Model Checking Infinite State Spaces

Javier Esparza

Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh

Model Checking

An approach to the verification problem which formalises

system	satisfies	property
	as	
Kripke structure	is model of	temporal formula

Other possibilities are

characteristic temporal formula	implies	temporal formula
Kripke structure	is simulated by	most general Kripke structure

Nothing in the essence of the approach requires the Kripke structure to be finite

Actually, Kripke structures for real systems are very often infinite

The finiteness constraint is due to our current technology, not to the approach itself

Sources of infinity

Data manipulation: unbounded counters, integer variables, lists . . .

Control structures: procedures , process creation . . .

Asynchronous communication: unbounded FIFO queues

Parameters: number of processes, of principals, of input gates, delays, . . .

Real-time: discrete or dense domains

A bit of history

- Late 80s, early 90s: First theoretical papers

  Decidability/Undecidability results for Place/Transition Petri nets

  Efficient model-checking algorithms for context-free processes

  Region construction for timed automata

- 90s: Research program

  1. Decidability analysis
  2. Design of algorithms or semi-algorithms
  3. Design of implementations
  4. Tools
  5. Applications

- Late 90s, 00s: General techniques emerge

  Automata-theoretic approach to model-checking

  Symbolic reachability

  Accelerations

Programme

The automata-theoretic approach

Symbolic search: forward and backward

Case study: broadcast protocols

Accelerations

Case study: pushdown systems

Widenings

The automata-theoretic approach

Safety property ϕ	$\Longrightarrow$	Automaton $\mathcal{A}_{\neg\phi}$	$\Longrightarrow$	$\mathcal{L}(\neg\phi)$
Liveness property ϕ	$\Longrightarrow$	Büchi automaton $\mathcal{B}_{\neg\phi}$	$\Longrightarrow$	$\mathcal{L}_\omega(\neg\phi)$
System S	$\Longrightarrow$	Kripke structure $\mathcal{K}_S$	$\Longrightarrow$	$\mathcal{L}(S)$, $\mathcal{L}_\omega(S)$

Safety: $S \models \phi$ iff $\mathcal{L}(\mathcal{K}_S \times \mathcal{A}_{\neg\phi}) = \emptyset$

Liveness: $S \models \phi$ iff $\mathcal{L}_\omega(\mathcal{K}_S \times \mathcal{B}_{\neg\phi}) = \emptyset$

Closure under product with automata:

for every S and $\mathcal{A}$ there is a system $S \otimes \mathcal{A}$ such that $\mathcal{L}(S \otimes \mathcal{A}) = \mathcal{L}(\mathcal{K}_S \times \mathcal{A})$

Closure under product with Büchi automata:

for every S and $\mathcal{B}$ there is a system $S \otimes \mathcal{B}$ such that $\mathcal{L}_\omega(S \otimes \mathcal{B}) = \mathcal{L}_\omega(\mathcal{K}_S \times \mathcal{B})$

For system classes closed under product, model checking reducible to

- Reachability

  Given: system S, sets I and F of initial and final configurations of $\mathcal{K}$
  To decide: if F can be reached from I,
  i.e., if there exist $i \in I$ and $f \in F$ such that $i \longrightarrow^* f$

- Repeated reachability

  Given: System S, sets I and F of initial and final configurations of S
  To decide: if F can be repeatedly reached from I,
  i.e. if there exist $i \in I$ and $f_1, f_2, \ldots \in F$ such that $i \longrightarrow^* f_1 \longrightarrow^* f_2 \cdots$

I and F are usually infinite

Symbolic search

A general framework for the reachability problem

Let C denote a (possibly infinite) set of configurations

Forward search

$post(C)$ = immediate successors of C

Initialize $C := I$

Iterate $C := C \cup post(C)$ until

$C \cap F \neq \emptyset$; return "reachable", or

a fixpoint is reached; return "non-reachable"

Backward search

$pre(C)$ = immediate predecessors of C

Initialize $C := F$

Iterate $C := C \cup pre(C)$ until

$C \cap I \neq \emptyset$; return "reachable", or

a fixpoint is reached; return "non-reachable"

Problem: when are the procedures effective?

Backward search effective if ...

1. each $C \in \mathcal{C}$ has a symbolic finite representation
2. $F \in \mathcal{C}$
3. if $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$ (and effectively computable)
4. emptyness of $C \cap I$ is decidable
5. $C_1 = C_2$ is decidable (to check if fixpoint has been reached)
6. any chain $C_1 \subseteq C_2 \subseteq C_3 \ldots$ reaches a fixpoint after finitely many steps

(1) - (5) guarantee partial correctness, (6) guarantees termination

For forward search replace $pre(C)$ by $post(C)$ and exchange I and F

Shape of I determined by system, shape of F by specification

Parametrized protocols

Defined for *n* processes.

Correctness: the desired properties hold for every *n*

Processes modelled as communicating finite automata

For each value of *n* the system has a finite state space (only one source of infinity)

Turing powerful, and so further restrictions sensible:

Broadcast Protocols

Broadcast protocols

Introduced by Emerson and Namjoshi in LICS ’98

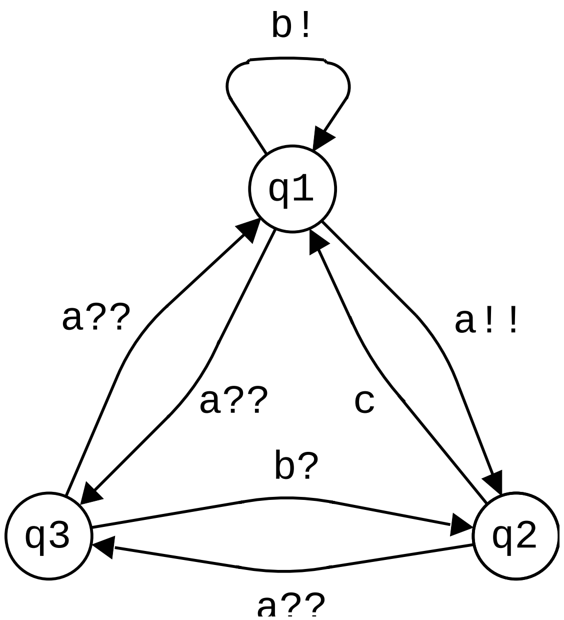
All processes execute the same algorithm, i.e., all finite automata are identical

Processes are undistinguishable (no IDs)

Communication mechanisms:

- Rendezvous: two processes exchange a message and move to new states
- Broadcasts: a process sends a message to all others
  all processes move to new states

Syntax



a!! : broadcast a message along (channel) *a*
a?? : receive a broadcasted message along *a*
b! : send a message to one process along *b*
b? : receive a message from one process along *b*
c : change state without communicating with anybody

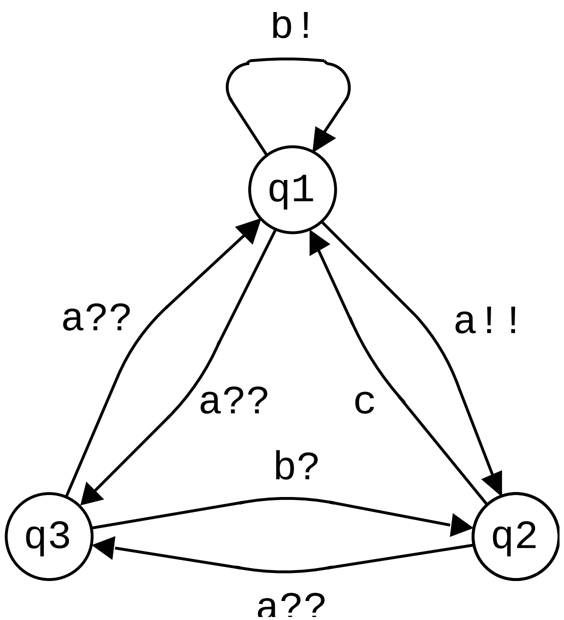
Semantics

$\boxed{\text{The global state of a broadcast protocol is completely determined by the number of processes in each state.}}$

Configuration: mapping $c: Q \longrightarrow \mathbb{N}$

represented by the vector $(c(q_1), \ldots, c(q_n))$

Semantics for an initial configuration: finite transition system with configurations as nodes



$(3,1,2)$	$\longrightarrow$	$(4,0,2)$	(silent move *c*)
$(3,1,2)$	$\longrightarrow$	$(3,2,1)$	(rendezvous *b*)
$(3,1,2)$	$\longrightarrow$	$(2,1,3)$	(broadcast *a*)
$(185,3425,17)$	$\longrightarrow$	$(17,1,3609)$	(broadcast *a*)

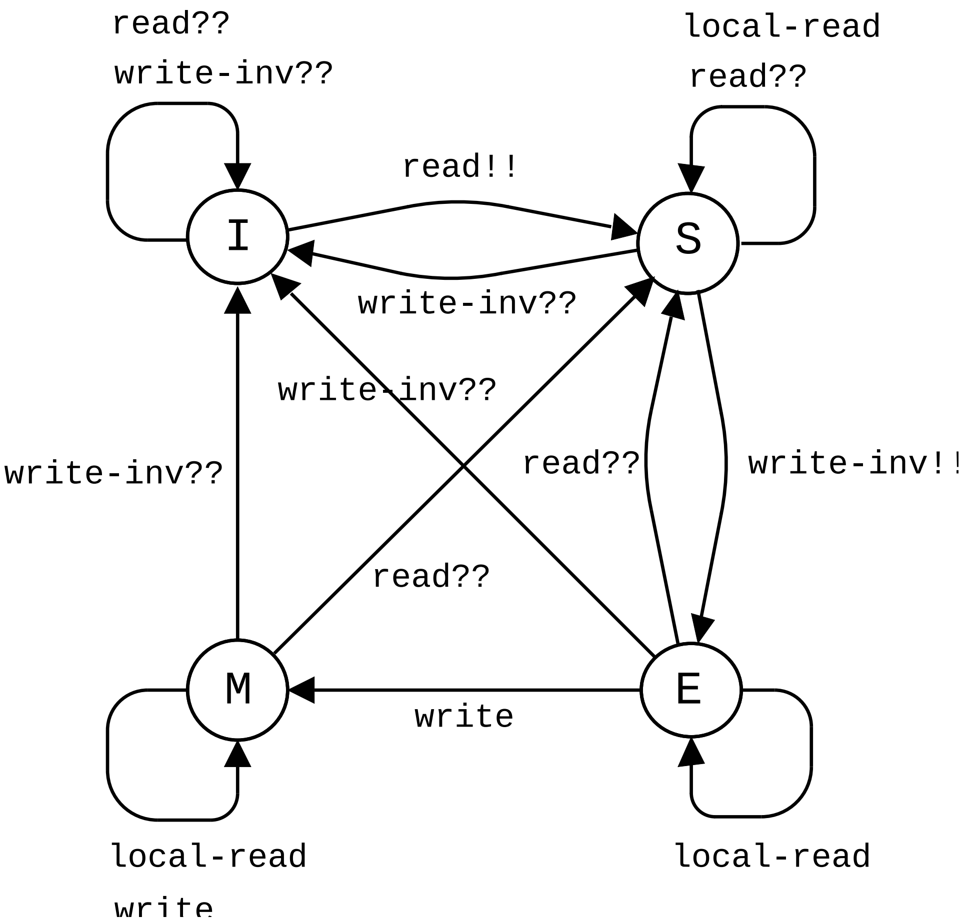
Parametrized configuration: partial mapping $p : Q \longrightarrow \mathbb{N}$

- Intuition: “configuration with holes”
- Formally: set of configurations (total mappings matching p)

Infinite transition system (Kripke structure) of the broadcast protocol:

- Fix an initial parametrized configuration p_0.
- Take the union of all finite transition systems $\mathcal{K}_c$ for each configuration $c \in p_0$.

A MESI-protocol



Reachability in broadcast protocols

Typical I: parametric configuration

Typical F: upward-closed sets

U is an upward-closed set of configurations if

$$c \in U \text{ and } c' \geq c \text{ implies } c' \in U$$

where $\geq$ is the pointwise order on $\mathbb{N}^n$.

