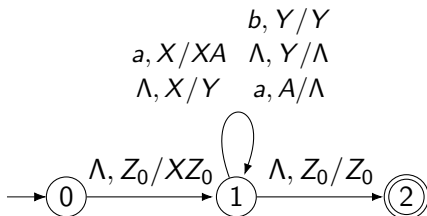
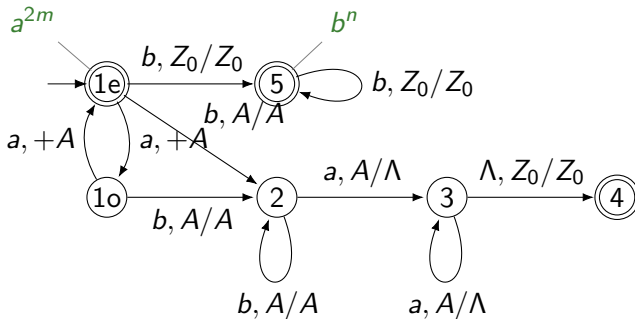
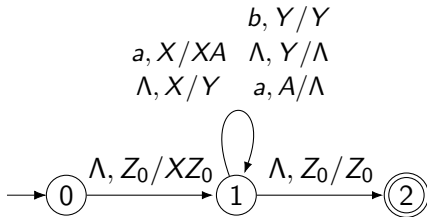


- Eind deze week: huiswerkopgave 3



$$a^m b^n a^m \quad m, n \geq 0$$



The first PDA in the previous slide (about $a^m b^n a^m$) is not deterministic. Actually it is working like a grammar: in state 1 the following productions are simulated:

$$X \rightarrow aXA \mid Y$$

$$Y \rightarrow bY \mid \Lambda$$

$$A \rightarrow a$$

The second automaton is deterministic. We have to distinguish the cases where $m = 0$ (state 5) and $n = 0$ (states 1e and 1o).

$$L = \{ a^i b^j \mid i \neq j \}$$

$$S \rightarrow X \mid Y \quad (\text{choice!})$$

$$X \rightarrow aXb \mid aX \mid a \quad (i > j)$$

$$Y \rightarrow aYb \mid Yb \mid b \quad (i < j)$$

$$S \Rightarrow X \Rightarrow aXb \Rightarrow aaXb \Rightarrow aaaXbb \Rightarrow aaaabb$$

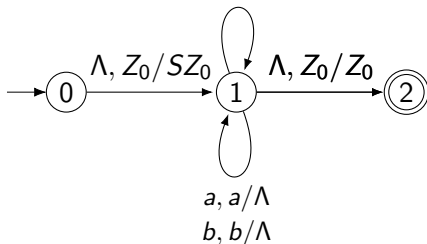
$$L = \{ a^i b^j \mid i \neq j \}$$

$$S \rightarrow X \mid Y \quad (\text{choice!})$$

$$X \rightarrow aXb \mid aX \mid a \quad (i > j)$$

$$Y \rightarrow aYb \mid Yb \mid b \quad (i < j)$$

$$\begin{array}{ll} \Lambda, S/X & \Lambda, S/Y \\ \Lambda, X/aXb & \Lambda, Y/aYb \\ \Lambda, X/aX & \Lambda, Y/Yb \\ \Lambda, X/a & \Lambda, Y/b \end{array}$$



CFG $G = (V, \Sigma, S, P)$

Definition (Nondeterministic Top-Down PDA)

$NT(G) = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$, as follows:

- $Q = \{q_0, q_1, q_2\}$
- $A = \{q_2\}$
- $\Gamma = V \cup \Sigma \cup \{Z_0\}$
- start $\delta(q_0, \Lambda, Z_0) = \{(q_1, SZ_0)\}$
- **expand** $\delta(q_1, \Lambda, A) = \{(q_1, \alpha) \mid A \rightarrow \alpha \text{ in } P\}$ for $A \in V$
- **match** $\delta(q_1, \sigma, \sigma) = \{(q_1, \Lambda)\}$ for $\sigma \in \Sigma$
- finish $\delta(q_1, \Lambda, Z_0) = \{(q_2, Z_0)\}$ check empty stack

[M] Def 5.17

From lecture 7:

$$AEqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$$

aaabbb, ababab, aababb, ...

