

Reachability in Pushdown Systems

Deepak D'Souza

Department of Computer Science and Automation
Indian Institute of Science, Bangalore.

03 November 2016

Outline

- 1 Modelling Programs as Pushdown Systems
- 2 Definitions
- 3 Reachability in Pushdown Systems
- 4 Saturation algorithm for Pre^*
- 5 Correctness of saturation algo

Modelling programs with procedures

```
0 void main() {
1     int a;
2     a = fact(3);
3 }
4
5 int fact(int x) {
6     int res;
7     if (x==1)
8         res = 1;
9     else
10        res = x * fact(x-1);
11    return res;
12 }
```

	arg	3
ret	addr	2
	a	0

After call to fact in line 2

arg	2
ret addr	8
res	0
x	3
ret addr	2
a	0

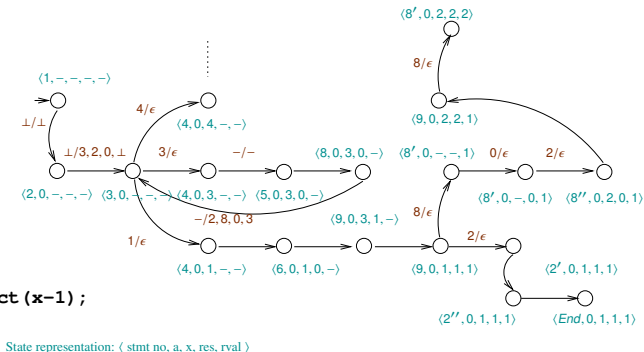
After call to fact in line 8

Pushdown system induced by program

```

0 void main() {
1     int a;
2     a = fact(3);
3 }
4
3 int fact(int x) {
4     int res;
5     if (x==1)
6         res = 1;
7     else
8         res = x * fact(x-1);
9     return res;
10 }

```



Pushdown Systems

A **pushdown system** is of the form

$$\mathcal{P} = (P, \Gamma, \Delta)$$

where

- P is a finite set of states
- Γ is the stack alphabet,
- $\Delta \subseteq P \times \Gamma \times P \times \Gamma^*$ is the non-deterministic transition relation.
 - Each transition is of the form $pa \rightarrow q\gamma$.

A pushdown system is thus like a PDA but with no input and no initial/final states.

Can model several useful classes of systems

- PDA with input abstracted away
- Programs with finite state but with procedure calls (or “Boolean Programs”)

Example Pushdown System

Example pushdown system $\mathcal{P}_1$

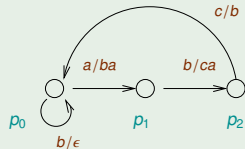
$$p_0 a \rightarrow p_1 ba$$

$$p_1 b \rightarrow p_2 ca$$

$$p_2 c \rightarrow p_0 b$$

$$p_0 b \rightarrow p_0 \epsilon.$$

Diagram representation



Sequence of configurations reachable from $p_2 cbba$:

$$p_2 cbba \xRightarrow{1} p_0 bbba \xRightarrow{1} p_0 bba \xRightarrow{1} p_0 ba \xRightarrow{1} p_0 a \xRightarrow{1} p_1 ba \xRightarrow{1} p_2 caa \xRightarrow{1} p_0 baa$$

Configuration graph induced by a pushdown system

$\mathcal{P}$ induces a (possibly infinite) graph whose

- nodes are configurations of $\mathcal{P}$ represented by strings in $P \cdot \Gamma^*$
- edges are $c \rightarrow c'$ iff $c \xRightarrow{1} c'$ in $\mathcal{P}$.

Given a set of configurations C of $\mathcal{P}$ we can define

$$Pre^*(C) = \{c \mid \exists c' \in C : c \xRightarrow{*} c'\}.$$

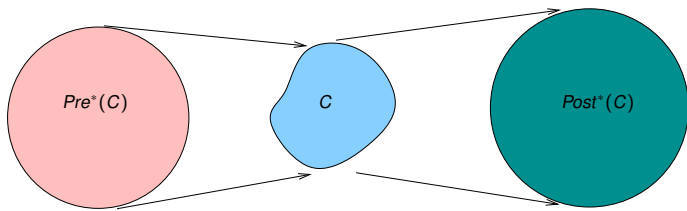
And similarly

$$Post^*(C) = \{c' \mid \exists c \in C : c \xRightarrow{*} c'\}.$$

Reachability in Pushdown Systems

Theorem (Büchi, 1964)

