From lecture 10:

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

Definition (Nondeterministic Top-Down PDA)

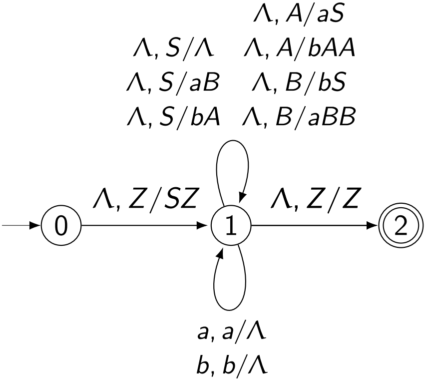
$NT(G) = (Q, \Sigma, \Gamma, q_0, Z, A, \delta)$, as follows:

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

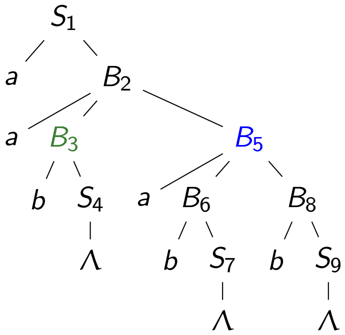
[M] Def 5.17

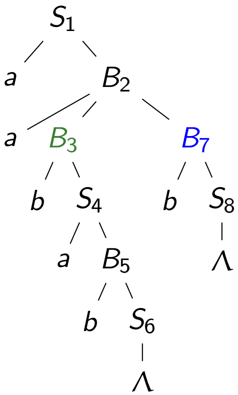
$$AeqB = \{\ x \in \{a, b\}^* \mid n_a(x) = n_b(x)\ \}$$

$$S \to \Lambda \mid aB \mid bA$$
$$A \to aS \mid bAA$$
$$B \to bS \mid aBB$$



From lecture 8:





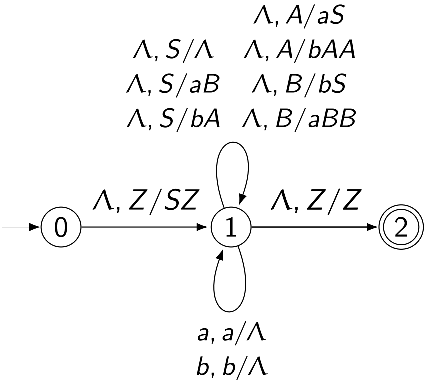
$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow$
$aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow$
$aababB \Rightarrow aababbS \Rightarrow aababb$

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

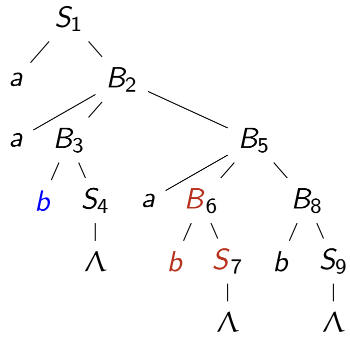
Top-down = expand-match

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

$$S \to \Lambda \mid aB \mid bA$$
$$A \to aS \mid bAA$$
$$B \to bS \mid aBB$$



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



preorder: leftmost

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

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

Theorem

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

Proof: $L(G) \subseteq L(NT(G))\ldots$

The details of the proof in the other direction do not have to be known for the exam.

[M] Th 5.18

One leftmost derivation step:

$$y_i A_i \alpha_i \Rightarrow y_i \beta_i \alpha_i = y_i x_{i+1} A_{i+1} \alpha_{i+1}$$

With $y_i = x_0 x_1 \ldots x_i$:

$$x_0 x_1 \ldots x_i A_i \alpha_i \Rightarrow x_0 x_1 \ldots x_i \beta_i \alpha_i = x_0 x_1 \ldots x_i x_{i+1} A_{i+1} \alpha_{i+1}$$

Complete leftmost derivation:

$$\begin{array}{rcl} S & = & x_0 A_0 \alpha_0 \\ & \Rightarrow & x_0 x_1 A_1 \alpha_1 \\ & \Rightarrow & x_0 x_1 x_2 A_2 \alpha_2 \\ & \Rightarrow & \ldots \\ & \Rightarrow & x_0 x_1 x_2 \ldots x_m A_m \alpha_m \\ & \Rightarrow & x_0 x_1 x_2 \ldots x_m x_{m+1} = x \end{array}$$

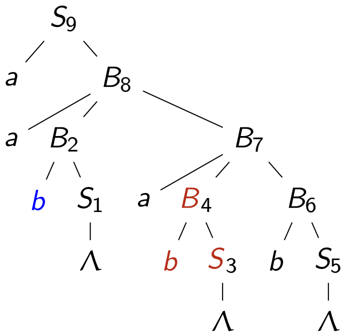
$AeqB = \{ x \in \{a, b\}^* \mid n_a(x) = n_b(x) \}$

$S \to \Lambda \mid aB \mid bA$

$A \to aS \mid bAA$

$B \to bS \mid aBB$

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



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

postorder: rightmost, in reverse

$S \overset{r}{\Rightarrow}_9 aB \Rightarrow_8 aaBB \Rightarrow_7 aaBaBB \Rightarrow_6$
$aaBaBbS \Rightarrow_5 aaBaBb \Rightarrow_4 aaBabSb$
$\Rightarrow_3 aaBabb \Rightarrow_2 aabSabb \Rightarrow_1 aababb$

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

Consider a production $A \to \alpha$

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 separate instructions.