Sets D of “dangerous” configurations are typically upward-closed

Example: states M and S of MESI protocol should be mutually exclusive

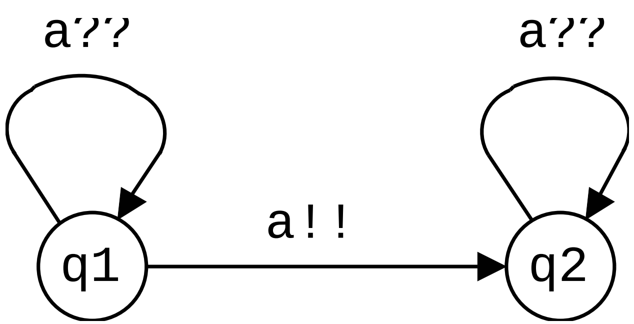
$$D = \{(m, e, s, i) \mid m \geq 1 \wedge s \geq 1\}$$

Is reachability decidable if I is a parametric configuration
and F is an upward-closed set?

First try: Forward search

Since $I \in \mathcal{C}$ required by (2), the family $\mathcal{C}$ must contain all parametrized configurations.

Satisfies (1) - (5) but not (6). Termination fails in very simple cases.



$$(\sqcup, 0) \xrightarrow{a} (\sqcup, 1) \xrightarrow{a} (\sqcup, 2) \xrightarrow{a} \ldots$$

Second try: Backward search

Since $F \in \mathcal{C}$ required by (2), the family $\mathcal{C}$ must contain all upward-closed sets.

[Abdulla et al I&C 160, 2000], [E. et al, LICS'99] :

Backward search satisfies (1) - (6)

1. An upward-closed set can be finitely represented by its set of minimal elements w.r.t. the pointwise order $\leq$

1. An upward-closed set can be finitely represented by its set of minimal elements w.r.t. the pointwise order $\leq$

- An upward-closed set is determined by its minimal elements
- Any subset of $\mathbb{N}^k$ has finitely many minimal elements

1. An upward-closed set can be finitely represented by its set of minimal elements w.r.t. the pointwise order $\leq$

- An upward-closed set is determined by its minimal elements
- Any subset of $\mathbb{N}^k$ has finitely many minimal elements

  Every infinite sequence $c_1, c_2, c_3, \ldots$ of vectors of $\mathbb{N}^k$ contains a non-decreasing infinite subsequence $c_{i_1} \leq c_{i_2} \leq c_{i_3} \ldots$ (Dickson's lemma)

1. An upward-closed set can be finitely represented by its set of minimal elements w.r.t. the pointwise order $\leq$

- An upward-closed set is determined by its minimal elements
- Any subset of $\mathbf{N}^k$ has finitely many minimal elements

Every infinite sequence $c_1, c_2, c_3, \ldots$ of vectors of $\mathbf{N}^k$ contains a non-decreasing infinite subsequence $c_{i_1} \leq c_{i_2} \leq c_{i_3} \ldots$ (Dickson's lemma)

Assume some $X \subseteq \mathbf{N}^k$ has infinitely many minimal elements
Enumerate them in a sequence $m_1, m_2 \ldots$

1. An upward-closed set can be finitely represented by its set of minimal elements w.r.t. the pointwise order $\leq$

- An upward-closed set is determined by its minimal elements
- Any subset of $\mathbb{N}^k$ has finitely many minimal elements

  Every infinite sequence $c_1, c_2, c_3, \ldots$ of vectors of $\mathbb{N}^k$ contains a non-decreasing infinite subsequence $c_{i_1} \leq c_{i_2} \leq c_{i_3} \ldots$ (Dickson's lemma)

  Assume some $X \subseteq \mathbb{N}^k$ has infinitely many minimal elements
  Enumerate them in a sequence $m_1, m_2 \ldots$

  By Dicksons lemma, $m_i \leq m_j$ for some $i < j$

1. An upward-closed set can be finitely represented by its set of minimal elements w.r.t. the pointwise order $\leq$

- An upward-closed set is determined by its minimal elements
- Any subset of $\mathbb{N}^k$ has finitely many minimal elements

Every infinite sequence $c_1, c_2, c_3, \ldots$ of vectors of $\mathbb{N}^k$ contains a non-decreasing infinite subsequence $c_{i_1} \leq c_{i_2} \leq c_{i_3} \ldots$ (Dickson's lemma)

Assume some $X \subseteq \mathbb{N}^k$ has infinitely many minimal elements
Enumerate them in a sequence $m_1, m_2 \ldots$

By Dicksons lemma, $m_i \leq m_j$ for some $i < j$

But then m_j is not minimal

1. An upward-closed set can be finitely represented by its set of minimal elements w.r.t. the pointwise order $\leq$ $\surd$

- An upward-closed set is determined by its minimal elements
- Any subset of $\mathbb{N}^k$ has finitely many minimal elements

  Every infinite sequence $c_1, c_2, c_3, \ldots$ of vectors of $\mathbb{N}^k$ contains a non-decreasing infinite subsequence $c_{i_1} \leq c_{i_2} \leq c_{i_3} \ldots$ (Dickson's lemma)

  Assume some $X \subseteq \mathbb{N}^k$ has infinitely many minimal elements
  Enumerate them in a sequence $m_1, m_2 \ldots$

  By Dicksons lemma, $m_i \leq m_j$ for some $i < j$

  But then m_j is not minimal

  Contradiction

2. F is upward-closed $\surd$

2. F is upward-closed $\surd$

3. If U is upward-closed then so is $U \cup pre(U)$

2. F is upward-closed $\surd$

3. If U is upward-closed then so is $U \cup pre(U)$

Since union of upward-closed sets is upward-closed, it suffices to prove that $pre(U)$ is upward-closed

2. F is upward-closed $\surd$

3. If U is upward-closed then so is $U \cup pre(U)$

Since union of upward-closed sets is upward-closed, it suffices to prove that $pre(U)$ is upward-closed

Take $c \in pre(U)$ and $c' \geq c$. We show $c' \in pre(U)$

2. F is upward-closed $\surd$

3. If U is upward-closed then so is $U \cup pre(U)$

Since union of upward-closed sets is upward-closed, it suffices to prove that $pre(U)$ is upward-closed

Take $c \in pre(U)$ and $c' \geq c$. We show $c' \in pre(U)$

$$\begin{array}{ccc} c & \xrightarrow{a} & u \in U \\ \leq & & \\ c' & & \end{array}$$

2. F is upward-closed $\surd$

3. If U is upward-closed then so is $U \cup pre(U)$

Since union of upward-closed sets is upward-closed, it suffices to prove that $pre(U)$ is upward-closed

Take $c \in pre(U)$ and $c' \geq c$. We show $c' \in pre(U)$

$$\begin{array}{ccc} c & \xrightarrow{a} & u \in U \\ \leq & & \\ c' & \xrightarrow{a} & u' \end{array}$$

2. F is upward-closed $\surd$

3. If U is upward-closed then so is $U \cup pre(U)$

Since union of upward-closed sets is upward-closed, it suffices to prove that $pre(U)$ is upward-closed

Take $c \in pre(U)$ and $c' \geq c$. We show $c' \in pre(U)$

$$\begin{array}{ccc} c & \xrightarrow{a} & u \in U \\ \leq & & \leq \\ c' & \xrightarrow{a} & u' \end{array}$$

2. F is upward-closed $\surd$

3. If U is upward-closed then so is $U \cup pre(U)$ $\surd$

Since union of upward-closed sets is upward-closed, it suffices to prove that $pre(U)$ is upward-closed

Take $c \in pre(U)$ and $c' \geq c$. We show $c' \in pre(U)$

$$\begin{array}{ccc} c & \xrightarrow{a} & u \in U \\ \leq & & \leq \\ c' & \xrightarrow{a} & u' \in U \end{array}$$

4. $C \cap I$ is decidable $\checkmark$

4. $C \cap I$ is decidable $\surd$

5. $C_1 = C_2$ is decidable $\surd$

4. $C \cap I$ is decidable $\surd$

5. $C_1 = C_2$ is decidable $\surd$

6. Any chain $U_1 \subseteq U_2 \subseteq U_3 \ldots$ of upward-closed sets reaches a fixpoint after finitely many steps

4. $C \cap I$ is decidable $\surd$

5. $C_1 = C_2$ is decidable $\surd$

6. Any chain $U_1 \subseteq U_2 \subseteq U_3 \ldots$ of upward-closed sets reaches a fixpoint after finitely many steps

Assume this is not the case: $U_1 \subset U_2 \subset U_3 \ldots$

4. $C \cap I$ is decidable $\surd$

5. $C_1 = C_2$ is decidable $\surd$

6. Any chain $U_1 \subseteq U_2 \subseteq U_3 \ldots$ of upward-closed sets reaches a fixpoint after finitely many steps

Assume this is not the case: $U_1 \subset U_2 \subset U_3 \ldots$

Pick some minimal element $m_1 \in U_1$
Pick for every $i > 1$ some minimal element $m_i \notin U_1 \cup \ldots \cup U_{i-1}$
Consider the sequence $m_1, m_2, m_3, \ldots$

4. $C \cap I$ is decidable $\surd$

5. $C_1 = C_2$ is decidable $\surd$

6. Any chain $U_1 \subseteq U_2 \subseteq U_3 \ldots$ of upward-closed sets reaches a fixpoint after finitely many steps

Assume this is not the case: $U_1 \subset U_2 \subset U_3 \ldots$

Pick some minimal element $m_1 \in U_1$
Pick for every $i > 1$ some minimal element $m_i \notin U_1 \cup \ldots \cup U_{i-1}$
Consider the sequence $m_1, m_2, m_3, \ldots$

Let $i < j$; since $m_j \notin U_i$, we have $m_i \not\leq m_j$ (upward-closedness)

4. $C \cap I$ is decidable $\surd$

5. $C_1 = C_2$ is decidable $\surd$

6. Any chain $U_1 \subseteq U_2 \subseteq U_3 \ldots$ of upward-closed sets reaches a fixpoint after finitely many steps

Assume this is not the case: $U_1 \subset U_2 \subset U_3 \ldots$

Pick some minimal element $m_1 \in U_1$
Pick for every $i > 1$ some minimal element $m_i \notin U_1 \cup \ldots \cup U_{i-1}$
Consider the sequence $m_1, m_2, m_3, \ldots$

Let $i < j$; since $m_j \notin U_i$, we have $m_i \not\leq m_j$ (upward-closedness)

So infinitely many elements of $m_1, m_2, m_3 \ldots$ are incomparable

4. $C \cap I$ is decidable $\surd$

5. $C_1 = C_2$ is decidable $\surd$

6. Any chain $U_1 \subseteq U_2 \subseteq U_3 \ldots$ of upward-closed sets reaches a fixpoint after finitely many steps $\surd$

Assume this is not the case: $U_1 \subset U_2 \subset U_3 \ldots$

Pick some minimal element $m_1 \in U_1$
Pick for every $i > 1$ some minimal element $m_i \notin U_1 \cup \ldots \cup U_{i-1}$
Consider the sequence $m_1, m_2, m_3, \ldots$

Let $i < j$; since $u_j \notin U_i$, we have $m_i \not\leq m_j$ (upward-closedness)

So infinitely many elements of $m_1, m_2, m_3 \ldots$ are incomparable

Contradiction to Dickson's lemma

Repeated reachability in broadcast protocols

The following problem is undecidable:

Given:	a broadcast protocol,
	an initial parametric configuration $p = (\sqcup, 0, \ldots, 0)$
To decide:	is there an integer n such that the transition system
	with $(n, 0, \ldots, 0)$ as initial configuration
	has an infinite computation ?

Can be reformulated as a repeated reachability problem where
$I = (\sqcup, 0, \ldots, 0)$ and $F =$ set of all configurations

Application to the MESI-protocol

Are the states M and S mutually exclusive?