Let $\mathcal{P}$ be a pushdown system, and let C be a regular set of configurations of $\mathcal{P}$. Then $Pre^(C)$ and $Post^*(C)$ are also regular sets. Moreover given an NFA for C we can construct an NFA accepting $Pre^*(C)$ and $Post^*(C)$ respectively.*



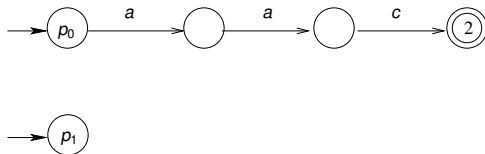
Saturation algorithm for Pre^*

Let $\mathcal{P} = (P, \Gamma, \Delta)$ be a pushdown system, and C be a set of configurations of $\mathcal{P}$.

A **P -automaton** for C is an NFA $\mathcal{A} = (Q, \Gamma, P, \Delta', F)$ that accepts from an initial state $p \in P$ exactly the words w such that $pw \in C$.

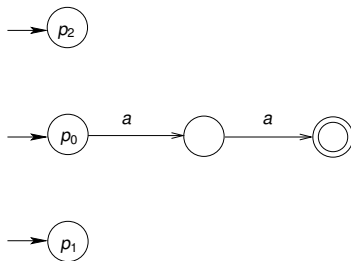
- The control states of $\mathcal{P}$ are used as initial states of $\mathcal{A}$.
- $\mathcal{A}$ must not have a transition to an initial state.

Example P -automaton for $\{p_0aac\}$:



Example P -automaton

Example P -automaton for $\{p_0aa\}$:

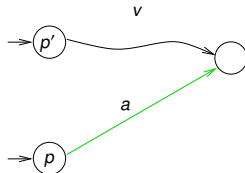


Saturation algo for Pre^*

Input: Pushdown system $\mathcal{P} = (P, \Gamma, \Delta)$, and P -automaton $\mathcal{A}$ for C .

Output: $\overline{\mathcal{A}}$ accepting $Pre^*(C)$.

- Repeat until no more new edges can be added to $\mathcal{A}$:
 - If $pa \rightarrow p'v \in \Delta$ and $p' \xrightarrow{v} q$ in $\mathcal{A}$, then add $p \xrightarrow{a} q$ to $\mathcal{A}$.



- Return $\overline{\mathcal{A}}$.

Run saturation algo for Pre^*

Example pushdown system

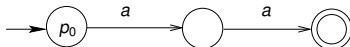
$$p_0 a \rightarrow p_1 b a$$

$$p_1 b \rightarrow p_2 c a$$

$$p_2 c \rightarrow p_0 b$$

$$p_0 b \rightarrow p_0 \epsilon.$$

P -automaton for $C = \{p_0 aa\}$:



Run saturation algo for Pre^*

Example pushdown system

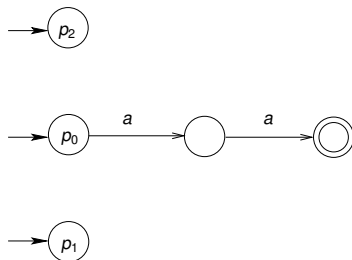
$$p_0 a \rightarrow p_1 b a$$

$$p_1 b \rightarrow p_2 c a$$

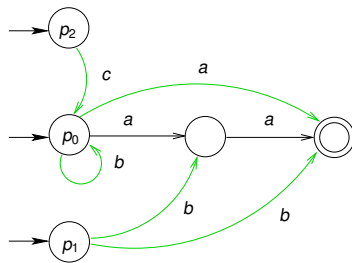
$$p_2 c \rightarrow p_0 b$$

$$p_0 b \rightarrow p_0 \epsilon.$$

P -automaton for $C = \{p_0 a a\}$:

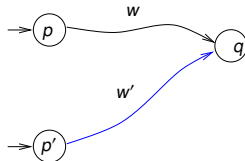


Saturated P -automaton:



Correctness

- $Pre^*(C) \subseteq L(\overline{\mathcal{A}})$.
 - Prove by induction on n that if $pw \xRightarrow{n} p'w' \in C$ then $pw \in L(\overline{\mathcal{A}})$.
- $L(\overline{\mathcal{A}}) \subseteq Pre^*(C)$.
 - Let $\mathcal{A}_i$ be P -automaton after i -th step of algo.
 - Claim 1: If $pw \in L(\mathcal{A}_i)$ then $pw \xRightarrow{*} p'w' \in Pre^*(C)$.
 - Proof by induction on i gets into rough weather.
 - Strengthen Claim to: If $p \xrightarrow{w} q$ in $\mathcal{A}_i$ then there exists $p'w'$ such that $p' \xrightarrow{w'} q$ in $\mathcal{A}$ and $pw \xRightarrow{*} p'w'$.

