
$$r_1 \quad : \quad \langle p, m_0 \rangle \xhookrightarrow{a} \langle p, m_1 \rangle$$

$$r_2 \quad : \quad \langle p, m_0 \rangle \xhookrightarrow{b} \langle p, m_2 \rangle$$

$$r_3 \quad : \quad \langle p, m_1 \rangle \xhookrightarrow{\tau} \langle p, m_0 m_3 \rangle$$

$$r_4 \quad : \quad {\color{blue}\langle p, m_2 \rangle \xhookrightarrow{a} \langle p, m_3 \rangle}$$

$$r_5 \quad : \quad \langle p, m_3 \rangle \xhookrightarrow{\tau} \langle p, \epsilon \rangle$$



Example

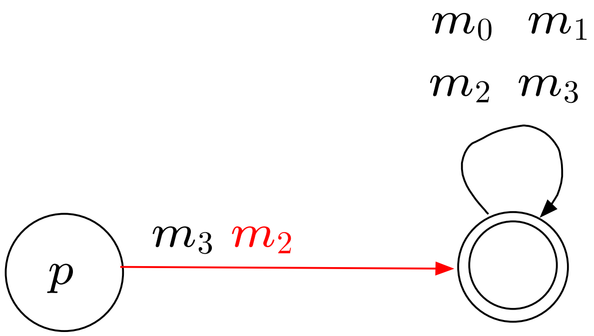
$$r_1 \quad : \quad \langle p, m_0 \rangle \overset{a}{\hookrightarrow} \langle p, m_1 \rangle$$

$$r_2 \quad : \quad \langle p, m_0 \rangle \overset{b}{\hookrightarrow} \langle p, m_2 \rangle$$

$$r_3 \quad : \quad \langle p, m_1 \rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3 \rangle$$

$$r_4 \quad : \quad \langle p, m_2 \rangle \overset{a}{\hookrightarrow} \langle p, m_3 \rangle$$

$$r_5 \quad : \quad \langle p, m_3 \rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon \rangle$$



Example

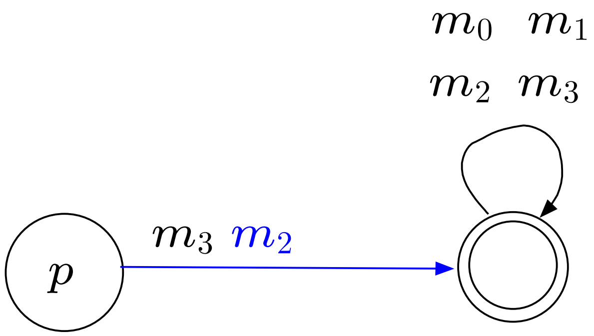
$$r_1 \quad : \quad \langle p, m_0 \rangle \overset{a}{\hookrightarrow} \langle p, m_1 \rangle$$

$$r_2 \quad : \quad \langle p, m_0 \rangle \overset{b}{\hookrightarrow} \langle p, m_2 \rangle$$

$$r_3 \quad : \quad \langle p, m_1 \rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3 \rangle$$

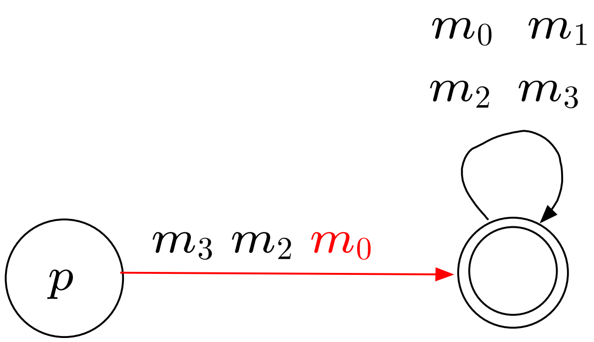
$$r_4 \quad : \quad \langle p, m_2 \rangle \overset{a}{\hookrightarrow} \langle p, m_3 \rangle$$

$$r_5 \quad : \quad \langle p, m_3 \rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon \rangle$$