Check if the upward-closed set with minimal element

$$m = 1,\ e = 0,\ s = 1,\ i = 0$$

can be reached from the initial p-configuration

$$m = 0,\ e = 0,\ s = 0,\ i = \sqcup$$

Application to the MESI-protocol

Are the states M and S mutually exclusive?

Check if the upward-closed set with minimal element

$$m = 1,\ e = 0,\ s = 1,\ i = 0$$

can be reached from the initial p-configuration

$$m = 0,\ e = 0,\ s = 0,\ i = \sqcup$$

Proceed as follows:

$$D: \quad m \geq 1 \wedge s \geq 1$$

Application to the MESI-protocol

Are the states M and S mutually exclusive?

Check if the upward-closed set with minimal element

$$m = 1,\ e = 0,\ s = 1,\ i = 0$$

can be reached from the initial p-configuration

$$m = 0,\ e = 0,\ s = 0,\ i = \sqcup$$

Proceed as follows:

$$\begin{aligned} D\colon &\quad m \geq 1 \wedge s \geq 1 \\ D \cup pre(D)\colon &\quad (m \geq 1 \wedge s \geq 1) \vee \\ &\quad (m = 0 \wedge e = 1 \wedge s \geq 1) \end{aligned}$$

Application to the MESI-protocol

Are the states M and S mutually exclusive?

Check if the upward-closed set with minimal element

$$m = 1,\ e = 0,\ s = 1,\ i = 0$$

can be reached from the initial p-configuration

$$m = 0,\ e = 0,\ s = 0,\ i = \sqcup$$

Proceed as follows:

$$\begin{aligned}
D\colon &\quad m \geq 1 \wedge s \geq 1 \\
D \cup pre(D)\colon &\quad (m \geq 1 \wedge s \geq 1) \vee \\
&\quad (m = 0 \wedge e = 1 \wedge s \geq 1) \\
D \cup pre(D) \cup pre^2(D)\colon &\quad D \cup pre(D)
\end{aligned}$$

Case studies (by Delzanno)

Broadcast protocols must be extended with more complicated guards.

Termination guarantee gets lost, but can be recovered

Upward-closed sets represented by linear constraints

Backward-search algorithm must be refined

Possibly more iterations, but each iteration has lower complexity

Berkeley RISC, Illinois, Xerox PARC Dragon, DEC Firefly

At most 7 iterations and below 100 seconds (SPARC5, Pentium 133)

Futurebus +

8 steps and 200 seconds (Pentium 133)

Symbolic search for other models

FIFO-automata with lossy channels

[Abdulla and Jonsson, I&C 127, 1993], [Abdulla et al, CAV'98, LNCS 1427]

Configuration: pair $(q, \mathbf{w})$, where q state and $\mathbf{w} = (w_1, \ldots, w_n)$ vector of words representing the queue contents

Family $\mathcal{C}$: upward-closed sets with respect to the subsequence order

abba $\leq$ *bbaabaaabbabb*

Dickson's lemma $\rightarrow$ Higman's lemma

Backward search satisfies (1) - (6)

Timed automata

[Alur and Dill, TCS 126, 1994]

Configuration: pair $(q, \mathbf{x})$, where q state and $\mathbf{x}$ vector of real numbers

Family $\mathcal{C}$: regions or zones

Forward and backward search satisfy (1) - (6)

Pushdown systems

A pushdown system (PDS) is a triple (P, Γ, Δ), where

- P is a finite set of control locations
- Γ is a finite stack alphabet
- $\Delta \subseteq (P \times \Gamma) \times (P \times \Gamma^*)$ is a finite set of rules.

A configuration is a pair $\langle p, v \rangle$, where $p \in P$, $v \in \Gamma^*$

If $\langle p, \gamma \rangle \hookrightarrow \langle p', v \rangle \in \Delta$ then $\langle p, \gamma w \rangle \longrightarrow \langle p', vw \rangle$ for every $w \in \Gamma^*$

Normalisation: $|v| \leq 2$

PDSs as models of sequential programs

Programs determined by

control flow of procedures

- assignments, conditionals, loops
- procedure calls with parameter passing / return values

local variables of each procedure

global variables

State space determined by

program pointer

values of global variables

values of local variables (of current procedure)

activation records (return addresses, copies of locals)

Interpretation of $\langle p, \gamma v \rangle$

p holds values of global variables

γ holds (program pointer, values of local variables)

v holds stack of (return address, saved locals)

Restriction: finite datatypes

Correspondence between statements and rules

$\langle p, \gamma \rangle \hookrightarrow \langle p', \gamma' \rangle$	simple statement
$\langle p, \gamma \rangle \hookrightarrow \langle p', \gamma' \gamma'' \rangle$	procedure call
$\langle p, \gamma \rangle \hookrightarrow \langle p', \epsilon \rangle$	return statement

Reachability in pushdown systems

A set of configurations C is regular if for every control point p, the set $\{w \in \Gamma^* \mid \langle p, w \rangle \in C\}$ is regular

Typically, I and F are regular sets of configurations (even very simple ones, like $\langle p, \Gamma^* \rangle$)

Family $\mathcal{C}$: regular sets

Backward search: Do conditions (2) - (6) hold ?

1. Each regular set can be finitely represented by a multi-automaton $\surd$

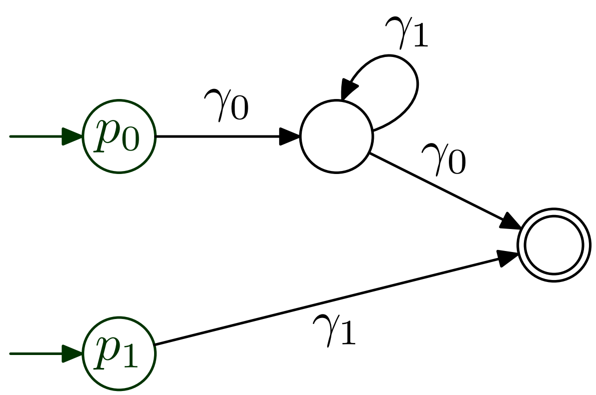
Multi-automata for a pushdown system:

P as set of initial states and Γ as alphabet

$\langle p, v \rangle$ recognized if $p \xrightarrow{v} q$ for some final state q

Example: $P = \{p_0, p_1\}$ and $\Gamma = \{\gamma_0, \gamma_1\}$

Automaton coding the set $\langle p_0, \gamma_0 \gamma_1^* \gamma_0 \rangle \cup \langle p_1, \gamma_1 \rangle$:



2. $F \in \mathcal{C}$ $\checkmark$

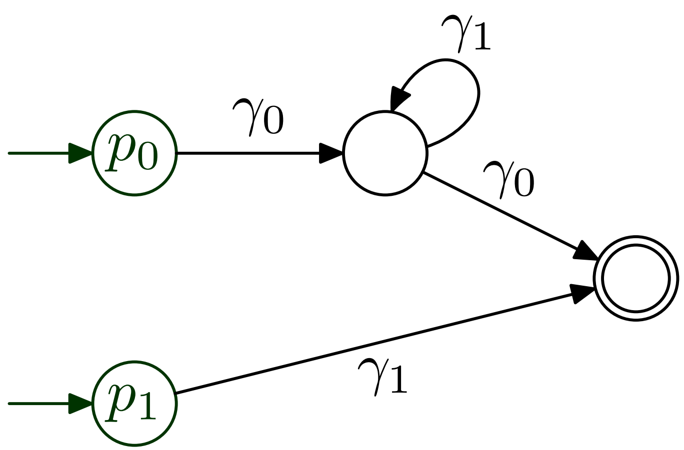
2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

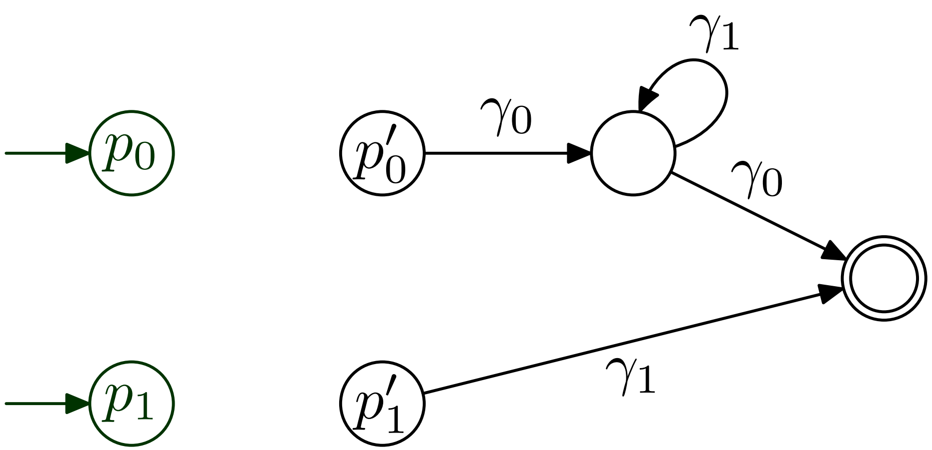
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

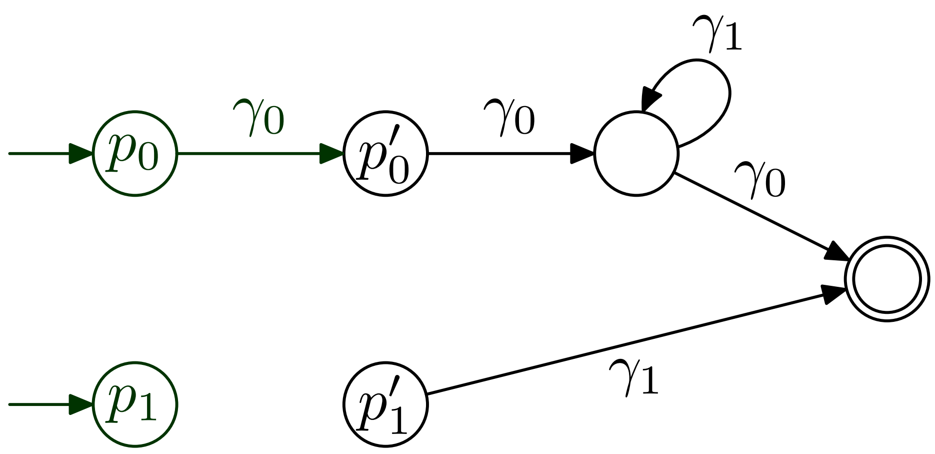
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

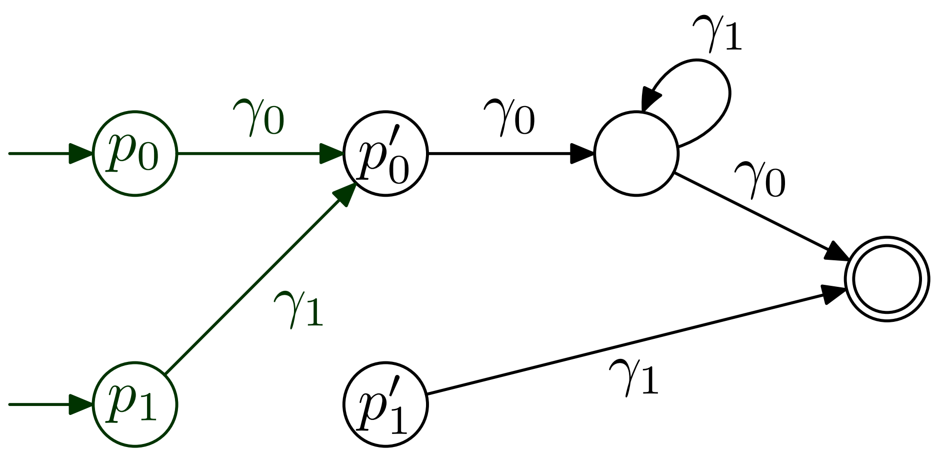
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \, , \, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \, , \, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\checkmark$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