$$S \rightarrow \Lambda \mid aB \mid bA$$

$$A \rightarrow aS \mid bAA$$

$$B \rightarrow bS \mid aBB$$

A generates $n_a(x) = n_b(x) + 1$

B generates $n_a(x) + 1 = n_b(x)$

$S \Rightarrow aB \Rightarrow aa\textcolor{teal}{B}B \Rightarrow aa\textcolor{teal}{b}S\textcolor{blue}{B} \Rightarrow \dots$ (different options)

(1) $aa\textcolor{teal}{b}B \Rightarrow aa\textcolor{teal}{b}aBB \Rightarrow aa\textcolor{teal}{b}abSB \Rightarrow aa\textcolor{teal}{b}abB \Rightarrow aa\textcolor{teal}{b}abbS \Rightarrow aa\textcolor{teal}{b}abb$

(2) $aa\textcolor{teal}{b}aBB \Rightarrow aa\textcolor{teal}{b}abSB \Rightarrow aa\textcolor{teal}{b}abB \Rightarrow aa\textcolor{teal}{b}abbS \Rightarrow aa\textcolor{teal}{b}abb$

(2') $aa\textcolor{teal}{b}aBB \Rightarrow aa\textcolor{teal}{b}aBbS \Rightarrow aa\textcolor{teal}{b}abSbS \Rightarrow aa\textcolor{teal}{b}abSb \Rightarrow aa\textcolor{teal}{b}abb$

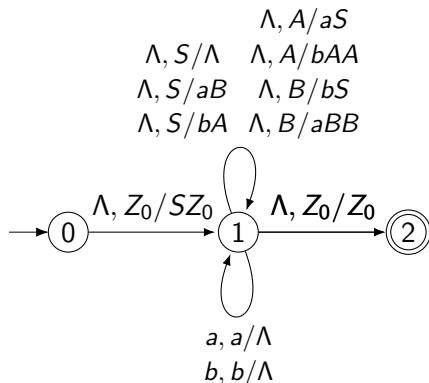
[M] E 4.8

$$AEqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$$

$$S \rightarrow \Lambda \mid aB \mid bA$$

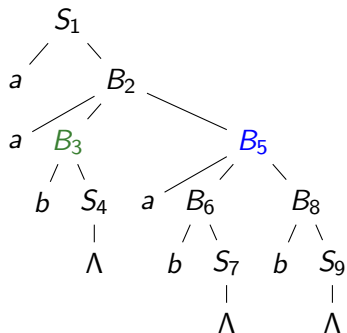
$$A \rightarrow aS \mid bAA$$

$$B \rightarrow bS \mid aBB$$

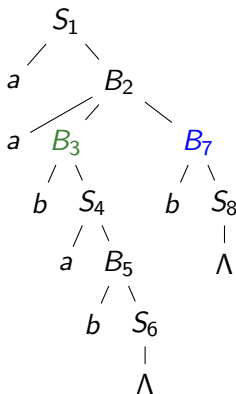


Derivation tree & leftmost derivations

From lecture 7:



$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow$
 $aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow$
 $aababbB \Rightarrow aababbS \Rightarrow aababb$



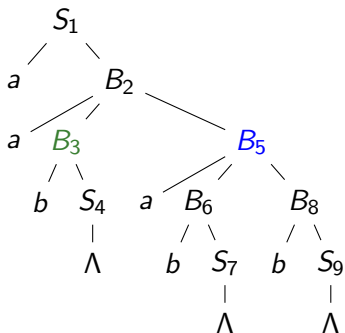
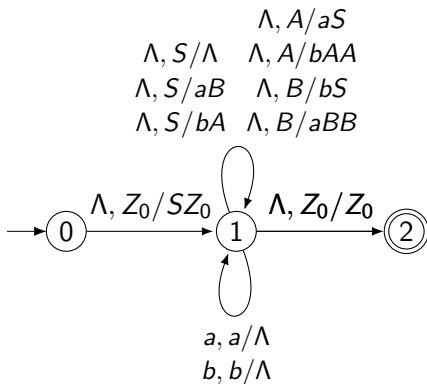
$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow$
 $aabaBB \Rightarrow aababSB \Rightarrow aababbB \Rightarrow$
 $aababbS \Rightarrow aababb$

$$AEqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$$

$$S \rightarrow \Lambda \mid aB \mid bA$$

$$A \rightarrow aS \mid bAA$$

$$B \rightarrow bS \mid aBB$$



q_0 $aababbb$ Z_0

...