Example

$$
\begin{aligned}
r_1 &: \langle p, m_0\rangle \overset{a}{\hookrightarrow} \langle p, m_1\rangle \\
r_2 &: \langle p, m_0\rangle \overset{b}{\hookrightarrow} \langle p, m_2\rangle \\
r_3 &: \langle p, m_1\rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3\rangle \\
r_4 &: \langle p, m_2\rangle \overset{a}{\hookrightarrow} \langle p, m_3\rangle \\
r_5 &: \langle p, m_3\rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon\rangle
\end{aligned}
$$



Example

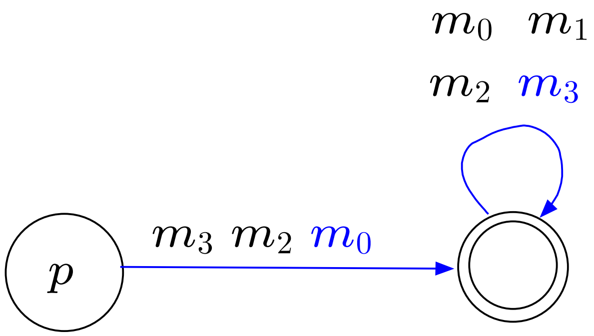
$$r_1 \quad : \quad \langle p, m_0 \rangle \overset{a}{\hookrightarrow} \langle p, m_1 \rangle$$

$$r_2 \quad : \quad \langle p, m_0 \rangle \overset{b}{\hookrightarrow} \langle p, m_2 \rangle$$

$$r_3 \quad : \quad \langle p, m_1 \rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3 \rangle$$

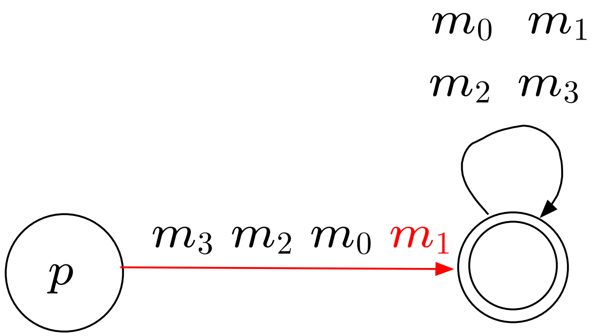
$$r_4 \quad : \quad \langle p, m_2 \rangle \overset{a}{\hookrightarrow} \langle p, m_3 \rangle$$

$$r_5 \quad : \quad \langle p, m_3 \rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon \rangle$$



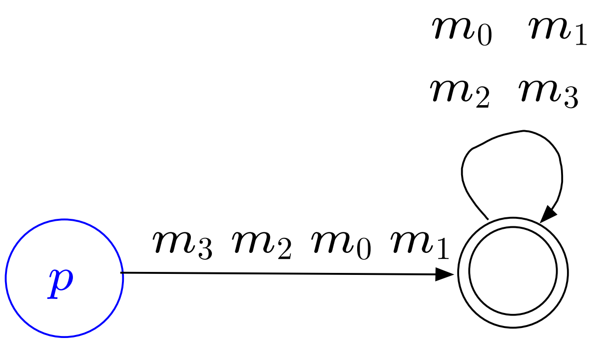
Example

$$
\begin{aligned}
r_1 &: \langle p, m_0 \rangle \xhookrightarrow{a} \langle p, m_1 \rangle \\
r_2 &: \langle p, m_0 \rangle \xhookrightarrow{b} \langle p, m_2 \rangle \\
r_3 &: {\color{blue}\langle p, m_1 \rangle \xhookrightarrow{\tau} \langle p, m_0 m_3 \rangle} \\
r_4 &: \langle p, m_2 \rangle \xhookrightarrow{a} \langle p, m_3 \rangle \\
r_5 &: \langle p, m_3 \rangle \xhookrightarrow{\tau} \langle p, \epsilon \rangle
\end{aligned}
$$



Example

$$
\begin{aligned}
r_1 &: \langle p, m_0 \rangle \overset{a}{\hookrightarrow} \langle p, m_1 \rangle \\
r_2 &: \langle p, m_0 \rangle \overset{b}{\hookrightarrow} \langle p, m_2 \rangle \\
r_3 &: \langle p, m_1 \rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3 \rangle \\
r_4 &: \langle p, m_2 \rangle \overset{a}{\hookrightarrow} \langle p, m_3 \rangle \\
r_5 &: \langle p, m_3 \rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon \rangle
\end{aligned}
$$