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 $A \to \alpha$ in P

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.

$\Gamma = \ldots$

[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 $B \to \alpha$ 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

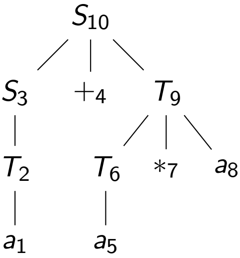
If G is a context-free grammar, then the nondeterministic bottom-up PDA $\mathrm{NB}(G)$ *accepts the language* $L(G)$.

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

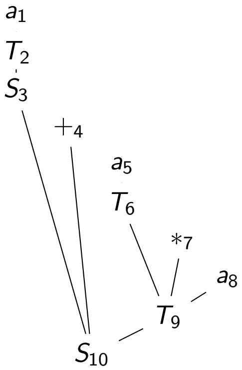
[M] Th 5.23

shift-reduce

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



stack [reverse]	input
Z	$a + a * a$
$Z\ a_1$	$+a * a$
$Z\ T_2$	$+a * a$
$Z\ S_3$	$+a * a$
$Z\ S_3\ +_4$	$a * a$
$Z\ S_3\ +_4\ a_5$	$*a$
$Z\ S_3\ +_4\ T_6$	$*a$
$Z\ S_3\ +_4\ T_6\ *_7$	a
$Z\ S_3\ +_4\ T_6\ *_7\ a_8$	
$Z\ S_3\ +_4\ T_9$	
$Z\ S_{10}$	
$-$	



Due to Chomsky, Evey, and Schützenberger (1962/3).

Theorem

Context-free grammars and Pushdown automata are equivalent.

$\hookrightarrow$(1) PDA acceptance by empty stack

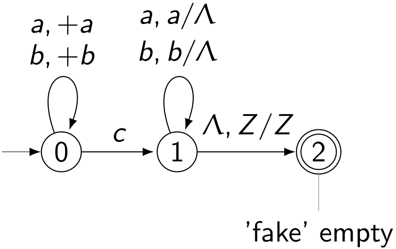
$\hookrightarrow$(2) triplet construction, CFG nonterminals $[p, A, q]$ for PDA computations

References

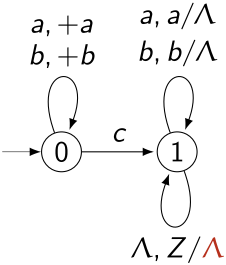
N. Chomsky. Context-free grammars and push-down storage. Quarterly Progress Report No. 65, Research Laboratory of Electronics, M.I.T., Princeton, New Jersey (1962)

R.J. Evey. The theory and application of pushdown store machines. In Mathematical Linguistics and Automatic Translation, NSF-IO, pages 217–255. Harvard University, May 1963,
and

M. P. Schützenberger. On context-free languages and pushdown automata. Inform. and Control, 6:217–255, 1963. doi:10.1016/S0019-9958(63)90306-1



check state



check stack

ABOVE

On many cases the PDA moves to the accepting state after checking that the stack is empty, when the topmost symbol is a special Z that always has been at the bottom of the stack.

It is more natural to accept directly by looking at the stack rather than by looking at the state. This leads to the notion of the *empty stack* language of a PDA.

$M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$

Definition

Language accepted by M by *empty stack*

$L_e(M) = \{ x \in \Sigma^* \mid (q_0, x, Z_0) \vdash^* (q, \Lambda, \Lambda)$ for some state $q \in Q \}$

[M] D 5.27

Theorem

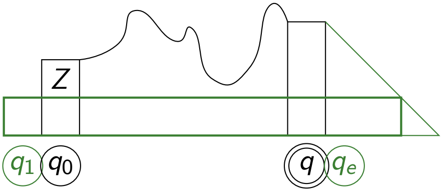
If M is a PDA then there is a PDA M_1 such that $L_e(M_1) = L(M)$.

Sketch of proof. . .