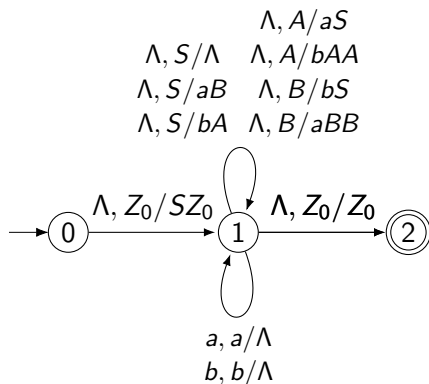
Top-down = expand-match

$$AEqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$$

$$S \rightarrow \Lambda \mid aB \mid bA$$

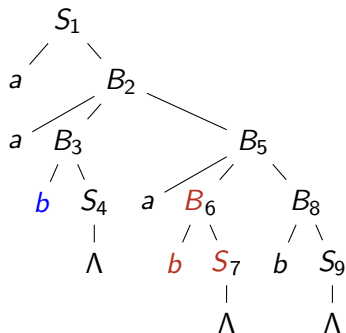
$$A \rightarrow aS \mid bAA$$

$$B \rightarrow bS \mid aBB$$



q_0	$aababb$	Z_0	
q_1	$aababb$	$S Z_0$	1 : $S \rightarrow aB$
q_1	$aababb$	$aB Z_0$	match a
q_1	$a ababb$	$B Z_0$	2 : $B \rightarrow aBB$
q_1	$a ababb$	$aBB Z_0$	match a
q_1	$aa babb$	$BB Z_0$	3 : $B \rightarrow bS$
q_1	$aa babb$	$bSB Z_0$	match b
q_1	$aab abb$	$SB Z_0$	4 : $S \rightarrow \Lambda$
q_1	$aab abb$	$B Z_0$	5 : $B \rightarrow aBB$
q_1	$aab abb$	$aBB Z_0$	match a
q_1	$aaba bb$	$BB Z_0$	6 : $B \rightarrow bS$
q_1	$aaba bb$	$bSB Z_0$	match b
q_1	$aabab b$	$SB Z_0$	7 : $S \rightarrow \Lambda$
q_1	$aabab b$	$B Z_0$	8 : $B \rightarrow bS$
q_1	$aabab b$	$bS Z_0$	match b
q_1	$aababb$	$S Z_0$	9 : $S \rightarrow \Lambda$
q_1	$aababb$	Z_0	
q_2	$aababb$	Z_0	

Top-down = expand-match

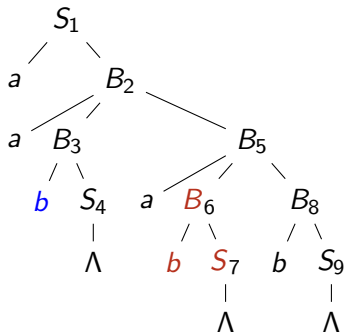


preorder: leftmost

$S \xRightarrow{\ell} aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow$
 $aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow$
 $aababB \Rightarrow aababbS \Rightarrow aababb$

q_0	<i>aababb</i>	Z_0	
q_1	<i>aababb</i>	$S Z_0$	1 : $S \rightarrow aB$
q_1	<i>aababb</i>	$aB Z_0$	match a
q_1	<i>a ababb</i>	$B Z_0$	2 : $B \rightarrow aBB$
q_1	<i>a ababb</i>	$aBB Z_0$	match a
q_1	<i>aa babb</i>	$BB Z_0$	3 : $B \rightarrow bS$
q_1	<i>aa babb</i>	$bSB Z_0$	match b
q_1	<i>aab abb</i>	$SB Z_0$	4 : $S \rightarrow \Lambda$
q_1	<i>aab abb</i>	$B Z_0$	5 : $B \rightarrow aBB$
q_1	<i>aab abb</i>	$aBB Z_0$	match a
q_1	<i>aaba bb</i>	$BB Z_0$	6 : $B \rightarrow bS$
q_1	<i>aaba bb</i>	$bSB Z_0$	match b
q_1	<i>aabab b</i>	$SB Z_0$	7 : $S \rightarrow \Lambda$
q_1	<i>aabab b</i>	$B Z_0$	8 : $B \rightarrow bS$
q_1	<i>aabab b</i>	$bS Z_0$	match b
q_1	<i>aababb</i>	$S Z_0$	9 : $S \rightarrow \Lambda$
q_1	<i>aababb</i>	Z_0	
q_2	<i>aababb</i>	Z_0	