Example

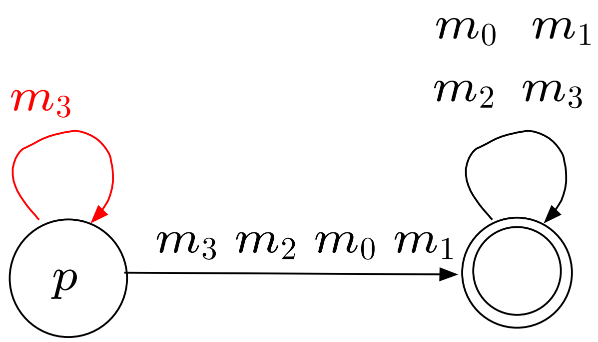
$$r_1 \quad : \quad \langle p, m_0 \rangle \overset{a}{\hookrightarrow} \langle p, m_1 \rangle$$

$$r_2 \quad : \quad \langle p, m_0 \rangle \overset{b}{\hookrightarrow} \langle p, m_2 \rangle$$

$$r_3 \quad : \quad \langle p, m_1 \rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3 \rangle$$

$$r_4 \quad : \quad \langle p, m_2 \rangle \overset{a}{\hookrightarrow} \langle p, m_3 \rangle$$

$$r_5 \quad : \quad \langle p, m_3 \rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon \rangle$$



Example

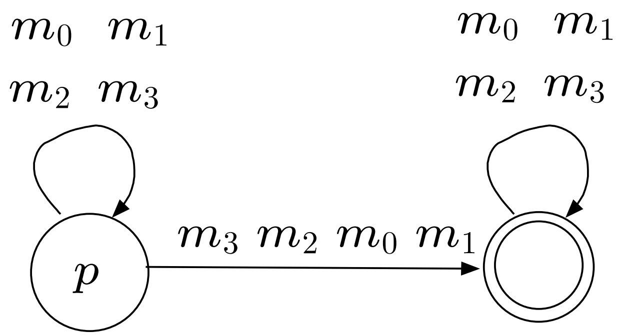
$$r_1 \quad : \quad \langle p, m_0 \rangle \overset{a}{\hookrightarrow} \langle p, m_1 \rangle$$

$$r_2 \quad : \quad \langle p, m_0 \rangle \overset{b}{\hookrightarrow} \langle p, m_2 \rangle$$

$$r_3 \quad : \quad \langle p, m_1 \rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3 \rangle$$

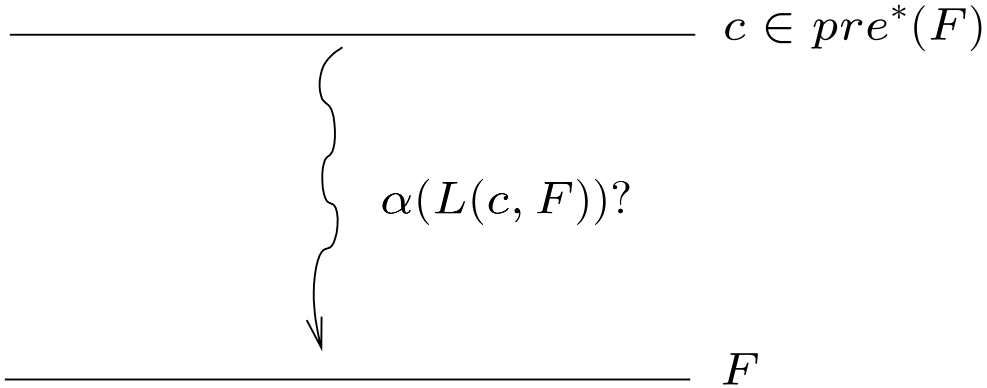
$$r_4 \quad : \quad \langle p, m_2 \rangle \overset{a}{\hookrightarrow} \langle p, m_3 \rangle$$

$$r_5 \quad : \quad \langle p, m_3 \rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon \rangle$$