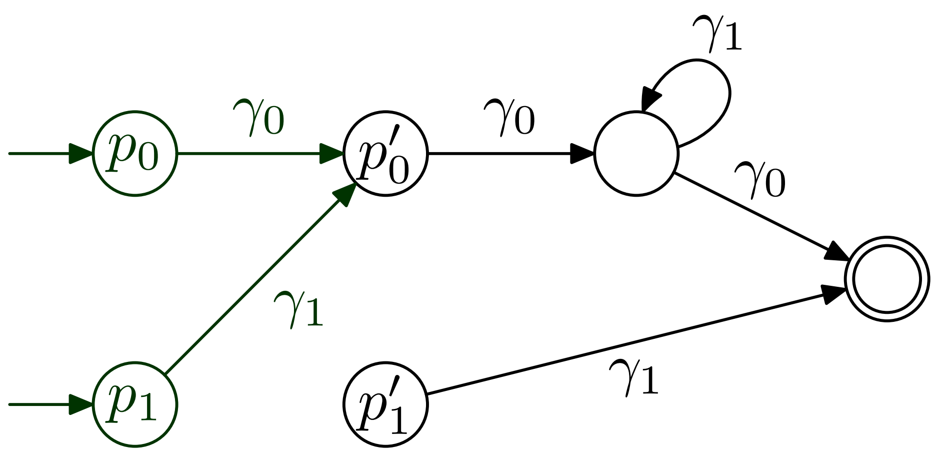
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

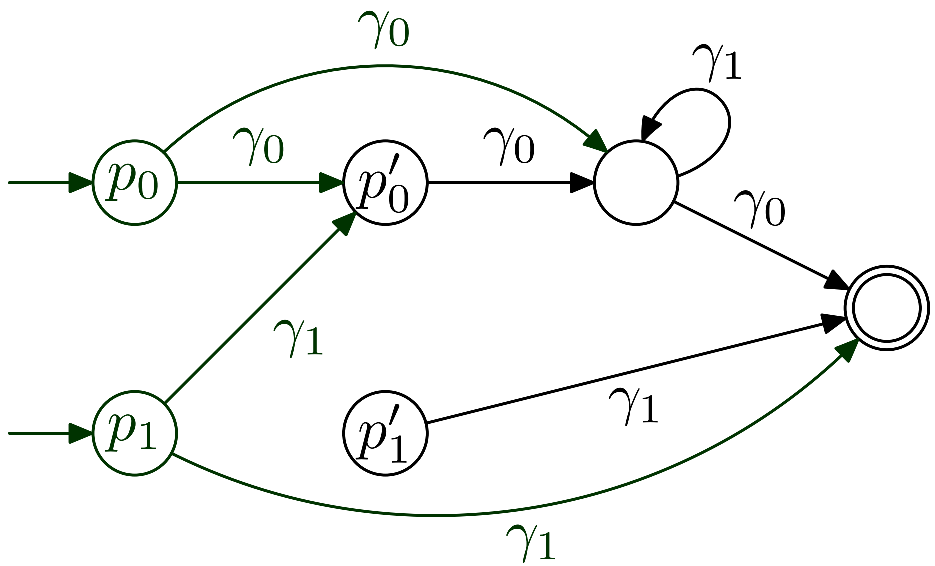
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

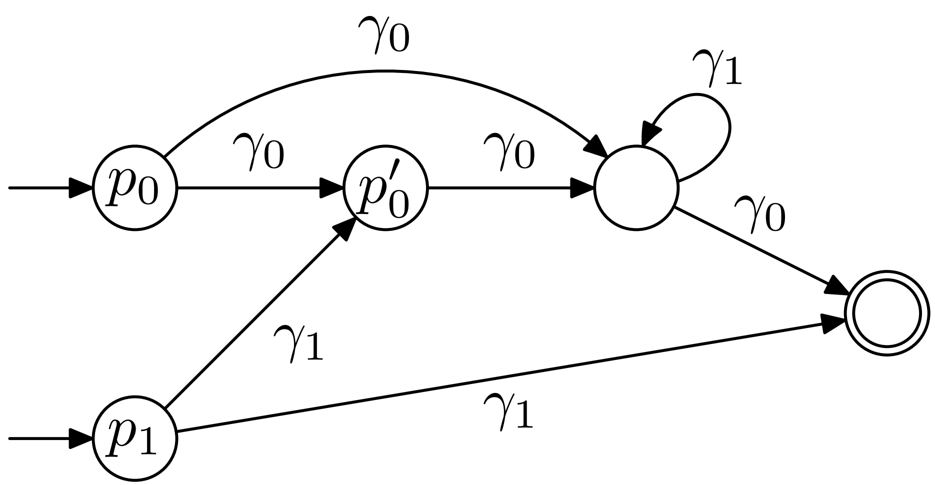
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

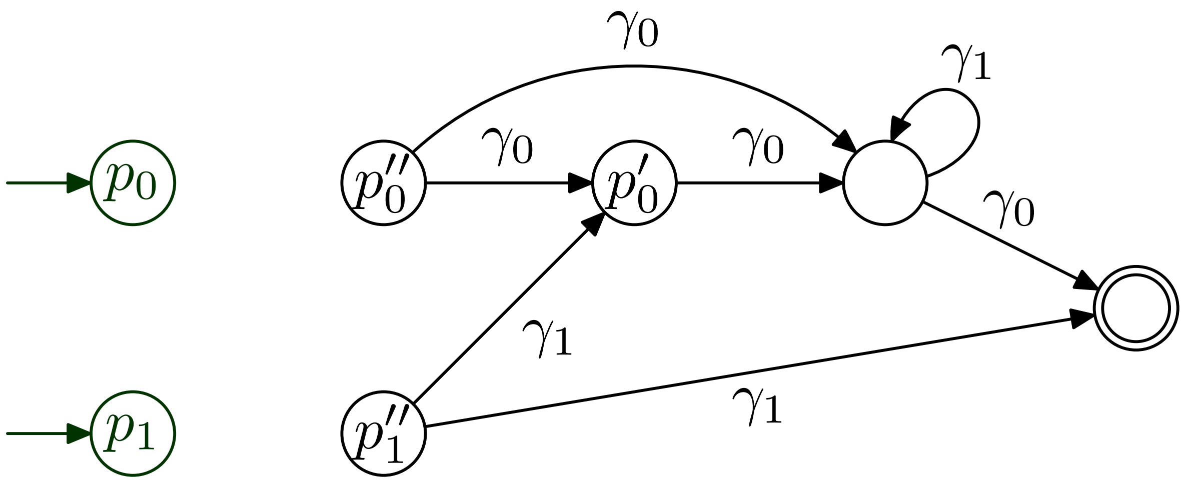
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

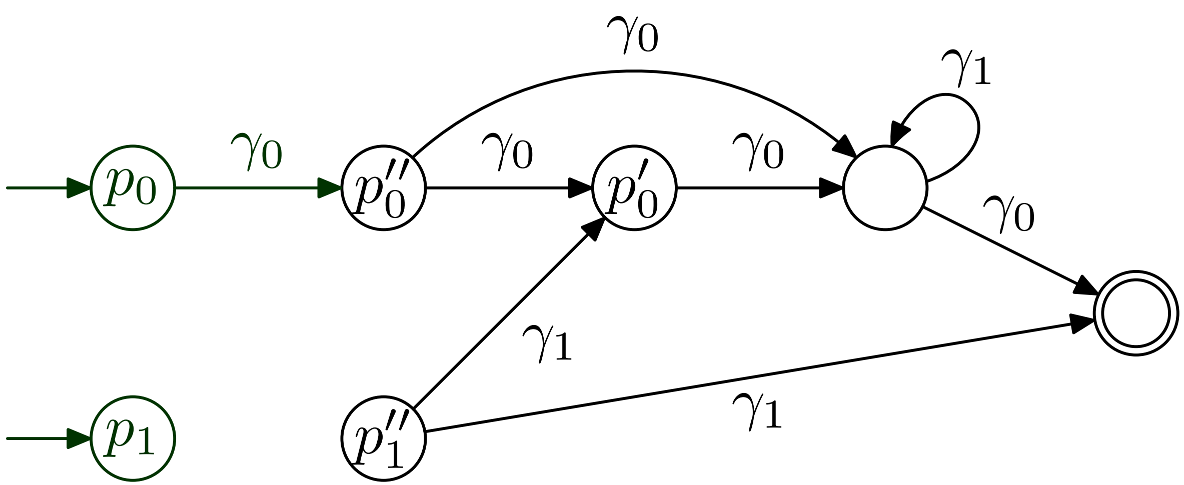
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \;,\; \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \;,\; \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\checkmark$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

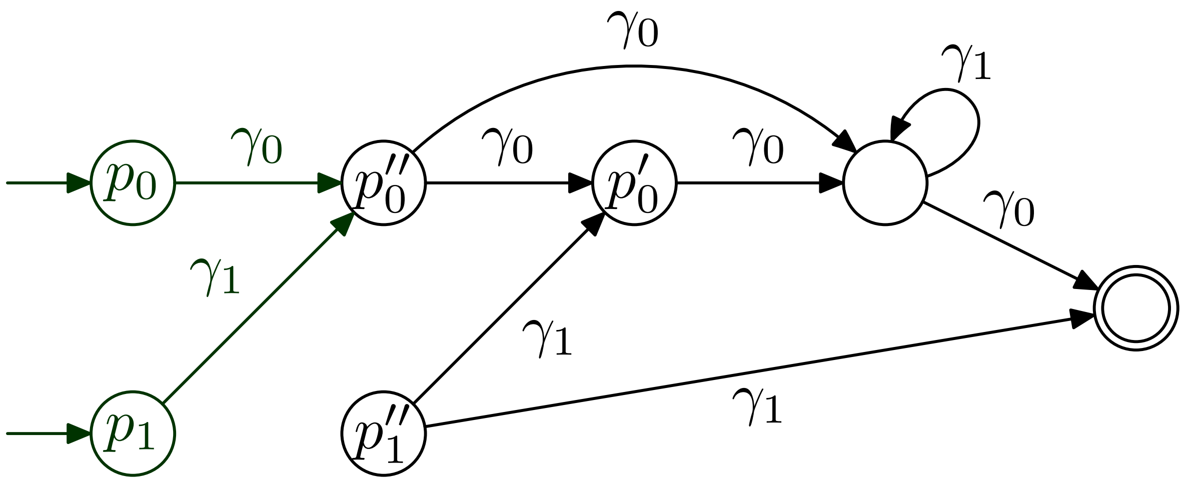
$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1\gamma_0 \rangle \}$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

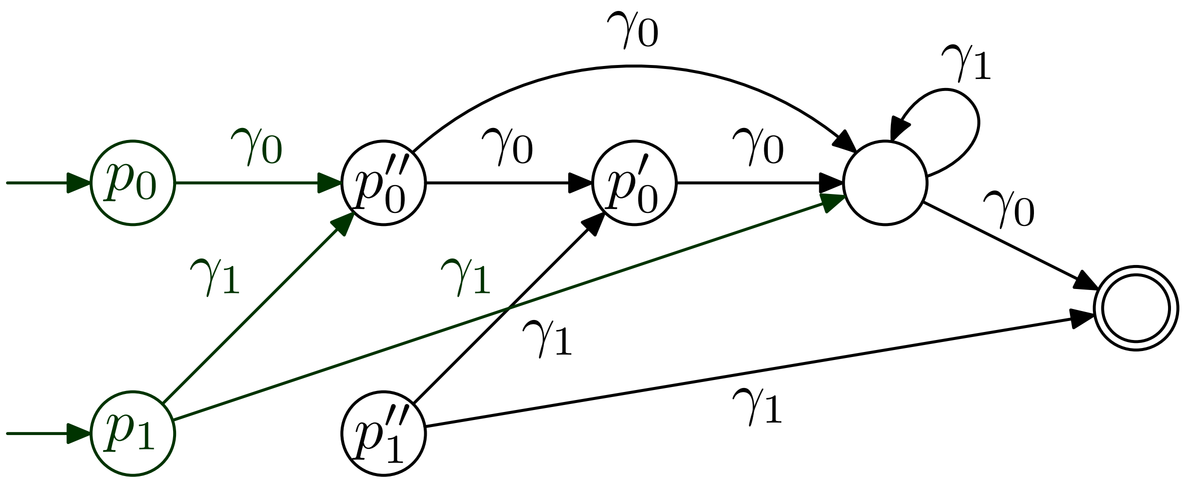
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \, , \, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \, , \, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

