COMS3261: Computer Science Theory

Fall 2013

Mihalis Yannakakis

Lecture 12, 10/14/13

Acceptance by empty stack

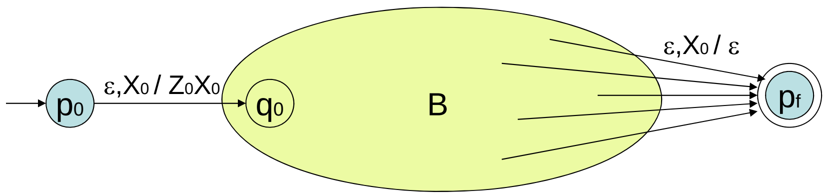
- Different notion of acceptance by PDA: when stack is empty (whence PDA stops and accepts – no need for F)
- $N(A) = \{ w \in \Sigma^* \mid (q_0,w,Z_0) \vdash^* (p,\varepsilon,\varepsilon) \text{ for some } p \in Q \}$

- For a specific PDA A, the language L(A) (acceptance by final state) and N(A) (acceptance by empty stack) not same
- But expressive power of two styles of acceptance is same:

- Theorem: A language L is = L(A) for some PDA A iff L=N(B) for some PDA B

Empty stack to Final state

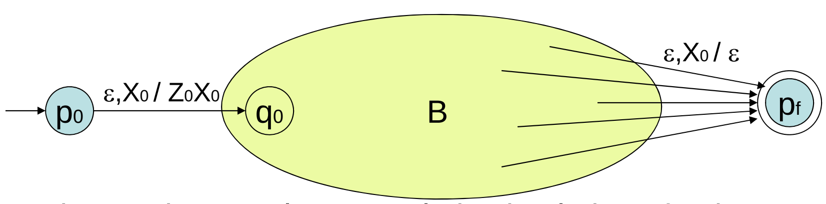
- Given PDA $B=(Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ construct another PDA A such that L(A) = N(B)
- Add new start stack symbol X_0, start state p_0, final state p_f to get new PDA A $=(Q\cup\{p_0, p_f\}, \Sigma, \Gamma\cup\{X_0\}, \delta, p_0, X_0, \{p_f\})$



• Starting from p_0, X_0, add Z_0 to stack and move to q_0 to simulate B. If B empties its stack ($\Leftrightarrow$ stack of A $=X_0$) transition to p_f

Cost of translation: linear

Empty stack to Final state - Proof



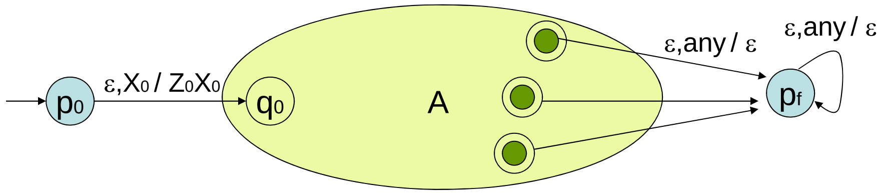
• Observation: X_0 plays no role in simulation of B (never pops Z_0 to look at it). Therefore, if (q_0,w,Z_0) $|\text{--}^*_B$ $(q,\varepsilon,\varepsilon)$ then
(p_0,w,X_0) $|\text{--}_A$ (q_0,w,Z_0X_0) $|\text{--}^*_A$ (q,ε,X_0) $|\text{--}_A$ (p_f,ε,X_0)

• If (p_0,w,X_0) $|\text{--}_A$ $(p_f,\varepsilon, ..)$ then
(p_0,w,X_0) $|\text{--}_A$ (q_0,w,Z_0X_0) $|\text{--}^*_A$ (q,ε,X_0) $|\text{--}_A$ (p_f,ε,X_0) (for some q)
$\Rightarrow$ (q_0,w,Z_0) $|\text{--}^*_B$ $(q,\varepsilon,\varepsilon)$ (because X_0 irrelevant)

$\Rightarrow$ L(A) =N(B)

Final state to Empty stack

- Given PDA A = $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ construct another PDA B such that L(A) = N(B)
- Construct B = $(Q \cup \{p_0, p_f\}, \Sigma, \Gamma \cup \{X_0\}, \delta, p_0, X_0)$



- Once A reaches an accepting state, can transition to state p_f and empty the stack

PDAs ≡ CFGs

Theorem: Context-free Languages = (by definition)
= Languages generated by CFGs
= Languages accepted by PDAs.