Computing $\alpha\big(L(I,F)\big)$

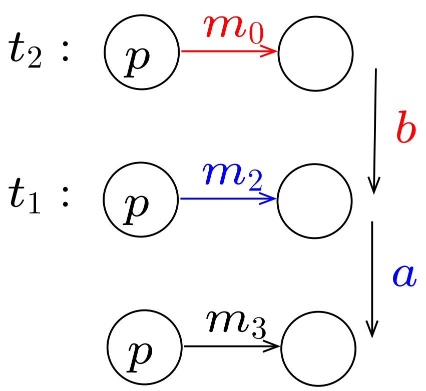
- Compute $pre^*(F)$: Finite automaton $\mathcal{A}$ $\surd$
- Tag the transitions of $\mathcal{A}$ $\Rightarrow$ Compute $\alpha(L(c,F))$ for each $c \in pre^*(F)$



- Return $\alpha(L(I,F))$

Computing $\alpha\big(L(c, F)\big)$

$L(t)$: the sequences of actions (rules) that can be considered to create the transition t in $\mathcal{A}(pre^*(F))$ from the transitions of $\mathcal{A}(F)$



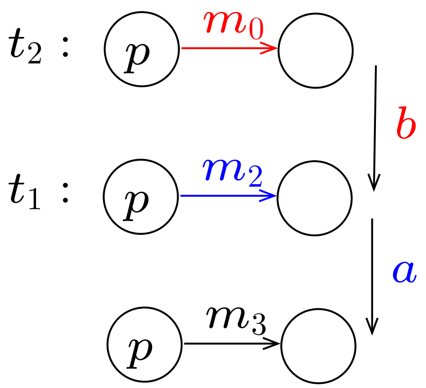
$$\langle p, m_0\rangle \overset{b}{\hookrightarrow} \langle p, m_2\rangle$$

$$\langle p, m_2\rangle \overset{a}{\hookrightarrow} \langle p, m_3\rangle$$

$a \in L(t_1), ba \in L(t_2)$

Computing $\alpha\big(L(c,F)\big)$

$L(t)$: the sequences of actions (rules) that can be considered to create the transition t in $\mathcal{A}(pre^*(F))$ from the transitions of $\mathcal{A}(F)$

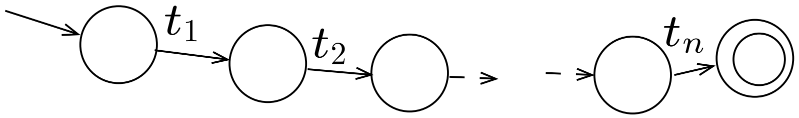


$\langle p, m_0\rangle \overset{b}{\hookrightarrow} \langle p, m_2\rangle$

$\langle p, m_2\rangle \overset{a}{\hookrightarrow} \langle p, m_3\rangle$

$a \in L(t_1), ba \in L(t_2)$

ρ:

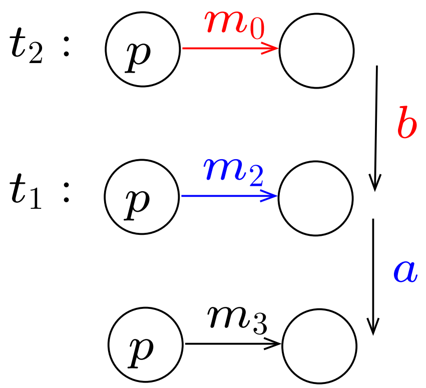


$c \in pre^*(F)$

$$\alpha(L(c,F)) = \bigoplus_{\rho} \alpha(L(t_1)) \odot \alpha(L(t_2)) \odot \cdots \odot \alpha(L(t_n))$$

Computing $\alpha\big(L(c,F)\big)$

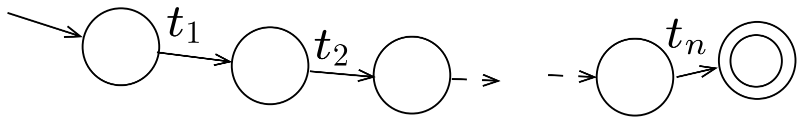
$L(t)$: the sequences of actions (rules) that can be considered to create the transition t in $\mathcal{A}(pre^*(F))$ from the transitions of $\mathcal{A}(F)$