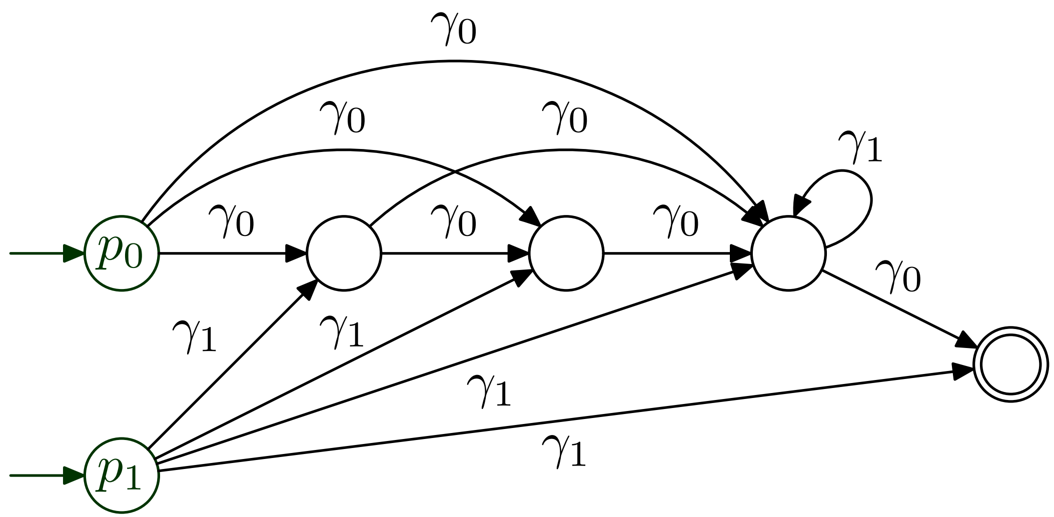
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$

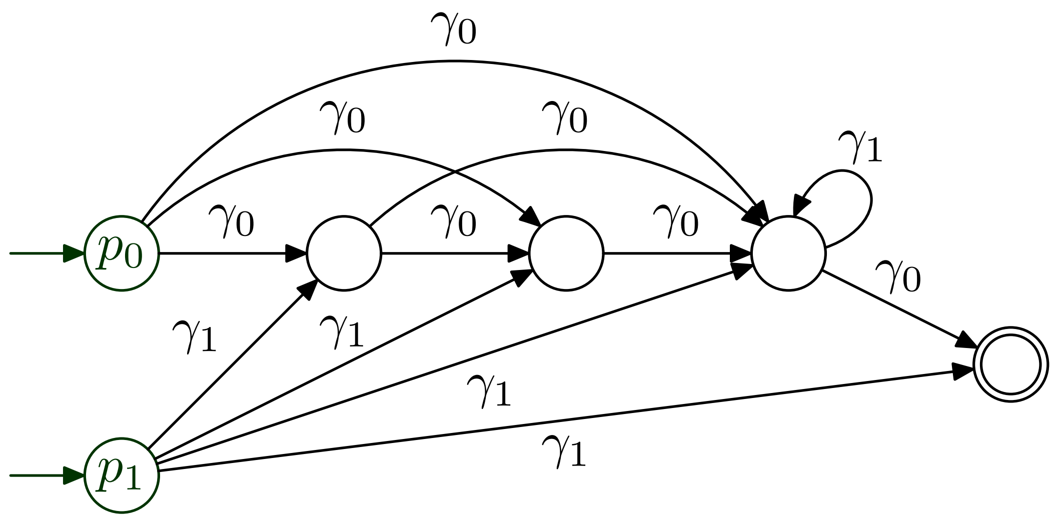
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



2. $F \in \mathcal{C}$ $\surd$

3. If $C \in \mathcal{C}$, then $C \cup pre(C) \in \mathcal{C}$ $\surd$

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



4. Emptyness of $C \cap I$ is decidable $\surd$

4. Emptyness of $C \cap I$ is decidable $\surd$

5. $C_1 = C_2$ is decidable $\surd$

6. Any chain $C_1 \subseteq C_2 \subseteq C_3 \ldots$ eventually reaches a fixpoint

6. Any chain $C_1 \subseteq C_2 \subseteq C_3 \ldots$ eventually reaches a fixpoint

$P = \{p_0, p_1\}$, $\Gamma = \{\gamma_0, \gamma_1\}$

$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$

6. Any chain $C_1 \subseteq C_2 \subseteq C_3 \ldots$ eventually reaches a fixpoint

$P = \{p_0, p_1\}$, $\Gamma = \{\gamma_0, \gamma_1\}$

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle , \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

$$C_0 \;=\; F \qquad\qquad = \; \langle p_0, \gamma_0 \gamma_1^* \gamma_0 \rangle \cup \langle p_1, \gamma_1 \rangle$$

6. Any chain $C_1 \subseteq C_2 \subseteq C_3 \ldots$ eventually reaches a fixpoint

$P = \{p_0, p_1\}$, $\Gamma = \{\gamma_0, \gamma_1\}$

$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \, , \, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \, , \, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$

$$
\begin{array}{lcllcl}
C_0 & = & F & & = & \langle p_0, \gamma_0 \gamma_1^* \gamma_0 \rangle \, \cup \, \langle p_1, \gamma_1 \rangle \\
C_1 & = & C_0 \cup pre(C_0) & & = & \langle p_0, (\gamma_0 + \gamma_0^2) \gamma_1^* \gamma_0 \rangle \cup \\
 & & & & & \langle p_1, \gamma_1 (\epsilon + \gamma_0) \gamma_1^* (\epsilon + \gamma_0) \rangle
\end{array}
$$

6. Any chain $C_1 \subseteq C_2 \subseteq C_3 \ldots$ eventually reaches a fixpoint

$P = \{p_0, p_1\}$, $\Gamma = \{\gamma_0, \gamma_1\}$

$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \, , \, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \, , \, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$

$$
\begin{array}{rcl}
C_0 & = & F & = & \langle p_0, \gamma_0 \gamma_1^* \gamma_0 \rangle \ \cup \ \langle p_1, \gamma_1 \rangle \\
C_1 & = & C_0 \cup pre(C_0) & = & \langle p_0, (\gamma_0 + \gamma_0^2) \gamma_1^* \gamma_0 \rangle \cup \\
 & & & & \langle p_1, \gamma_1 (\epsilon + \gamma_0) \gamma_1^* (\epsilon + \gamma_0) \rangle \\
 & \ldots & & & \\
C_i & = & C_{i-1} \cup pre(C_{i-1}) & = & \langle p_0, (\gamma_0 + \ldots + \gamma_0^{i+1}) \gamma_1^* \gamma_0 \rangle \cup \\
 & & & & \langle p_1, \gamma_1 (\epsilon + \gamma_0 + \ldots + \gamma_0^{i}) \gamma_1^* (\epsilon + \gamma_0) \rangle \\
 & \ldots & & &
\end{array}
$$

6. Any chain $C_1 \subseteq C_2 \subseteq C_3 \ldots$ eventually reaches a fixpoint NO!

$P = \{p_0, p_1\}$, $\Gamma = \{\gamma_0, \gamma_1\}$

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

$$\begin{array}{lclcl}
C_0 & = & F & = & \langle p_0, \gamma_0 \gamma_1^* \gamma_0 \rangle \;\cup\; \langle p_1, \gamma_1 \rangle \\
C_1 & = & C_0 \cup pre(C_0) & = & \langle p_0, (\gamma_0 + \gamma_0^2) \gamma_1^* \gamma_0 \rangle \cup \\
 & & & & \langle p_1, \gamma_1 (\epsilon + \gamma_0) \gamma_1^* (\epsilon + \gamma_0) \rangle \\
 & \ldots & & & \\
C_i & = & C_{i-1} \cup pre(C_{i-1}) & = & \langle p_0, (\gamma_0 + \ldots + \gamma_0^{i+1}) \gamma_1^* \gamma_0 \rangle \cup \\
 & & & & \langle p_1, \gamma_1 (\epsilon + \gamma_0 + \ldots + \gamma_0^{i}) \gamma_1^* (\epsilon + \gamma_0) \rangle \\
 & \ldots & & &
\end{array}$$

However, the fixpoint

$$
\begin{aligned}
pre^*(F) \;=\; & \langle p_0, \gamma_0^{+}\gamma_1^{*}\gamma_0\rangle \cup \\
& \langle p_1, \gamma_1\gamma_0^{*}\gamma_1^{*}(\epsilon + \gamma_0)\rangle
\end{aligned}
$$

is regular

How can we compute it?

Accelerations

By definition, $pre(F) = \bigcup_{i \geq 0} C_i$
where $C_0 = F$ and $C_{i+1} = C_i \cup pre(C_i)$ for every $i \geq 0$

If convergence fails, try to compute an acceleration :
a sequence $D_0 \subseteq D_1 \subseteq D_2 \ldots$ such that

(a) $\forall i \geq 0 : C_i \subseteq D_i$

(b) $\forall i \geq 0 : D_i \subseteq \bigcup_{j \geq 0} C_j = pre(F)$

Property (a) ensures capture of (at least) the whole set $pre(F)$

Property (b) ensures that only elements of $pre(F)$ are captured

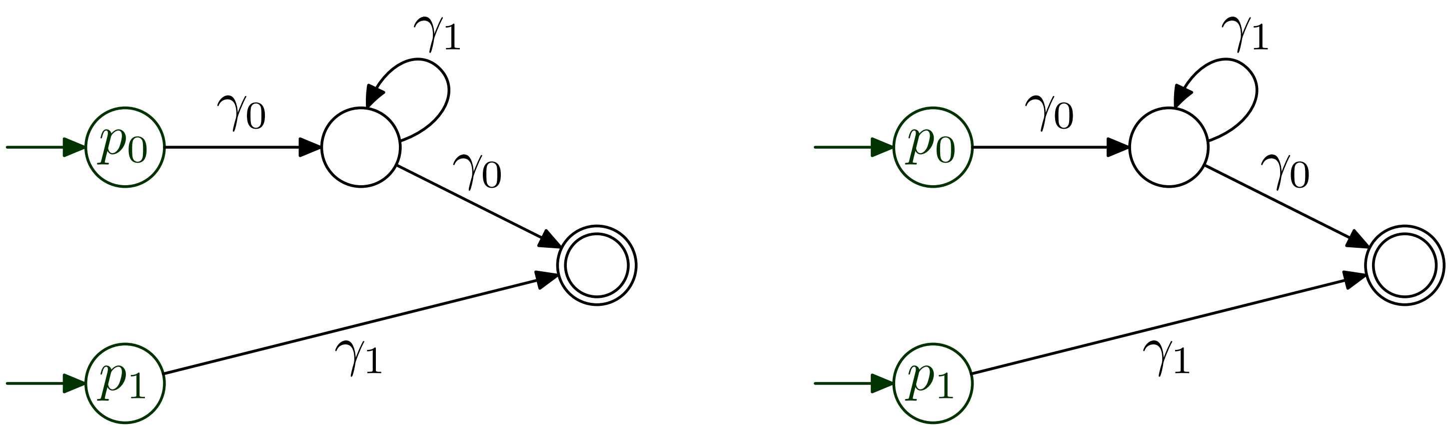
The acceleration guarantees termination if