Top-down = expand-match



preorder: leftmost

$S \xRightarrow{\ell} aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow$
 $aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow$
 $aababB \Rightarrow aababbS \Rightarrow aababb$

q_0	<i>aababb</i>	Z_0	
q_1	<i>aababb</i>	$S Z_0$	1 : $S \rightarrow aB$
q_1	<i>aababb</i>	$aB Z_0$	match a
q_1	<i>a ababb</i>	$B Z_0$	2 : $B \rightarrow aBB$
q_1	<i>a ababb</i>	$aBB Z_0$	match a
q_1	<i>aa babb</i>	$BB Z_0$	3 : $B \rightarrow bS$
q_1	<i>aa babb</i>	$bSB Z_0$	match b
q_1	<i>aab abb</i>	$SB Z_0$	4 : $S \rightarrow \Lambda$
q_1	<i>aab abb</i>	$B Z_0$	5 : $B \rightarrow aBB$
q_1	<i>aab abb</i>	$aBB Z_0$	match a
q_1	<i>aaba bb</i>	$BB Z_0$	6 : $B \rightarrow bS$
q_1	<i>aaba bb</i>	$bSB Z_0$	match b
q_1	<i>aabab b</i>	$SB Z_0$	7 : $S \rightarrow \Lambda$
q_1	<i>aabab b</i>	$B Z_0$	8 : $B \rightarrow bS$
q_1	<i>aabab b</i>	$bS Z_0$	match b
q_1	<i>aababb</i>	$S Z_0$	9 : $S \rightarrow \Lambda$
q_1	<i>aababb</i>	Z_0	
q_2	<i>aababb</i>	Z_0	

Theorem

If G is a context-free grammar, then the nondeterministic top-down PDA $NT(G)$ accepts the language $L(G)$.

The details of the proof of this result do not have to be known for the exam.

Intuition $L(G) \subseteq L(NT(G)) \dots$:

- wish to simulate derivation on stack
- match terminals when on top of stack
- topmost variable first, so leftmost derivation

[M] Th 5.18

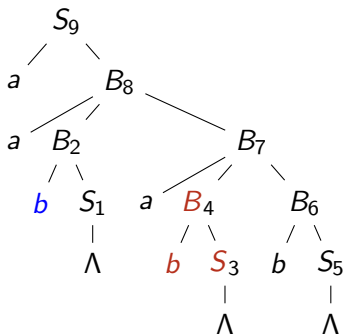
$$S \rightarrow abc$$

$$AEqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$$

$$S \rightarrow \Lambda \mid aB \mid bA$$

$$A \rightarrow aS \mid bAA$$

$$B \rightarrow bS \mid aBB$$



q_0 Z_0 $stack^r$ input shift a
 $aababbb$
 $\dots$

Bottom-up = shift-reduce

$$AEqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$$

$$S \rightarrow \Lambda \mid aB \mid bA$$

$$A \rightarrow aS \mid bAA$$

$$B \rightarrow bS \mid aBB$$

	stack ^r	input	
q_0	Z_0	$aababb$	shift a
q_0	$Z_0 a$	$a ababb$	shift a
q_0	$Z_0 aa$	$aa babb$	shift b
q_0	$Z_0 aab$	$aab abb$	1 : $S \rightarrow \Lambda$
q_0	$Z_0 aabS$	$aab abb$	2 : $B \rightarrow bS$
q_0	$Z_0 aaB$	$aab abb$	shift a
q_0	$Z_0 aaBa$	$aaba bb$	shift b
q_0	$Z_0 aaBab$	$aabab b$	3 : $S \rightarrow \Lambda$
q_0	$Z_0 aaBabS$	$aabab b$	4 : $B \rightarrow bS$
q_0	$Z_0 aaBaB$	$aabab b$	shift b
q_0	$Z_0 aaBaBb$	$aababb$	5 : $S \rightarrow \Lambda$
q_0	$Z_0 aaBaBbS$	$aababb$	6 : $B \rightarrow bS$
q_0	$Z_0 aaBaBB$	$aababb$	7 : $B \rightarrow aBB$
q_0	$Z_0 aaBB$	$aababb$	8 : $B \rightarrow aBB$
q_0	$Z_0 aB$	$aababb$	9 : $S \rightarrow aB$
q_0	$Z_0 S$	$aababb$	
q_1	Z_0	$aababb$	
q_2	Z_0	$aababb$	