$\langle p, m_0\rangle \overset{b}{\hookrightarrow} \langle p, m_2\rangle$

$\langle p, m_2\rangle \overset{a}{\hookrightarrow} \langle p, m_3\rangle$

$a \in L(t_1),\ ba \in L(t_2)$

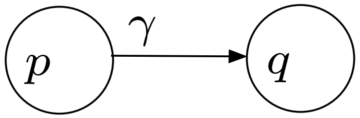
ρ: 

$c \in pre^*(F)$

$$\alpha(L(c,F)) = \bigoplus_{\rho} \alpha(L(t_1)) \odot \alpha(L(t_2)) \odot \cdots \odot \alpha(L(t_n))$$

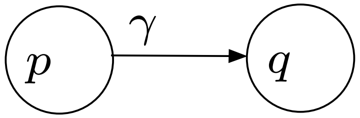
Problem: Compute $\alpha(L(t))$

Computing $\alpha\big(L(t)\big)$

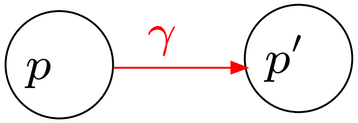


$$\alpha(\tau) \leq \alpha\big(L(p \xrightarrow{\gamma} q)\big)$$

Computing $\alpha\big(L(t)\big)$



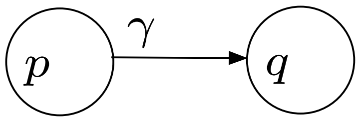
$$\alpha(\tau) \leq \alpha\big(L(p \xrightarrow{\gamma} q)\big)$$



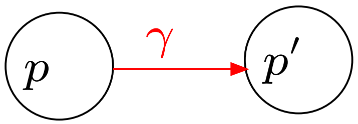
$$\langle p, \gamma\rangle \overset{a}{\hookrightarrow} \langle p', \epsilon\rangle$$

$$\alpha(a) \leq \alpha\big(L(p \xrightarrow{\gamma} p')\big)$$

Computing $\alpha\big(L(t)\big)$

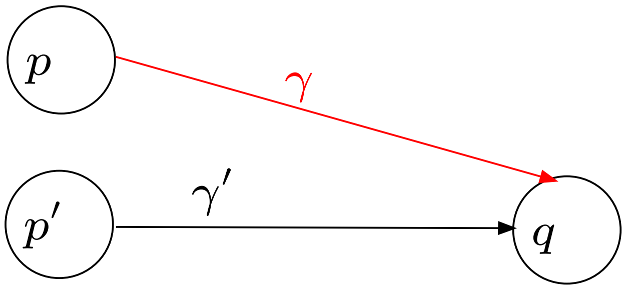


$$\alpha(\tau) \le \alpha\big(L(p \xrightarrow{\gamma} q)\big)$$



$$\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', \epsilon \rangle$$

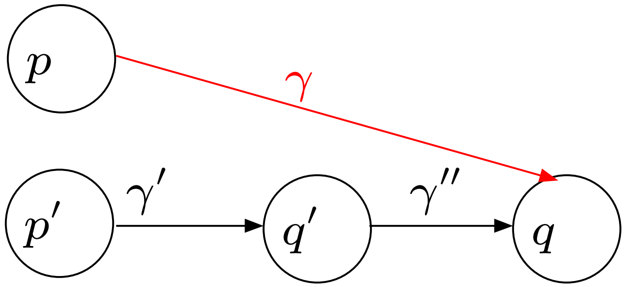
$$\alpha(a) \le \alpha\big(L(p \xrightarrow{\gamma} p')\big)$$



$$\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', \gamma' \rangle$$

$$\alpha(a) \odot \alpha\big(L(p' \xrightarrow{\gamma'} q)\big) \le \alpha\big(L(p \xrightarrow{\gamma} q)\big)$$

Computing $\alpha\big(L(t)\big)$



$$\langle p, \gamma \rangle \overset{a}{\hookrightarrow} \langle p', \gamma'\gamma'' \rangle$$

$$\alpha\big(a\big) \odot \alpha\big(L(p' \xrightarrow{\gamma'} q')\big) \odot \alpha\big(L(q' \xrightarrow{\gamma''} q)\big) \leq \alpha\big(L(p \xrightarrow{\gamma} q)\big)$$