(c) $\exists i \geq 0 : D_{i+1} = D_i$

An acceleration for pushdown systems

Idea: reuse the same states

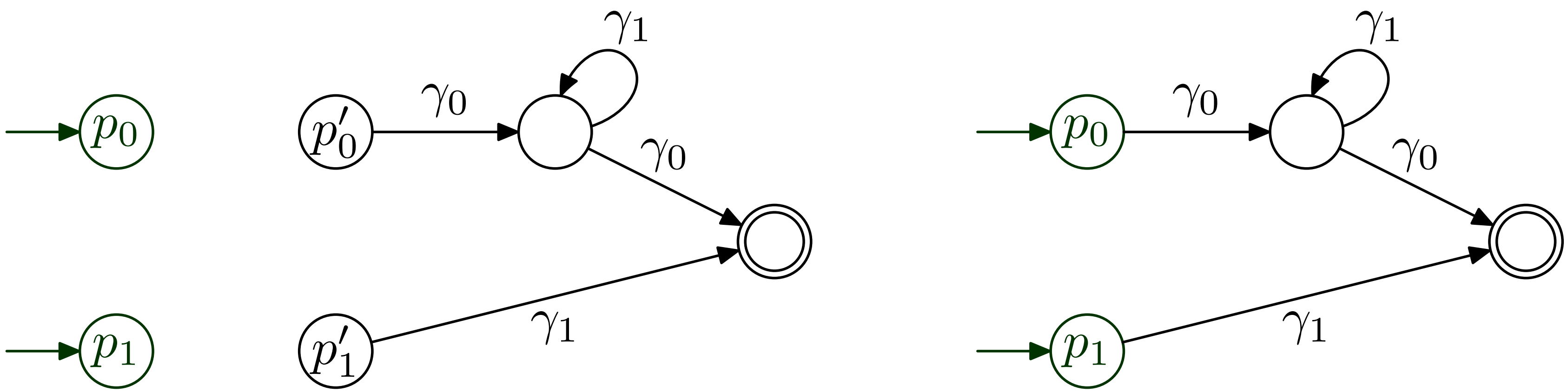
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \ \rangle \}$$



An acceleration for pushdown systems

Idea: try to reuse the same states

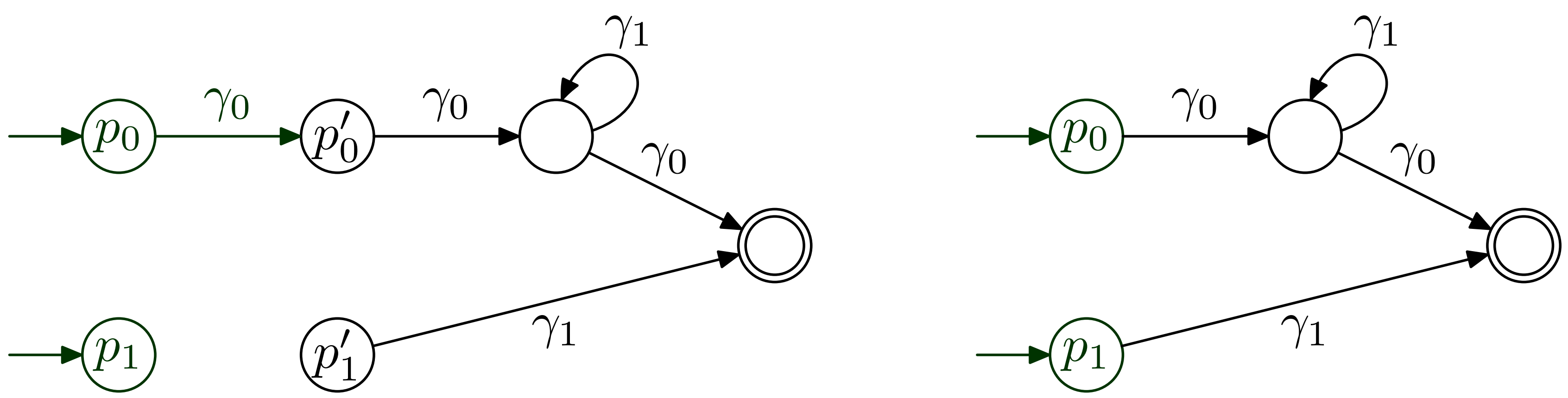
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



An acceleration for pushdown systems

Idea: try to reuse the same states

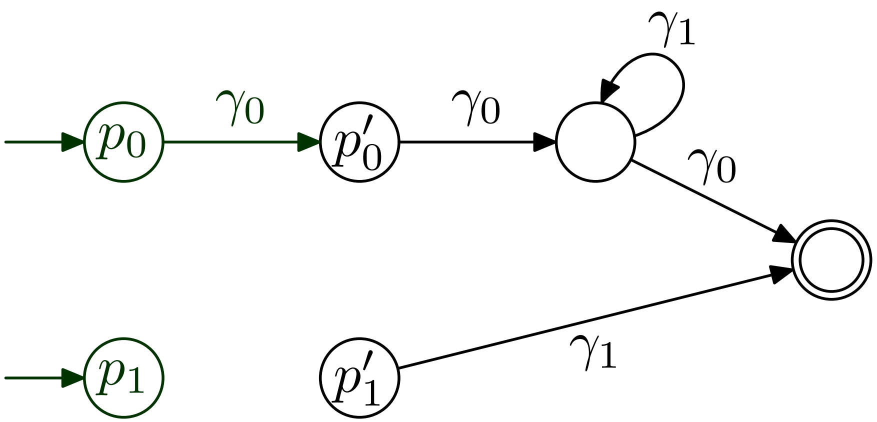
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \ \rangle \}$$

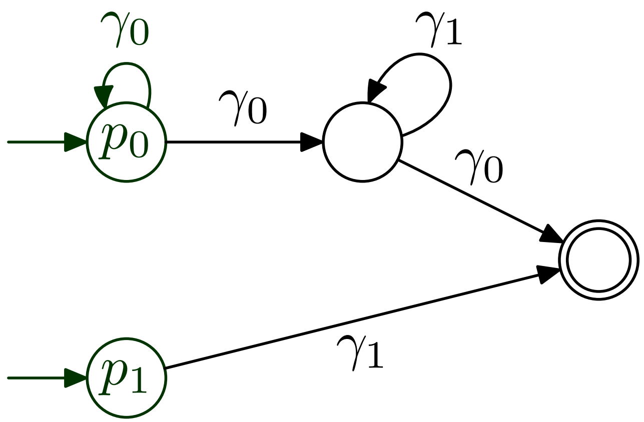


An acceleration for pushdown systems

Idea: try to reuse the same states

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle\ ,\ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle\ ,\ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1\gamma_0 \rangle \}$$