1. Given a CFG, we can construct a PDA that accepts the same language.
2. Given a PDA, we can construct a CFG that generates the same language.

CFG $\Rightarrow$ PDA

- Given CFG G=(V,T,P,S), construct empty-stack PDA M= ({q}, T, Γ=V$\cup$T, δ, q, S)

  - note: just 1 state q (so can omit in transition function)
- PDA M simulates a leftmost derivation of input string.
- For each production $A \rightarrow \beta$ of G, PDA has transition $(q,\varepsilon,A) \rightarrow (q,\beta)$
- In addition transitions $(q,a,a) \rightarrow (q, \varepsilon)$ (pop) for all $a \in T$
- Operation of PDA:
- If top of stack=variable then apply a production

  if top = terminal, then match with input symbol

Example: CFG for arithmetic expressions

- T = { a,b,0,1,+,*,(,)}
- V={E,I}, S=E
- Productions

  $E \rightarrow I \mid E{+}E \mid E{*}E \mid (E)$

  $I \rightarrow a \mid b \mid Ia \mid Ib \mid I0 \mid I1$

Example: CFG for arithmetic expressions

Input string a*b

leftmost derivation		PDA accepting computation
	E	(q, a*b, E)
	E*E	(q, a*b, E*E)
	I*E	(q,a*b, I*E)
	a*E	(q, a*b, a*E)
		(q, *b, *E)
		(q, b, E)
	a*I	(q, b, I)
	a*b	(q, b, b)
		(q, ε, ε)

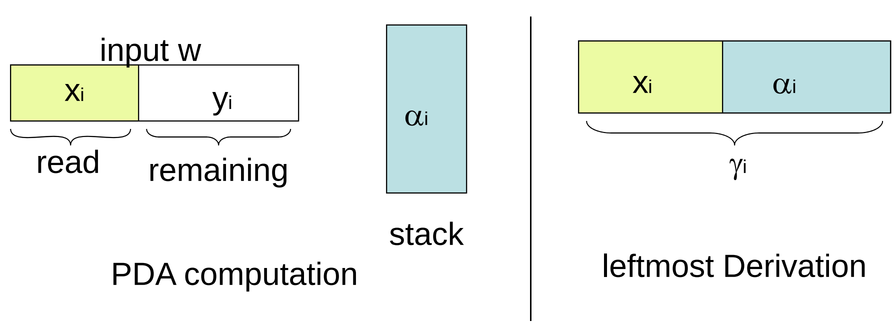
Theorem: L(G) = N(M)

$L(G) \subseteq N(M)$: Let $w \in L(G)$, consider a leftmost derivation of w:

$S = \gamma_1 \Rightarrow \gamma_2 \Rightarrow \ldots \Rightarrow \gamma_m = w$.

Each $\gamma_i = x_i \alpha_i$ where x_i is a prefix of w, i.e. $w = x_i y_i$, and α_i starts with a variable, i.e. $\alpha_i = A_i \sigma_i$ for some σ_i (except $\alpha_m = \varepsilon$)

Prove by induction on i that $(q,w,S) \vdash^* (q,y_i,\alpha_i)$



Proof ctd. $L(G) \subseteq N(M)$

$L(G) \subseteq N(M)$: Prove by induction on i that $(q,w,S) \vdash^* (q,y_i,\alpha_i)$

- Basis: $i=1 \Rightarrow \gamma_1 = S$, $x_1 = \varepsilon$, $\alpha_1 = S$, $y_1 = w$
- Induction step:

$\gamma_i = x_i\ \alpha_i = x_i\ A_i\ \sigma_i \Rightarrow_{lm} \gamma_{i+1} = x_i\ z\ \beta\ \sigma_i$ by a production $A_i \rightarrow z\ \beta$ where $z \in T^*$ and β starts with a variable or $=\varepsilon$

Note $x_i z$ is a prefix of $w = x_i\, y_i$, so z is a prefix of y_i

- Then PDA applies transition $(q,y_i, A_i\sigma_i) \vdash (q,y_i, z\beta\sigma_i)$ and then makes a sequence of moves that consumes z from the remaining input y_i, while popping it from the stack, so $\vdash^* (q,y_{i+1}, \beta\sigma_i)$ and $\alpha_{i+1} = \beta\sigma_i$

