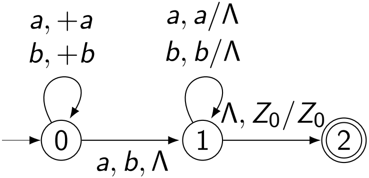
Deadline homework 3: Wednesday, 27 November, 23.59



$Q = \{0, 1, 2\}$
$\Sigma = \{a, b, c\}$
$\Gamma = \{a, b, Z_0\}$
$q_0 = 0$
$Z_0 = Z_0$
$A = \{2\}$

Theorem

The language Pal *cannot be accepted by a deterministic pushdown automaton.*

Proof. . .

[M] Thm 5.16

From lecture 3:

Definition

Let L be language over Σ, and let $x, y \in \Sigma^*$.
Then x, y are *distinguishable* wrt L (*L-distinguishable*),
if there exists $z \in \Sigma^*$ with

$$xz \in L \text{ and } yz \notin L \quad \text{or} \quad xz \notin L \text{ and } yz \in L$$

Such z *distinguishes* x and y wrt L.

[M] D 2.20

From lecture 3:

$Pal = \{x \in \{a, b\}^* \mid x = x^r\}$

For Every Pair x, y of Distinct Strings in $\{a, b\}^*$, x and y Are Distinghuishable with Respect to Pal.

[M] E. 2.27

Theorem

The language Pal *cannot be accepted by a deterministic pushdown automaton.*

Proof.

Assume M is DPDA for Pal.

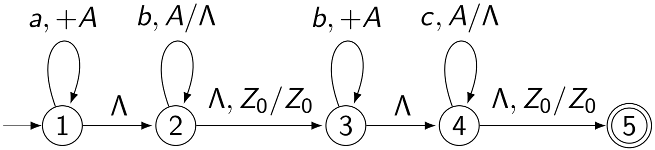
No assumption on form transitions M.

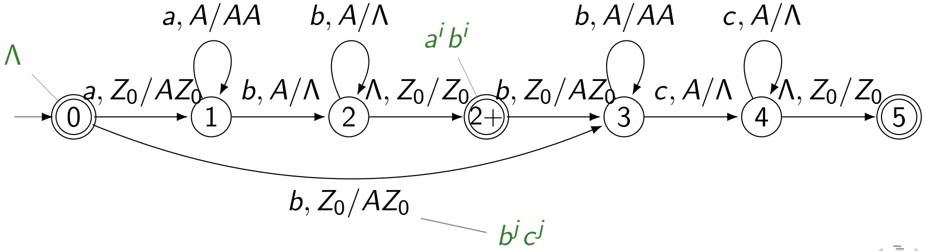
M reads every string $x \in \{a, b\}^*$ completely, with one path.

There exist different strings $r, s \in \{a, b\}^*$, such that for every $z \in \{a, b\}^*$, M treats rz and sz the same way.

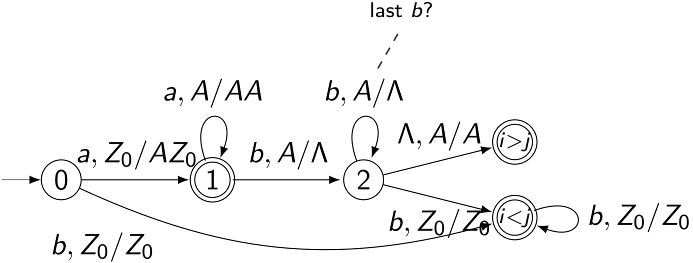
For a string $x \in \{a, b\}^*$, let y_x be a string such that height of stack after xy_x is minimal.
Let α_x be stack after xy_x.
(state, top stack symbol) determines how suffix z is treated.
Infinitely many strings xy_x. Why?
Finitely many pairs (q, X)
Different $r = uy_u$ and $s = vy_v$ arrive at same pair (q, A).
For any suffix z, rz and sz are treated the same:
$rz \in Pal \iff sz \in Pal$.
Contradiction.

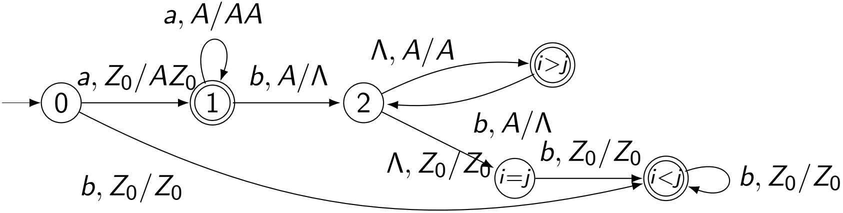
$S \to AB$
$A \to aAb \mid \Lambda$
$B \to bBc \mid \Lambda$