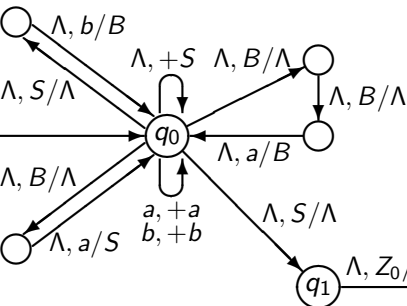
Bottom-up = shift-reduce

$$AEqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$$

$$S \rightarrow \Lambda \mid aB \mid bA$$

$$A \rightarrow aS \mid bAA$$

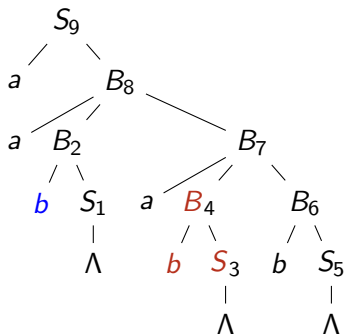
$$B \rightarrow bS \mid aBB$$



+ states/transitions for other productions

stack ^r	input	
q_0 Z_0	<i>aababb</i>	shift <i>a</i>
q_0 Z_0 <i>a</i>	<i>a ababb</i>	shift <i>a</i>
q_0 Z_0 <i>aa</i>	<i>aa babb</i>	shift <i>b</i>
q_0 Z_0 <i>aab</i>	<i>aab abb</i>	1 : $S \rightarrow \Lambda$
q_0 Z_0 <i>aabS</i>	<i>aab abb</i>	2 : $B \rightarrow bS$
q_0 Z_0 <i>aaB</i>	<i>aab abb</i>	shift <i>a</i>
q_0 Z_0 <i>aaBa</i>	<i>aaba bb</i>	shift <i>b</i>
q_0 Z_0 <i>aaBab</i>	<i>aabab b</i>	3 : $S \rightarrow \Lambda$
q_0 Z_0 <i>aaBabS</i>	<i>aabab b</i>	4 : $B \rightarrow bS$
q_0 Z_0 <i>aaBaB</i>	<i>aabab b</i>	shift <i>b</i>
q_0 Z_0 <i>aaBaBb</i>	<i>aababb</i>	5 : $S \rightarrow \Lambda$
q_0 Z_0 <i>aaBaBbS</i>	<i>aababb</i>	6 : $B \rightarrow bS$
q_0 Z_0 <i>aaBaBB</i>	<i>aababb</i>	7 : $B \rightarrow aBB$
q_0 Z_0 <i>aaBB</i>	<i>aababb</i>	8 : $B \rightarrow aBB$
q_0 Z_0 <i>aB</i>	<i>aababb</i>	9 : $S \rightarrow aB$
q_0 Z_0 <i>S</i>	<i>aababb</i>	
q_1 Z_0	<i>aababb</i>	
q_2 Z_0	<i>aababb</i>	

Bottom-up = shift-reduce

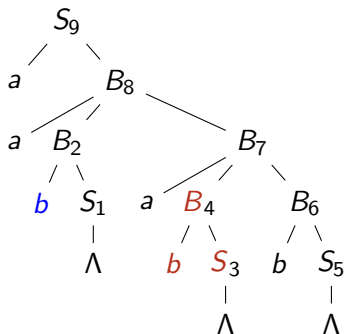


postorder: rightmost

$S \Rightarrow_9 aB \Rightarrow_8 aaBB \Rightarrow_7 aaBaBB \Rightarrow_6$
 $aaBaBbS \Rightarrow_5 aaBaBbS \Rightarrow_4 aaBaBbS \Rightarrow_3$
 $aaBabb \Rightarrow_2 aabSabb \Rightarrow_1 aababb$