[M] Th 5.28

Simulate $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$

Simulate $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$
– empty stack 'at' final state
– prohibit early empty stack

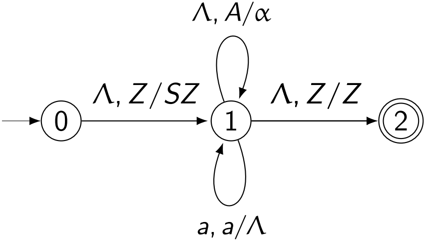


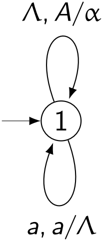
Construction PDA $M_1 = (Q_1, \Sigma, \Gamma_1, q_1, Z_1, A_1, \delta_1)$ such that $L_e(M_1) = L(M)$

– $Q_1 = Q \cup \{q_1, q_e\}$
– $\Gamma_1 = \Gamma \cup \{Z_1\}$
– new instructions:

$\delta_1(q_1, \Lambda, Z_1) = \{(q_0, Z_0 Z_1)\}$
$\delta_1(q, \Lambda, X) \ni (q_e, X)$ for $q \in A$, and $X \in \Gamma_1$
$\delta_1(q_e, \Lambda, X) = \{(q_e, \Lambda)\}$ for $X \in \Gamma_1$

$A \to \alpha \in P$, $a \in \Sigma$





Theorem

For every CFL L *there exists a single state* PDA M *such that* $L_e(M) = L$.

ABOVE

Now that we have empty stack acceptance we can reconsider the expand-match technique. In fact we do not need two extra states to introduce a bottom of stack symbol, and can make a single state PDA.

BELOW

The expand-match method can be used for any CFG. If we slightly restrict the grammars, we can combine each match with the expand step just before, that introduced the terminal. This gives a very direct translation between grammar and its leftmost derivation, and a single state PDA and its computation.

On this normal form each production is of the form $A \to a\alpha$, where $a \in \Sigma \cup \{\Lambda\}$ can be the only terminal at the right. That means that any terminal pushed on the stack will be on top, and immediately will be matched.

cfg G $\iff$ 1-pda M

$A \to \alpha$ — $\delta(-, \Lambda, A) \ni (-, \alpha)$ expand

$(-, a, a) = \{(-, \Lambda)\}$ match

normal form $\alpha \in (\Sigma \cup \{\Lambda\}) \cdot V^*$

$A \to a\alpha$ — $\delta(-, a, A) \ni (-, \alpha)$ combined

SimplePal: $S \to aSA \mid bSB \mid c$ $A \to a$ $B \to b$

leftmost derivation $\iff$ computation

	S		$(-, abbcbba, S)$
$\Rightarrow$	$a\,SA$	$\vdash$	$(-, bbcbba, SA)$
$\Rightarrow$	$ab\,SBA$	$\vdash$	$(-, bcbba, SBA)$
$\Rightarrow$	$abb\,SBBA$	$\vdash$	$(-, cbba, SBBA)$
$\Rightarrow$	$abbc\,BBA$	$\vdash$	$(-, bba, BBA)$
$\Rightarrow$	$abbcb\,BA$	$\vdash$	$(-, ba, BA)$
$\Rightarrow$	$abbcbb\,A$	$\vdash$	$(-, a, A)$
$\Rightarrow$	$abbcbba$	$\vdash$	$(-, \Lambda, \Lambda)$

Theorem

If $L = L_e(M)$ is the empty stack language of PDA *M, then there exists a* CFG *G such that $L = L(G)$.*

[M] Th 5.29

$M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$

Theorem

If $L = L_e(M)$ is the empty stack language of PDA *M, then there exists a* CFG *G such that $L = L(G)$.*

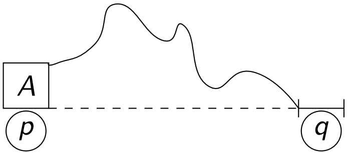
[M] Th 5.29

$M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$

triplet construction

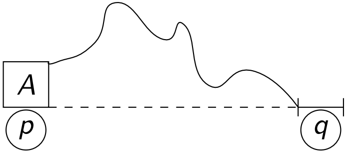
nonterminals $[p, A, q]$ $\quad p, q \in Q,\ A \in \Gamma$

$[p, A, q] \Rightarrow_G^* w$ iff $(p, w, A) \vdash_M^* (q, \Lambda, \Lambda)$

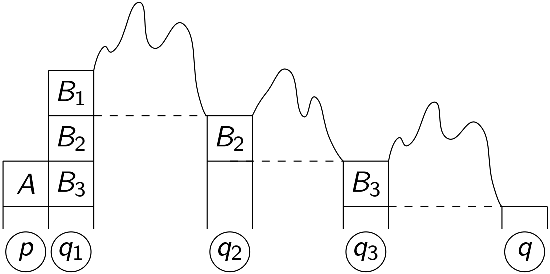


– nonterminals $[p, A, q]$ $\quad p, q \in Q,\ A \in \Gamma$

$[p, A, q] \Rightarrow_G^* w$ iff $(p, w, A) \vdash_M^* (q, \Lambda, \Lambda)$



– productions $S \to [q_0, Z_0, q]$ for all $q \in Q$



$[p, A, q] \to a\, [q_1, B_1, q_2][q_2, B_2, q_3] \cdots [q_n, B_n, q]$

for $(q_1, B_1 \cdots B_n) \in \delta(p, a, A)$, and $q, q_2, \ldots, q_n \in Q$

$[p, A, q] \to a$ for $(q, \Lambda) \in \delta(p, a, A)$