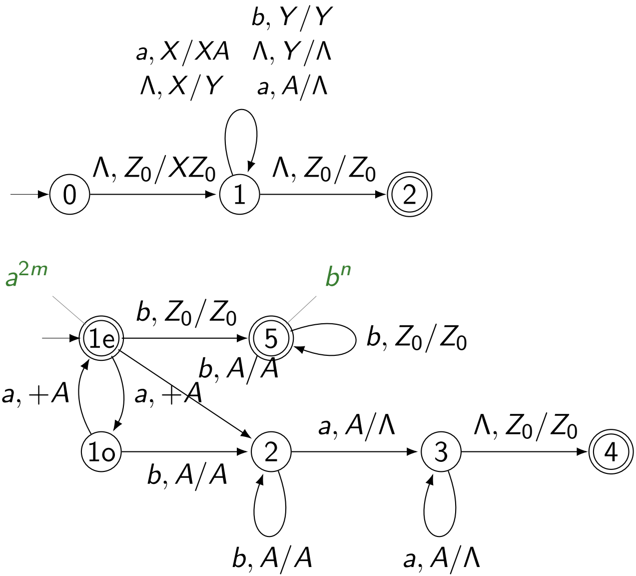




$\{ \ a^i b^j \mid i \neq j \ \}$






$b, Y/Y$
$a, X/XA$
$\Lambda, Y/\Lambda$
$\Lambda, X/Y$
$a, A/\Lambda$
$\Lambda, Z_0/XZ_0$
$\Lambda, Z_0/Z_0$
0
1
2
a^{2m}
b^n
$b, Z_0/Z_0$
$b, Z_0/Z_0$
1e
5
$b, A/A$
$a, +A$
$a, +A$
1o
2
3
4
$a, A/\Lambda$
$\Lambda, Z_0/Z_0$
$b, A/A$
$b, A/A$
$a, A/\Lambda$

ABOVE

The first PDA is not deterministic. Actually it is working like a grammar: in state 1 the following productions are simulated:

$X \to aXA \mid Y$

$Y \to bY \mid \Lambda$

$A \to a$

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

$pre(L) = \{\ x\#y \mid x \in L \text{ and } xy \in L\ \}$

$L = Pal = \{\Lambda, a, b, aa, bb, aaa, aba, bab, bbb, aaaa, abba, \ldots\}$
$pre(L) = \ldots$

$L = \{a^i b^j \mid i < j\} = \{b, bb, abb, bbb, abbb, bbbb, aabbb, abbbb, \ldots\}$
$pre(L) = \ldots$

$pre(L) = \{ x\#y \mid x \in L \text{ and } xy \in L \}$

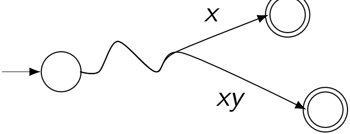
CFL not closed under *pre* ⊠
DCFL *is* closed under *pre* ⊠

[M] Exercise 5.20 & 6.22

CFL not closed under complement
DCFL is closed under complement ⊠
(the obvious proof does not work)

CFL is closed under regular operations $\cup, \cdot, *$
DCFL is not closed under either of these ⊠

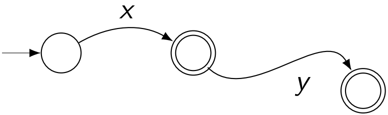
language L $\quad x \in L$, $xy \in L$



$K = \{\ a^n b^n \mid n \geq 1\ \} \cup \{\ a^n b^m c^n \mid m, n \geq 1\ \}$

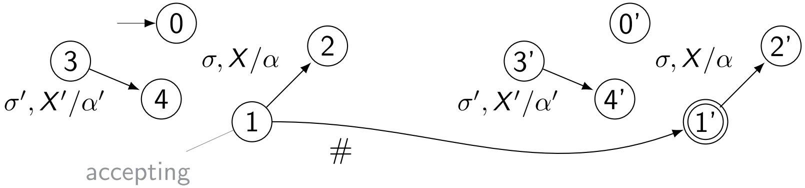
$\underline{a^n}\ \underline{b^n} \quad \underline{a^n}\ b^m\ \underline{c^n}$ different behaviour on b's

$pre(K) = \ldots$



DCFL is closed under *pre*

$pre(L) = \{ x\#y \mid x, xy \in L \}$

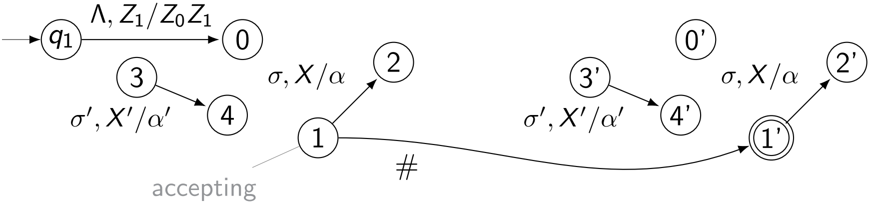


$M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ with $L = L(M)$
construct $M_1 = (Q_1, \Sigma \cup \{\#\}, \Gamma, q_1, Z_1, A_1, \delta_1)$ with $L(M_1) = pre(L)$