	stack ^r	input	
q_0	Z_0	<i>aababb</i>	shift <i>a</i>
q_0	$Z_0 a$	<i>a ababb</i>	shift <i>a</i>
q_0	$Z_0 aa$	<i>aa babb</i>	shift <i>b</i>
q_0	$Z_0 aab$	<i>aab abb</i>	1 : $S \rightarrow \Lambda$
q_0	$Z_0 aabS$	<i>aab abb</i>	2 : $B \rightarrow bS$
q_0	$Z_0 aaB$	<i>aab abb</i>	shift <i>a</i>
q_0	$Z_0 aaBa$	<i>aaba bb</i>	shift <i>b</i>
q_0	$Z_0 aaBab$	<i>aabab b</i>	3 : $S \rightarrow \Lambda$
q_0	$Z_0 aaBabS$	<i>aabab b</i>	4 : $B \rightarrow bS$
q_0	$Z_0 aaBaB$	<i>aabab b</i>	shift <i>b</i>
q_0	$Z_0 aaBaBb$	<i>aababb</i>	5 : $S \rightarrow \Lambda$
q_0	$Z_0 aaBaBbS$	<i>aababb</i>	6 : $B \rightarrow bS$
q_0	$Z_0 aaBaBB$	<i>aababb</i>	7 : $B \rightarrow aBB$
q_0	$Z_0 aaBB$	<i>aababb</i>	8 : $B \rightarrow aBB$
q_0	$Z_0 aB$	<i>aababb</i>	9 : $S \rightarrow aB$
q_0	$Z_0 S$	<i>aababb</i>	
q_1	Z_0	<i>aababb</i>	
q_2	Z_0	<i>aababb</i>	

Bottom-up = shift-reduce



postorder: rightmost, in reverse

$S \Rightarrow_9 aB \Rightarrow_8 aaBB \Rightarrow_7 aaBaBB \Rightarrow_6$
 $aaBaBbS \Rightarrow_5 aaBaBb \Rightarrow_4 aaBabSb$
 $\Rightarrow_3 aaBabb \Rightarrow_2 aabSabb \Rightarrow_1 aababb$

	stack ^r	input	
q_0	Z_0	<i>aababb</i>	shift <i>a</i>
q_0	$Z_0 a$	<i>a ababb</i>	shift <i>a</i>
q_0	$Z_0 aa$	<i>aa babb</i>	shift <i>b</i>
q_0	$Z_0 aab$	<i>aab abb</i>	1 : $S \rightarrow \Lambda$
q_0	$Z_0 aabS$	<i>aab abb</i>	2 : $B \rightarrow bS$
q_0	$Z_0 aaB$	<i>aab abb</i>	shift <i>a</i>
q_0	$Z_0 aaBa$	<i>aaba bb</i>	shift <i>b</i>
q_0	$Z_0 aaBab$	<i>aabab b</i>	3 : $S \rightarrow \Lambda$
q_0	$Z_0 aaBabS$	<i>aabab b</i>	4 : $B \rightarrow bS$
q_0	$Z_0 aaBaB$	<i>aabab b</i>	shift <i>b</i>
q_0	$Z_0 aaBaBb$	<i>aababb</i>	5 : $S \rightarrow \Lambda$
q_0	$Z_0 aaBaBbS$	<i>aababb</i>	6 : $B \rightarrow bS$
q_0	$Z_0 aaBaBB$	<i>aababb</i>	7 : $B \rightarrow aBB$
q_0	$Z_0 aaBB$	<i>aababb</i>	8 : $B \rightarrow aBB$
q_0	$Z_0 aB$	<i>aababb</i>	9 : $S \rightarrow aB$
q_0	$Z_0 S$	<i>aababb</i>	
q_1	Z_0	<i>aababb</i>	
q_2	Z_0	<i>aababb</i>	

ABOVE

To write down the construction of the shift-reduce PDA for a given CFG, we have two technical problems.

Consider a production $A \rightarrow \alpha$

First the stack (in standard notation) now contains the string α in reverse. Second, we pop α , that is, several symbols, rather than exactly one. This can be simulated by popping the symbols one-by-one, using $|\alpha|$ separate transitions.

shift $\delta(q_0, \sigma, X) = \{(q_0, \sigma X)\}$ for $\sigma \in \Sigma, X \in \Gamma$

reduce ' $\delta^*(q_0, \Lambda, \alpha^r) \ni (q_0, A)$ ' for $A \rightarrow \alpha$ in P

Special case: $\alpha = \Lambda$

Definition

The Nondeterministic Bottom-Up PDA $NB(G)$

Let $G = (V, \Sigma, S, P)$ be a context-free grammar.