Computing $\alpha\big(L(t)\big)\cdots$

Let $d_i = \alpha(L(t_i)) \Rightarrow$ we get equations of the form

$$f_i(d_1, \ldots, d_n) \leq d_i, 1 \leq i \leq n$$

f_i: polynomial constructed out of $d_1, \ldots, d_n, \odot$, and $\oplus$

$(d_1, \ldots, d_n)$ is the least solution of

$$f_i(x_1, \ldots, x_n) \leq x_i, 1 \leq i \leq n$$

Example

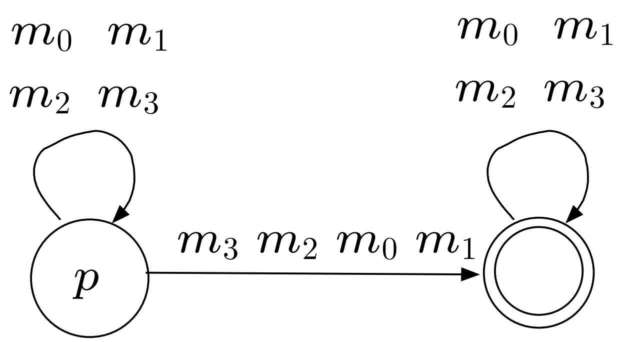
$$r_1 \quad : \quad \langle p, m_0 \rangle \overset{a}{\hookrightarrow} \langle p, m_1 \rangle$$

$$r_2 \quad : \quad \langle p, m_0 \rangle \overset{b}{\hookrightarrow} \langle p, m_2 \rangle$$

$$r_3 \quad : \quad \langle p, m_1 \rangle \overset{\tau}{\hookrightarrow} \langle p, m_0 m_3 \rangle$$

$$r_4 \quad : \quad \langle p, m_2 \rangle \overset{a}{\hookrightarrow} \langle p, m_3 \rangle$$

$$r_5 \quad : \quad \langle p, m_3 \rangle \overset{\tau}{\hookrightarrow} \langle p, \epsilon \rangle$$



Equations of our example

$$
\begin{aligned}
(x_4 \odot x_9) \oplus (x_8 \odot x_3) &\leq x_1 \\
(\alpha(a) \odot x_1) \oplus (\alpha(b) \odot x_2) &\leq x_4 \\
(\alpha(b) \odot x_7) \oplus (\alpha(a) \odot x_5) &\leq x_8 \\
\alpha(\tau) &\leq x_3 \\
\alpha(a) \odot x_3 &\leq x_2 \\
\alpha(a) \odot x_6 &\leq x_7 \\
x_8 \odot x_6 &\leq x_5 \\
\alpha(\tau) &\leq x_6 \\
\alpha(\tau) &\leq x_9
\end{aligned}
$$

How to solve these equations?

The solution by Hopkins and Kozen in commutative Kleene algebras

The least solution of

$$f_i(x_1, \ldots, x_n) \leq x_i, 1 \leq i \leq n$$

is the fixpoint of the chain

$$\mathbf{d_0} \leq \mathbf{d_1} \leq \cdots$$

given by

$$\mathbf{d_0} = \mathbf{f}(\bar{\mathbf{0}})$$
$$\mathbf{d_{k+1}} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}(\mathbf{d_k})^\star \odot \mathbf{d_k}$$

The Jacobian $\frac{\partial \mathbf{f}}{\partial x}$

- $(\frac{\partial \mathbf{f}}{\partial x})_{i,j} = \frac{\partial f_i}{\partial x_j}$
- $\frac{\partial x_i}{\partial x_i} = \bar{1}$, $\frac{\partial x_j}{\partial x_i} = \bar{0}$, for $i \neq j$, and $\frac{\partial a}{\partial x_i} = \bar{0}$, for $a \in K$.
- $\frac{\partial}{\partial x_i}(f \oplus g) = \frac{\partial f}{\partial x_i} \oplus \frac{\partial g}{\partial x_i}$
- $\frac{\partial}{\partial x_i}(f \odot g) = (f \odot \frac{\partial g}{\partial x_i}) \oplus (\frac{\partial f}{\partial x_i} \odot g)$
- $\frac{\partial}{\partial x_i}(f^\star) = (f^\star \odot \frac{\partial f}{\partial x_i})$