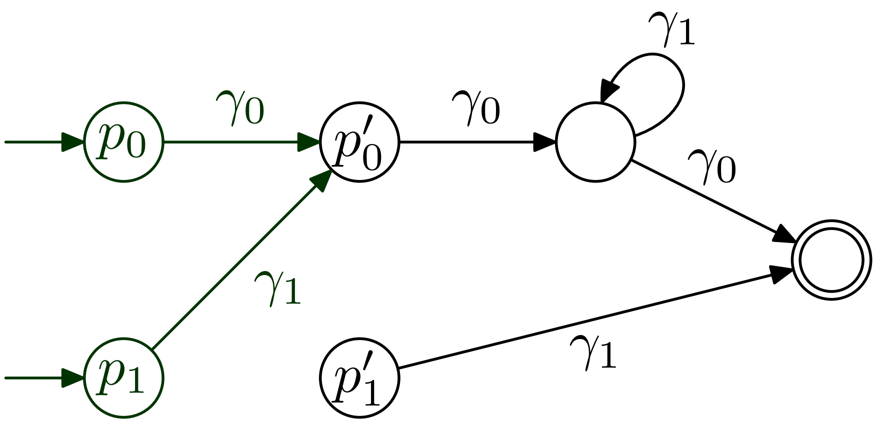


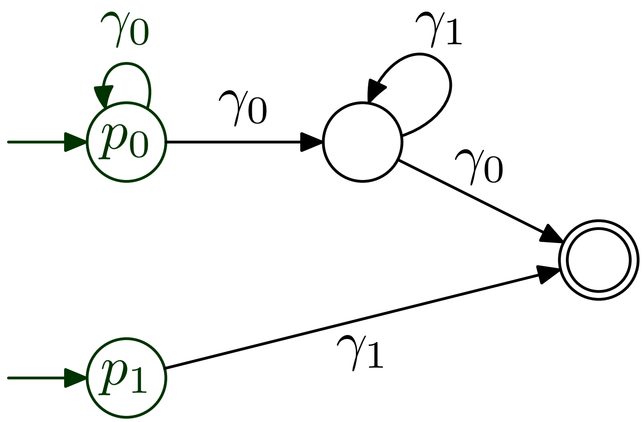


An acceleration for pushdown systems

Idea: try to reuse the same states

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

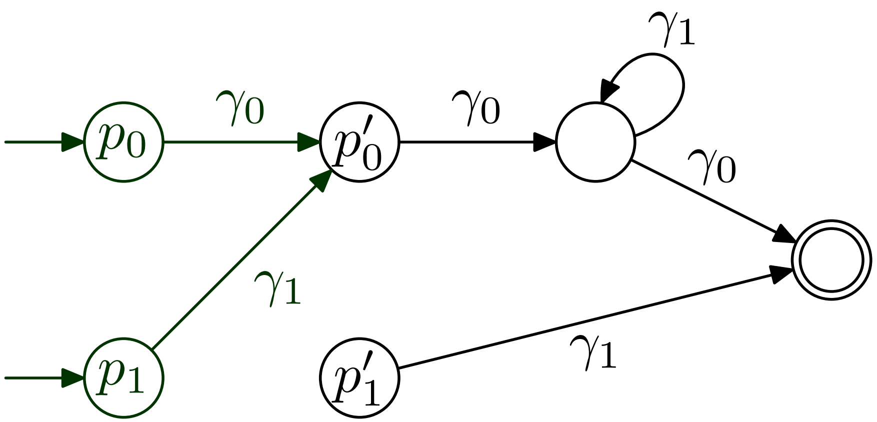


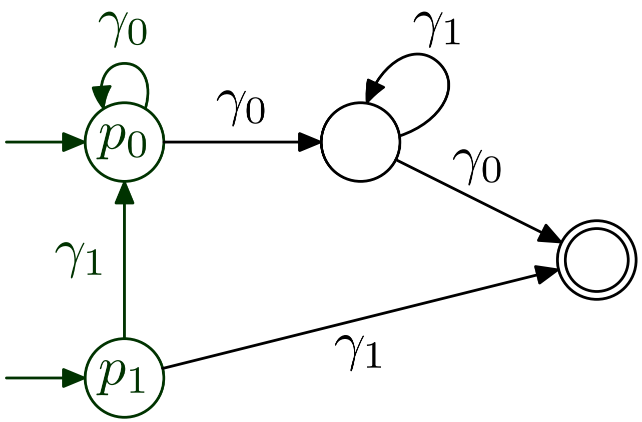


An acceleration for pushdown systems

Idea: try to reuse the same states

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

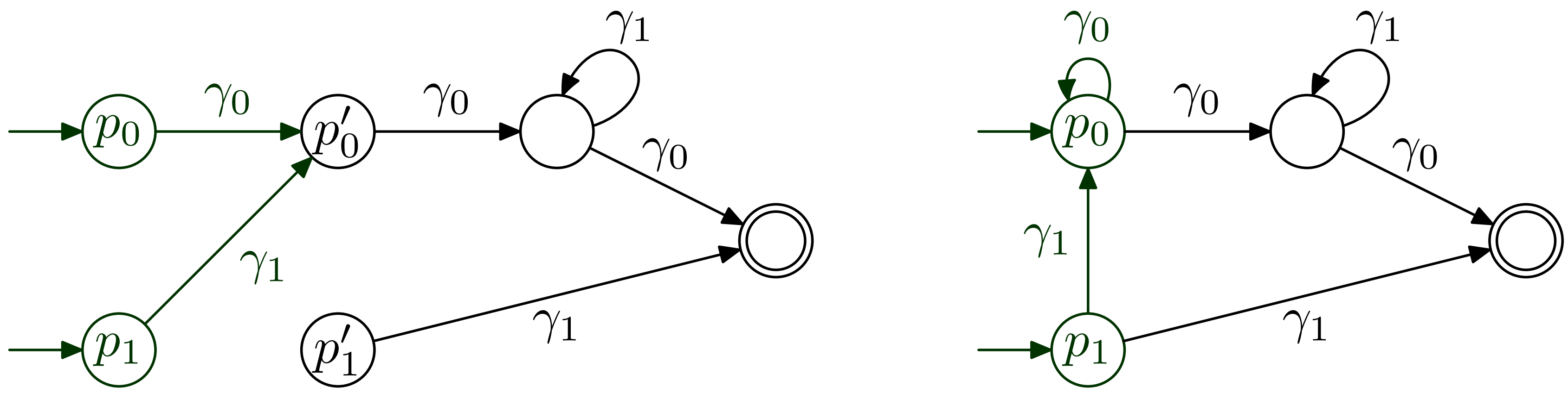




An acceleration for pushdown systems

Idea: try to reuse the same states

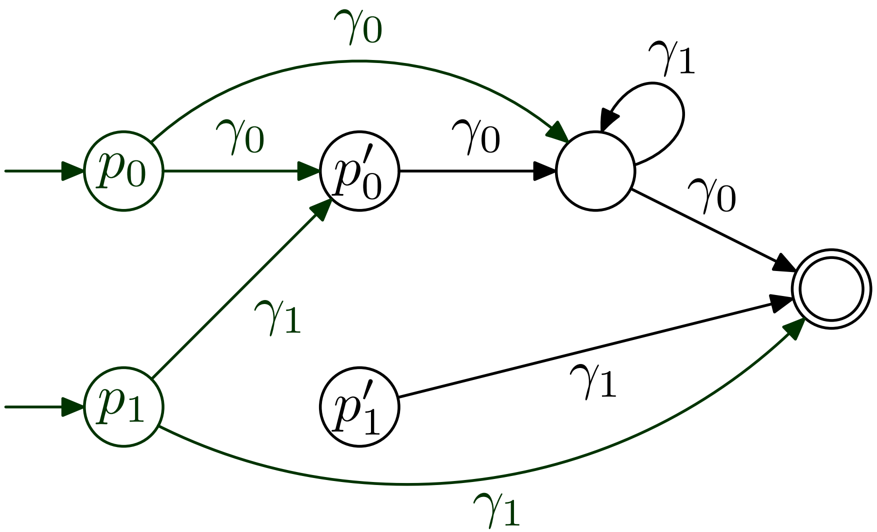
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

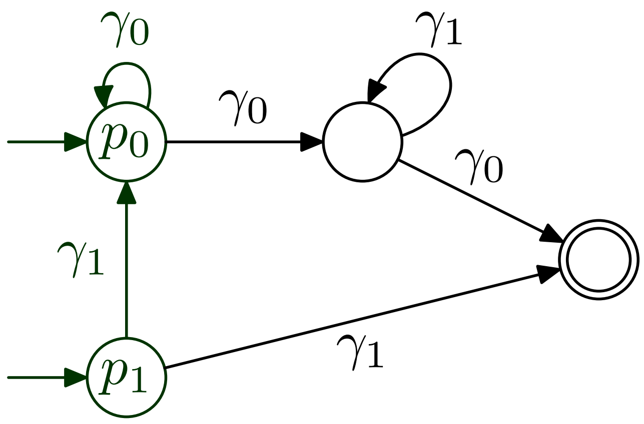


An acceleration for pushdown systems

Idea: try to reuse the same states

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

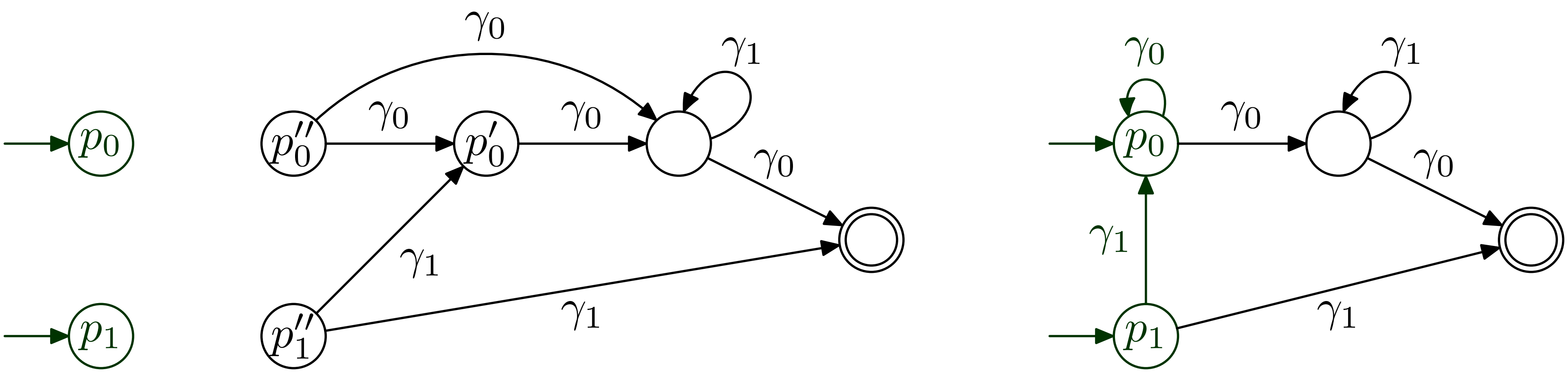




An acceleration for pushdown systems

Idea: try to reuse the same states

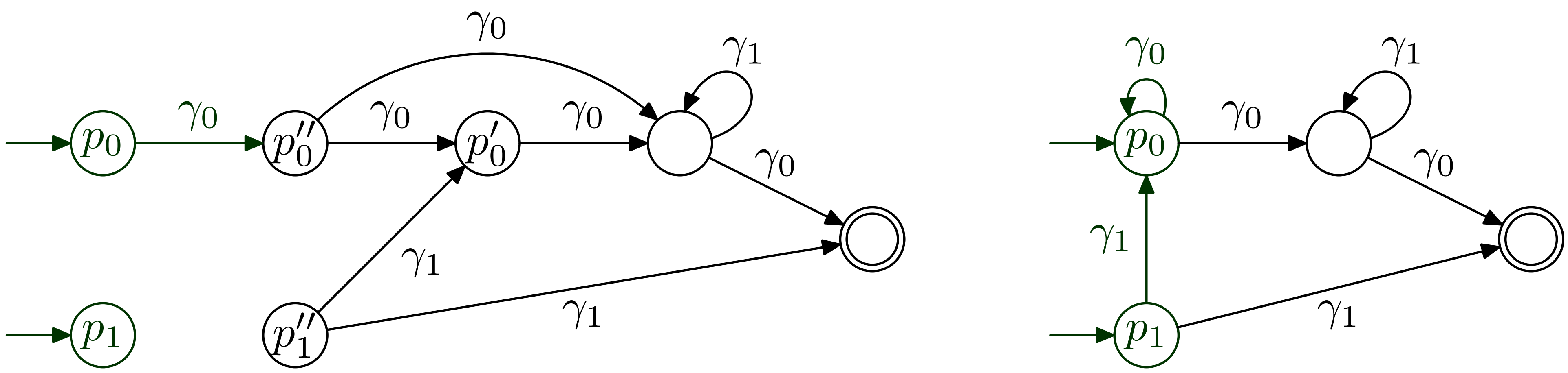
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



An acceleration for pushdown systems

Idea: try to reuse the same states

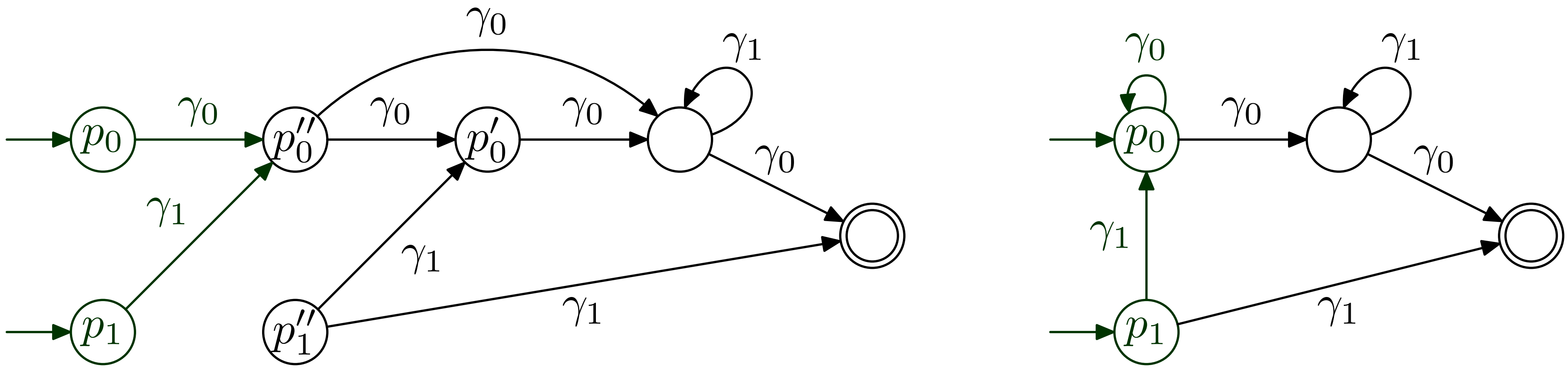
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\, \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



An acceleration for pushdown systems

Idea: try to reuse the same states

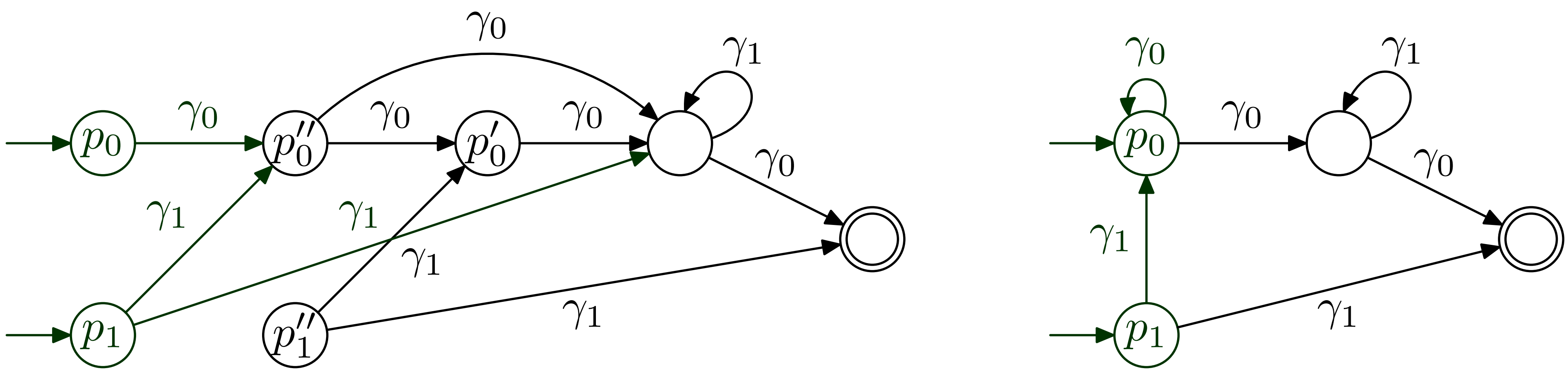
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \,,\ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