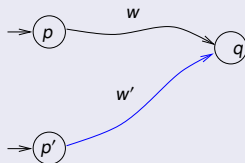


- Observe that strengthened Claim implies Claim 1 and completes proof.

Proof of Claim

Claim

If $p \xrightarrow{w} q$ in $\mathcal{A}_i$ then there exists $p'w'$ such that $p' \xrightarrow{w'} q$ in $\mathcal{A}$ and $pw \stackrel{*}{\Rightarrow} p'w'$.



Proof: By induction on i . For the induction step, suppose we added the edge (p_1, a, q_1) in $\mathcal{A}_{i+1}$ due to the PDA transition $p_1 a \rightarrow p_2 v$. Suppose $p \xrightarrow{w} q$ in $\mathcal{A}_{i+1}$.

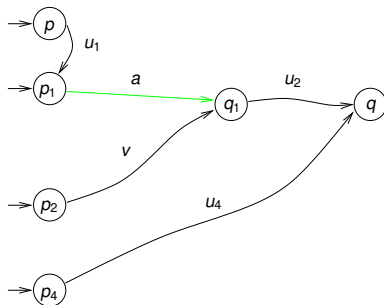
Proof of Claim - II

If this path does not use the new edge, it is a path in $\mathcal{A}_i$ itself and by induction hypothesis we are done.

If it uses the new edge 1 or more times consider the representative case of when it uses it exactly once. Say the path is

$$p \xrightarrow{u_1} p_1 \xrightarrow{a} q_1 \xrightarrow{u_2} q.$$

- By IH there has to be a path $p_3 u_3$ to p_1 in $\mathcal{A}$ such that $pu_1 \xRightarrow{*} p_3 u_3$. But since $\mathcal{A}$ has **no** incoming edges to the P -states, we must have $p_3 = p_1$ and $u_3 = \epsilon$. So $pu_1 \xRightarrow{*} p_1$.
- By IH we also have a path $p_4 u_4$ to q in $\mathcal{A}$ such that $p_2 v u_2 \xRightarrow{*} p_4 u_4$.
- Putting these together: $pw = pu_1 a u_2 \xRightarrow{*} p_1 a u_2 \xrightarrow{1} p_2 v u_2 \xRightarrow{*} p_4 u_4$, and $p_4 u_4$ is the required $p'w'$.



Proof of Claim - III (General argument)

By a second induction on the number of times the path $p \xrightarrow{w} q$ uses the edge $p_1 \xrightarrow{a} q_1$, we prove that $\exists p'w'$ such that $pw \xRightarrow{*} p'w'$ and $p' \xrightarrow{w'} q$ in $\mathcal{A}$. **Base:** If w does not use the new edge, it is a path in $\mathcal{A}_i$ itself and by IH-1 we are done. **Ind-step:** If it uses the new edge $k + 1$ times, let the path be $p \xrightarrow{u_1} p_1 \xrightarrow{a} q_1 \xrightarrow{u_2} q$, where u_2 does not use the new edge.

- Since $p \xrightarrow{u_1} p_1$ uses the new edge k times, by IH-2 there is a path p_3u_3 to p_1 in $\mathcal{A}$ such that $pu_1 \xRightarrow{*} p_3u_3$. But since $\mathcal{A}$ has **no** incoming edge to P -states, we have $p_3 = p_1$ and $u_3 = \epsilon$. So $pu_1 \xRightarrow{*} p_1$.
- By IH-1 we also have a path p_4u_4 to q in $\mathcal{A}$ such that $p_2vu_2 \xRightarrow{*} p_4u_4$.
- Thus: $pw = pu_1au_2 \xRightarrow{*} p_1au_2 \xRightarrow{1} p_2vu_2 \xRightarrow{*} p_4u_4$, and p_4u_4 is the required $p'w'$.