The nondeterministic bottom-up PDA corresponding to G is

$NB(G) = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$, defined as follows:

Q contains the initial state q_0 , the state q_1 , and the (only) accepting state q_2 , together with other states to be described shortly.

$\Gamma = \dots$

[M] D 5.22

Definition

The Nondeterministic Bottom-Up PDA $NB(G)$

Let $G = (V, \Sigma, S, P)$ be a context-free grammar.

The nondeterministic bottom-up PDA corresponding to G is

$NB(G) = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$, defined as follows:

Q contains the initial state q_0 , the state q_1 , and the (only) accepting state q_2 , together with other states to be described shortly.

$$\Gamma = V \cup \Sigma \cup \{Z_0\}$$

[M] D 5.22

Definition

The Nondeterministic Bottom-Up PDA $NB(G)$ (continued)

For every $\sigma \in \Sigma$ and every $X \in \Gamma$, $\delta(q_0, \sigma, X) = \{(q_0, \sigma X)\}$. This is a *shift* move.

For every production $B \rightarrow \alpha$ in G , and every nonnull string $\beta \in \Gamma^*$,
 $(q_0, \Lambda, \alpha^r \beta) \vdash^* (q_0, \Lambda, B\beta)$,

where this *reduction* is a sequence of one or more moves in which, if there is more than one, the intermediate configurations involve other states that are specific to this sequence and appear in no other moves of $NB(G)$.

One of the elements of $\delta(q_0, \Lambda, S)$ is (q_1, Λ) ,
and $\delta(q_1, \Lambda, Z_0) = \{(q_2, Z_0)\}$.

[M] D 5.22

Theorem

If G is a context-free grammar, then the nondeterministic bottom-up PDA $NB(G)$ accepts the language $L(G)$.

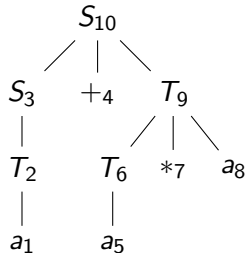
The details of the proof of this result do not have to be known for the exam.

[M] Th 5.23

Example: algebraic expressions

shift-reduce

post-order reduction $\equiv$ rightmost derivation, bottom-up

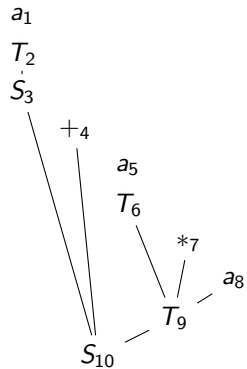


stack [reverse]

Z_0
 $Z_0 a_1$
 $Z_0 T_2$
 $Z_0 S_3$
 $Z_0 S_3 +_4$
 $Z_0 S_3 +_4 a_5$
 $Z_0 S_3 +_4 T_6$
 $Z_0 S_3 +_4 T_6 *_7$
 $Z_0 S_3 +_4 T_6 *_7 a_8$
 $Z_0 S_3 +_4 T_9$
 $Z_0 S_{10}$
 —

input

$a + a * a$
 $+ a * a$
 $+ a * a$
 $+ a * a$
 $a * a$
 $* a$
 $* a$
 a

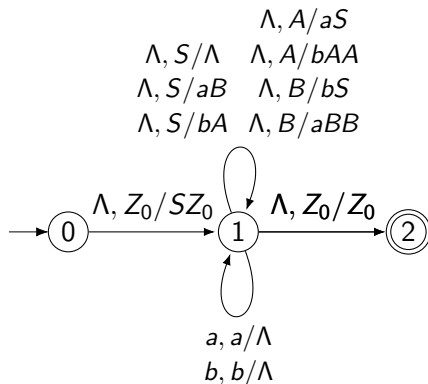


$$AEqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$$

$$S \rightarrow \Lambda \mid aB \mid bA$$

$$A \rightarrow aS \mid bAA$$

$$B \rightarrow bS \mid aBB$$



5.5. Parsing: make PDA (more) deterministic by looking ahead one symbol in input.

See Compiler Construction