An acceleration for pushdown systems

Idea: try to reuse the same states

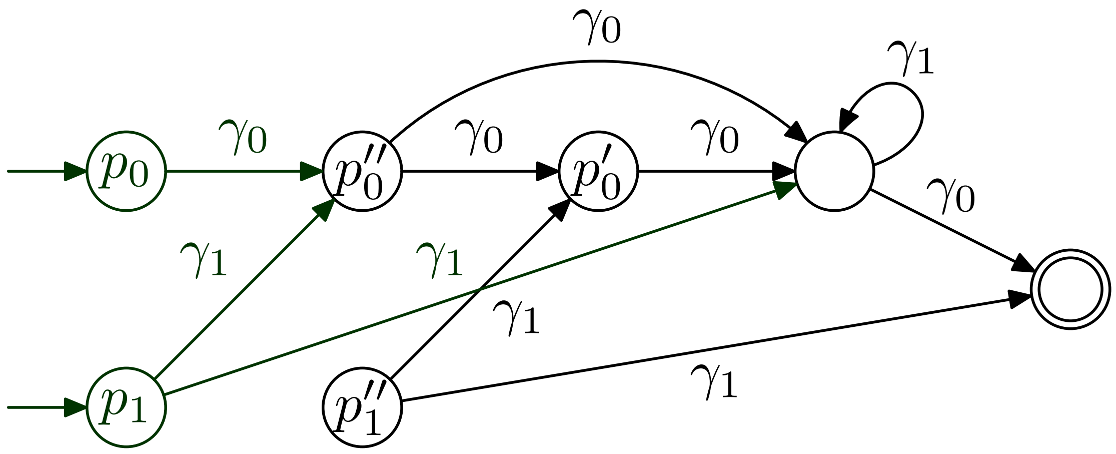
$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \; , \; \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \; , \; \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

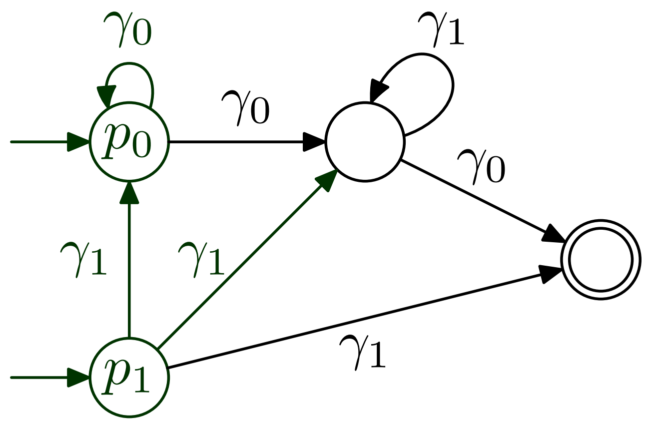


An acceleration for pushdown systems

Idea: try to reuse the same states

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

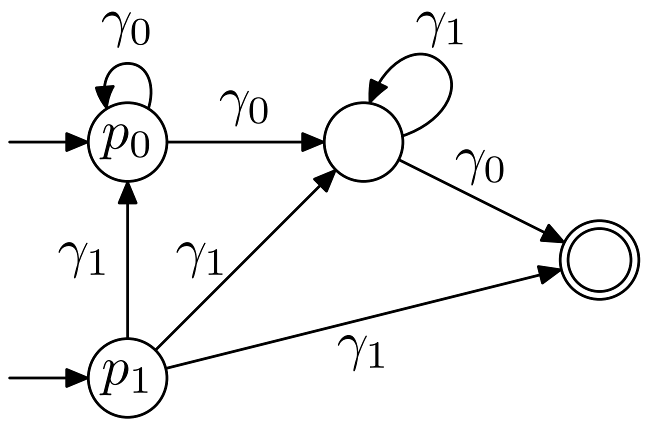




An acceleration for pushdown systems

Idea: try to reuse the same states

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

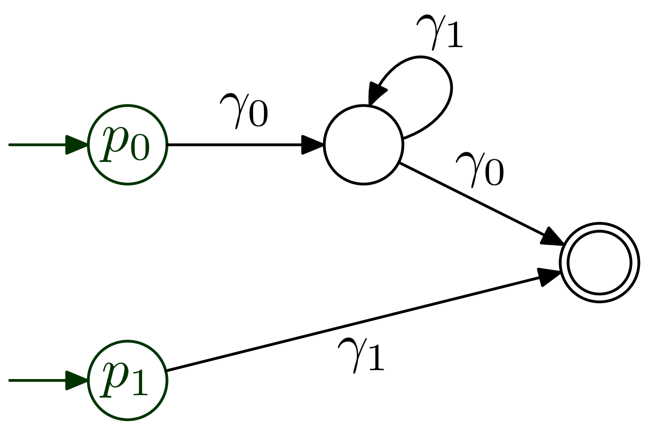


But does it work . . . ?

All predecessors are computed, and termination guaranteed

But: we might be adding non-predecessors

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \text{ , } \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \text{ , } \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$

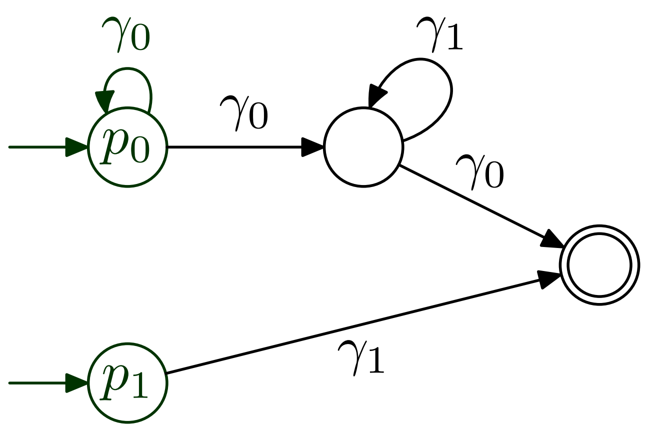


But does it work . . . ?

All predecessors are computed, and termination guaranteed

But: we might be adding non-predecessors

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \ , \ \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \ \rangle \}$$

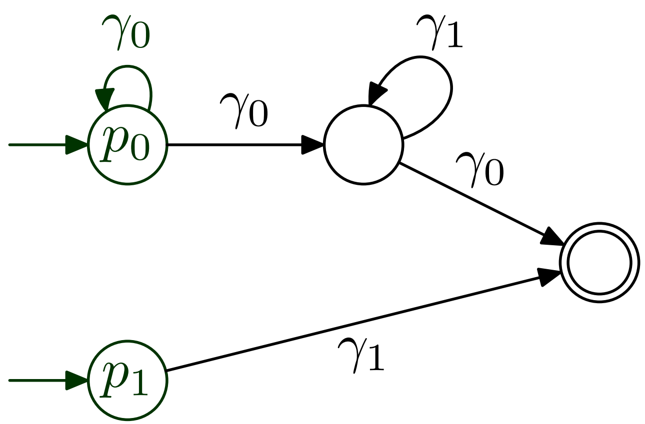


But does it work . . . ?

All predecessors are computed, and termination guaranteed

But: we might be adding non-predecessors

$$\Delta = \{ \langle p_0, \gamma_0 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \; , \; \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_0, \epsilon \rangle \; , \; \langle p_1, \gamma_1 \rangle \hookrightarrow \langle p_1, \gamma_1 \gamma_0 \rangle \}$$



Fortunately: correct if initial states have no incoming arcs

Repeated reachability for pushdown systems

Let $I = \langle p_0, \gamma_0 \rangle$ and $F = \langle p, \Gamma^* \rangle$

F can be repeatedly reached from I iff

$$\langle p_0, \gamma_0 \rangle \longrightarrow^* \langle p', \gamma w \rangle$$

and

$$\langle p', \gamma \rangle \longrightarrow^* \langle p, v \rangle \longrightarrow^* \langle p', \gamma u \rangle$$

for some p', γ, w, v, u

Repeated reachability can be reduced to computing several pre^*

Applications

Algorithms for pre^* and $post^*$ developed in [E. et al., CAV'00, CAV'01]
BDD technology to deal with variables

Implemented in the Moped model-checker

Used as replacement of Bebop in the SLAM project

Experimental results (by Schwoon) on

Test suite of 64 C-programs

Four drivers with between 2200 and 7600 lines of code

A serial driver with 27000 lines of code

For the drivers: locking-unlocking properties checked or bugs found in between 1 and 2 minutes

A general acceleration framework

Compute a symbolic reachability graph with elements of $\mathcal{C}$ as nodes:

Add I as first node

For each node C and each transition t, add an edge $C \xrightarrow{t} post[t](C)$

Replace $C \xrightarrow{\sigma} post[\sigma](C)$ by $C \xrightarrow{\sigma} X$, where X satisfies

(1) $post[\sigma](C) \subseteq X$, and

(2) X contains only reachable configurations

Acceleration through loops

A loop is a sequence $C \xrightarrow{\sigma} post[\sigma](C)$ such that

$$C \xrightarrow{\sigma} post[\sigma](C) \xrightarrow{\sigma} post[\sigma^2](C) \xrightarrow{\sigma} post[\sigma^3](C) \cdots$$

Examples: $c \xrightarrow{\sigma} c' \geq c$ in broadcast protocols

$\langle p, \gamma \rangle \xrightarrow{\sigma} \langle p, \gamma v \rangle$ in pushdown systems

Acceleration: given a loop $C \xrightarrow{\sigma} post[\sigma](C)$, replace $post[\sigma](C)$ by

$$X = post[\sigma^*](C) = C \cup post[\sigma](C) \cup post[\sigma^2](C) \cup \ldots$$

Problem: find a suitable class of loops such that $post[\sigma^*](C)$ belongs to $\mathcal{C}$

Other models

Counter machines [Boigelot and Wolper, CAV'94, LNCS 818]

Configuration: pair $(q, n_1, \ldots, n_k)$, where q state $n_1, \ldots, n_k$ integers

Family $\mathcal{C}$: Presburger sets

Suitable loops: syntactically defined

FIFO-automata with lossy channels [Abdulla et al, CAV'98, LNCS 1427]

Configuration: pair $(q, \mathbf{w})$, where s state and $\mathbf{w}$ vector of words representing the contents of the queues

Family $\mathcal{C}$: regular sets represented by simple regular expressions

Suitable loops: any

FIFO-automata with perfect channels [Boigelot and Godefroid, CAV'96, LNCS 1102], [Bouajjani and Habermehl, ICALP'97, LNCS 1256]

Arrays of parallel processes [Bouajjani et al, CAV'00, LNCS 1855]

Widenings

Accurate widenings

Replace $C \xrightarrow{\sigma} post[a](C)$ by $C \xrightarrow{\sigma} X$, where X satisfies

(1) $post[a](C) \subseteq X$, and

(2') X contains only reachable final configurations

Notice that X may contain unreachable non-final configurations!

Inaccurate widenings

Replace $C \xrightarrow{\sigma} post[a](C)$ by $C \xrightarrow{\sigma} X$, where X satisfies

$$(1) \quad post[a](C) \subseteq X$$

If no configuration of the graph belongs to F, then no reachable configuration belongs to F

If some configuration of the graph belongs to F, no information is gained

Accurate widenings in broadcast protocols

Fact: $post[\sigma](p) = T_\sigma(p)$ for a linear transformation $T_\sigma(p) = M_\sigma \cdot x + b_\sigma$

It follows: $post[\sigma^*](p) = \bigcup_{n \geq 0} T_\sigma^n(p)$

However, $post[\sigma^*](p)$ may not be a parametric configuration

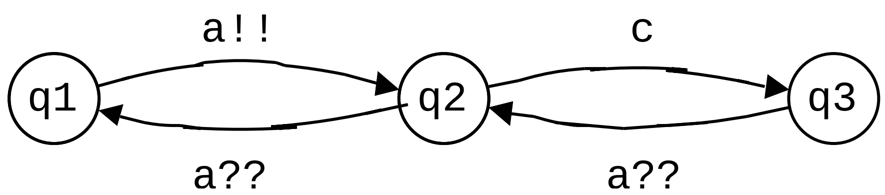
Accurate widening: widen $post[\sigma^*](p)$ to $lub\{T_\sigma^n(p) \mid n \geq 0\}$

Theorem: if the set F is upward-closed, this widening is accurate

Does widening lead to termination?

For arbitrary broadcast protocols: NO!   [E. et al, LICS'99]

Example in which the acceleration doesn't have any effect:



$$p_0 = (\sqcup, 0, 0)$$

For rendezvous communication only: YES
[Karp and Miller '69], [German and Sistla, JACM 39(3), 1992]

Conclusions

Decidability analysis very advanced

Many algorithms useful in practice

Many prototype implementations, some tools

The ADVANCE project:
  Advanced Verification Techniques for Telecommunication Protocols

Challenges:

- systems with several sources of infinity (automata-theoretic techniques)

- connection to program analysis