Proof : $N(M) \subseteq L(G)$

- Show that if $(q,x,A) \vdash^* (q,\varepsilon,\varepsilon)$ then $A \Rightarrow^* x$

By induction on the length of the PDA computation

Basis: 1 step. If $x=\varepsilon$ then production $A \rightarrow \varepsilon$

Induction: n>1 steps. $(q,x,A) \vdash (q,x,Y_1Y_2\ldots Y_k) \vdash \ldots \vdash (q,\varepsilon,\varepsilon)$

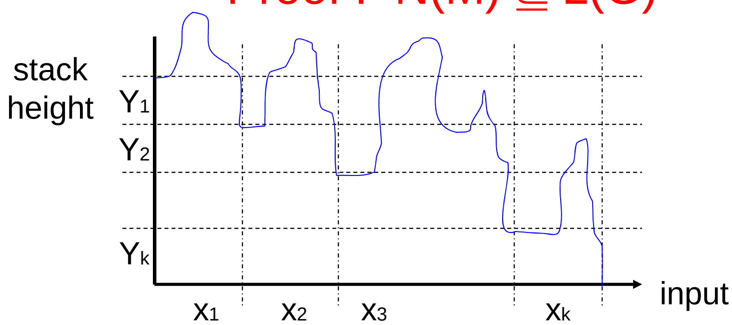
G has production $A \rightarrow Y_1Y_2\ldots Y_k$

Look at first time Y_1 is popped and Y_2 top : x_1 input prefix read

… first time Y_i is popped and Y_{i+1} top : $x_1\ldots x_i$ input prefix read

(If $Y_i \in T$ then $x_i = Y_i$, and stack popped & input symbol consumed in the next step)

Proof : $N(M) \subseteq L(G)$



$(q,x_1x_2..x_k,A)$ |--$(q,x_1x_2..x_k, Y_1Y_2..Y_k)$ |--* $(q,\varepsilon,\varepsilon)$

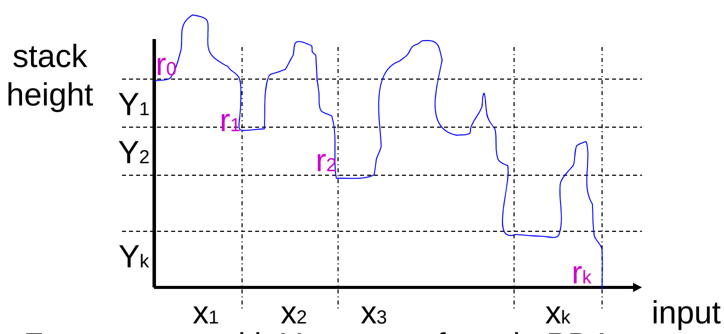
(q,x_1,Y_1) |-- $(q,\varepsilon,\varepsilon) \Rightarrow Y_1 \Rightarrow^* x_1$ by i.h. (or $Y_1=x_1$ if in T)

(q,x_2,Y_2) |-- $(q,\varepsilon,\varepsilon) \Rightarrow Y_2 \Rightarrow^* x_2$ by i.h. (or $Y_2=x_2$ if in T), …

$\therefore A \Rightarrow Y_1Y_2..Y_k)$ |--* $x_1x_2..x_k = x$

PDA to CFG

Picture how a PDA consumes an input x and empties a stack $Y_1Y_2..Y_k$



- From state r_0 with Y_1 on top of stack, PDA consumes x_1, pops Y_1, moves to state r_1 ,…..
- From state r_{i-1} with Y_i on top of stack, PDA consumes x_i, pops Y_i, moves to state r_i , for i=1,…, k

PDA $\Rightarrow$ CFG

- Given PDA $M=(Q, \Sigma, \Gamma, \delta, q_0, Z_0)$ that accepts by empty stack. Construct grammar $G=(V, \Sigma,P,S)$,
- Variables $V = \{ [qXp] \mid p,q \in Q, X \in \Gamma \} \cup \{S\}$
- meaning of $[qXp]$: derives strings x with property that $(q,x,X) \vdash^* (p,\varepsilon,\varepsilon)$
- Productions:
- $S \rightarrow [q_0 Z_0 p]$, for all $p \in Q$