The solution of our example

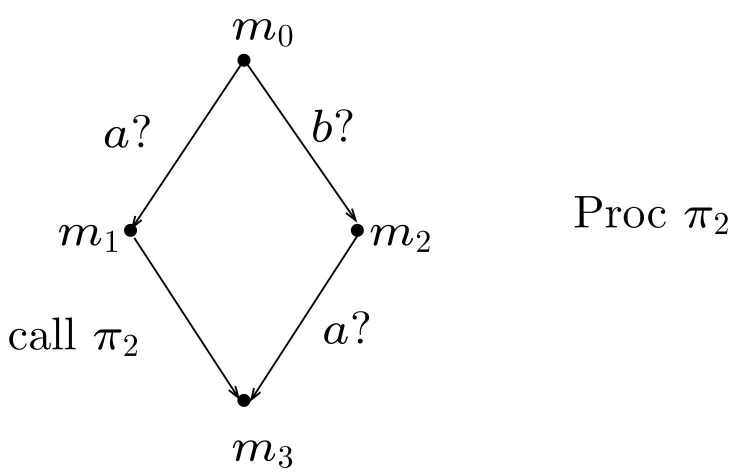
$$\begin{aligned}
d_1 = d_5 = d_8 &= \alpha(a)^\star \odot \alpha(a) \odot \alpha(b) \\
d_3 = d_6 &= d_9 = \alpha(\tau) \\
d_2 = d_7 &= \alpha(a) \\
d_4 &= (\alpha(a) \odot \alpha(a)^\star \odot \alpha(a) \odot \alpha(b)) \oplus (\alpha(a) \odot \alpha(b))
\end{aligned}$$

$$\alpha(L(m_0, m_3\Gamma^*)) = d_4$$

Different instantiations

- $\alpha_1(a)=[\{b\},\{a\}]$, $\alpha_1(b)=[\{a\},\{b\}]$, $\alpha_1(\tau)=[Act,\emptyset]$
- $\alpha_2(a)=\{\{a\}\}$, $\alpha_2(b)=\{\{b\}\}$, $\alpha_2(\tau)=\{\{\emptyset\}\}$
- $\alpha_3(a)=\{(1,0)\}$, $\alpha_3(b)=\{(0,1)\}$, $\alpha_3(\tau)=\{(0,0)\}$
- $\alpha(L(m_0, m_3\Gamma^*)) = (\alpha(a) \odot \alpha(a)^\star \odot \alpha(a) \odot \alpha(b)) \oplus (\alpha(a) \odot \alpha(b))$

Analysis of the example



- $\alpha_1(L(m_0, m_3\Gamma^*)) = [\emptyset, \{a, b\}]$ (a and b are required to reach m_3)
- $\alpha_2(L(m_0, m_3\Gamma^*)) = \{\{a, b\}\}$ ($\{a, b\}$ only possible alphabet to reach m_3)
- $\alpha_3(L(m_0, m_3\Gamma^*)) = \{(k, 1) \mid k \geq 1\}$ (an arbitrary number (≥ 1) of a's, and exactly one b)

Complexity

- $\mathcal{A}_{pre^*}$: $\mathcal{O}(|Q||\Delta| + |\delta|)$ transitions [Esparza et al'00]

- Complexity of the fixpoint algorithm is $\mathcal{O}(|\Delta|^3 \cdot t \cdot c)$:

  – t is the number of iterations of the fixpoint algorithm
  – c is the maximal cost of an $\oplus, \odot, \star$ operation

  Forbidden and required sets: $\mathcal{O}(Act|\Delta|^3)$

  Alphabets: $2^{\mathcal{O}(Act)}|\Delta|^3$

  Parikh images (worst case): double exponential in $|\Delta|$

Conclusion

- Extension of the pushdown approach to the concurrent case
- Gradual, possibility to trade efficiency for precision
- Generic algorithm which can be implemented
- Future work: Extension to PA