  (Reason: if $(q_0, x, Z_0) \vdash^* (p,\varepsilon,\varepsilon)$ want $S \Rightarrow^* x$, so that $x \in L(G)$)
- For every transition of PDA $(q,a,X) \rightarrow (r, Y_1 Y_2 .. Y_k)$, where $a=\varepsilon$ or $a \in \Sigma$, for every k-tuple of states $r_1, r_2, .., r_k$, we have production: $[q,X,r_k] \rightarrow a\ [r\ Y_1\ r_1]\ [r_1 Y_2\ r_2]\ \ldots\ [r_{k-1} Y_k\ r_k]$

Example of transition $\rightarrow$ production

- popping transition: $(q,a,X) \rightarrow (r,\varepsilon)$

  yields production: $[qXr] \rightarrow a$

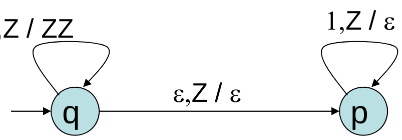
- change-symbol transition: $(q,a,X) \rightarrow (r,Y)$

  yields productions: $[qXr_1] \rightarrow a\ [rYr_1]$ for all $r_1 \in Q$

- change and push transition: $(q,a,X) \rightarrow (r,Y_1Y_2)$

  yields productions: $[qXr_2] \rightarrow a\ [rY_1r_1]\ [r_1Y_2\ r_2]$ for all $r_1, r_2 \in Q$

Example of translation



N(M)= { 0^n1^n| n≥0 }

- S → [q Zq] | [q Zp]
- Self-loop of q:
  [qZq] → 0 [qZq] [qZq] | 0 [qZp] [pZq]
  [qZp] → 0 [qZq] [qZp] | 0 [qZp] [pZp]
- Transition q → p: [qZp] → ε
- Self-loop at p: [pZp] → 1

Proof that L(G) = N(M)

- L(G) ⊇ N(M) : Show by induction on length of a computation of M that if (q,w,X) |--* (p,ε,ε) then [qXp] ⇒* w

  Since we have productions S → [q_0 Z_0 p], for all p∈Q it follows that if (q_0,w,Z_0) |--* (p,ε,ε) then S ⇒*w , and w∈L(G)

- L(G) ⊆ N(M): Show by induction on length of a derivation that if [qXp] ⇒* w then (q,w,X) |--* (p,ε,ε)

  Since a derivation S ⇒*w must be S⇒ [q_0 Z_0 p] ⇒*w for some p∈Q, it follows that (q_0,w,Z_0) |--* (p,ε,ε) and w ∈N(M)

Cost of translations

- CFG to PDA : linear

- PDA to CFG:

  #variables = $|Q|^2 \times |\Gamma|$

  #productions $\leq$ #transitions $\times$ $|Q|^{k\text{-max}}$ where k-max = max length of string put on stack in a PDA transition

- If PDA transitions put many symbols at once on stack then exponential

- But can transform to equivalent PDA that puts always ≤ 2 symbols on stack $\Rightarrow$ #productions $\leq$ #transitions $\times$ $|Q|^2$

  $\Rightarrow$ Polynomial blow-up. If length of description of PDA is n (includes $|Q|$, $|\Gamma|$, sum of lengths of transitions), then length of description of CFG is $O(n^3)$

Restricted PDAs

- Restricting PDAs to only pop, change top or push single symbol does not change expressive power
- For each transition $(q,a,X) \to (p,\alpha)$ where $a=\varepsilon$ or is in Σ and $\alpha = Y_k \ldots Y_1$. If k=0 or 1, nothing to do (pop or change)
- If $k \geq 2$, introduce k-1 new states $r_1,\ldots,r_{k-1}$ and transitions:

  $(q,a,X) \to (r_1,Y_1)$ (change top to Y_1)

  $(r_1,\varepsilon,Y_1) \to (r_2,Y_2Y_1)$ (push Y_2)

  … $(r_i,\varepsilon,Y_i) \to (r_{i+1},Y_{i+1}Y_i)$ (push Y_{i+1})

  $(r_{k-1},\varepsilon,Y_{k-1}) \to (p,Y_kY_{k-1})$ (push Y_k)

Then $(q,a,X) \mid\!-\!- (r_1,\varepsilon,Y_1) \mid\!-\!- (r_2,\varepsilon,Y_2Y_1) \mid\!-\!- \ldots \mid\!-\!- (p,\varepsilon,\alpha)$

(If we had allowed moves on ε stack, then push, pop enough)