- $Q_1 = Q \cup Q'$ where $Q' = \{ q' \mid q \in Q \}$ — primed copy
- $q_1 = q_0$, $Z_1 = Z_0$
- $A_1 = A' = \{ q' \mid q \in A \}$ — accepting states in copy
- $\delta_1(p', \sigma, X) = \{(q', \alpha) \mid (q, \alpha) \in \delta(p, \sigma, X)\}$ — two copies
  for all $p \in A, X \in \Gamma$: $\delta_1(p, \#, X) = \{(p', X)\}$ — move to primed copy

DCFL is closed under *pre*

$pre(L) = \{\ x\#y \mid x, xy \in L\ \}$



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

construct $M_1 = (Q_1, \Sigma \cup \{\#\}, \Gamma \cup \{Z_1\}, q_1, Z_1, A_1, \delta_1)$ with $L(M_1) = pre(L)$

– $Q_1 = Q \cup Q' \cup \{q_1\}$ where $Q' = \{\ q' \mid q \in Q\ \}$ primed copy

– $A_1 = A' = \{\ q' \mid q \in A\ \}$ accepting states in copy

– $\delta_1(p', \sigma, X) = \{(q', \alpha) \mid (q, \alpha) \in \delta(p, \sigma, X)\}$ two copies

$\delta_1(q_1, \Lambda, Z_1) = \{(q_0, Z_0Z_1)\}$ Z_1 under Z_0

for all $p \in A, X \in \Gamma_1$: $\delta_1(p, \#, X) = \{(p', X)\}$ move to primed copy

☒ABOVE

For $K = \{ a^n b^n \mid n \geq 1 \} \cup \{ a^n b^m c^n \mid m, n \geq 1 \}$
we have $pre(K) = K\# \cup \{a^n b^n \# b^k c^n \mid n \geq 1, k \geq 0\}$.
This language is not context-free, but K is, and thus the context-free languages are not closed under pre.

Again, this construction works because (for deterministic automata) the computation on uv *must* extend the computation on u.

Note the resulting PDA might not be deterministic at accepting states in original Q (like node 1 in the diagram), if that node has an outgoing Λ-transition.
There is however a method that avoids Λ-transitions at accepting states. Whenever $(q, \alpha) \in \delta(p, \Lambda, A)$ for an accepting state p, just 'predict' the next letter σ read, add a new state (q, σ), add $((q, \sigma), \alpha)$ to $\delta(p, \sigma, A)$ (which was empty beforehand, why?). Do this for every σ, and remove the Λ-transition. Then keep simulating Λ-transitions, until σ is read.

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

$S \to X \mid Y$ (choice!)

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

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

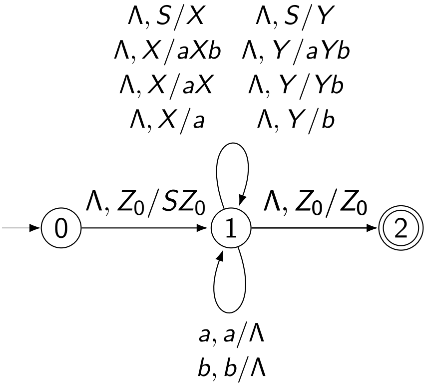
$S \Rightarrow X \Rightarrow aXb \Rightarrow aaXb \Rightarrow aaaXbb \Rightarrow aaaabb$

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

$S \to X \mid Y$ (choice!)

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

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



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* $\delta(q_1, \Lambda, A) = \{(q_1, \alpha) \mid A \to \alpha \text{ in } P\}$ for $A \in V$
- *match* $\delta(q_1, \sigma, \sigma) = \{(q_1, \Lambda)\}$ for $\sigma \in \Sigma$
- finish $\delta(q_1, \Lambda, Z_0) = \{(q_2, Z_0)\}$ check empty stack

[M] Def 5.17

From lecture 8:

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

$aaabbb, ababab, aababb, \ldots$

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

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

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

$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow \ldots$ (different options)

(1) $aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow aababB \Rightarrow aababbS \Rightarrow aababb$

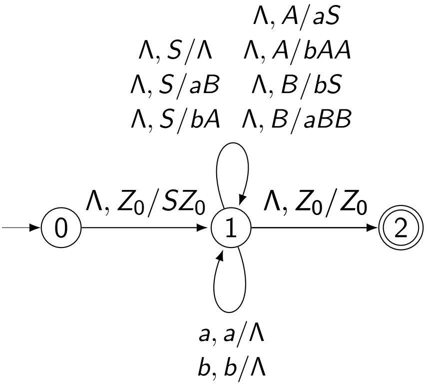
(2) $aaba\underline{B}B \Rightarrow aabab\underline{S}B \Rightarrow aabab\underline{B} \Rightarrow aababb\underline{S} \Rightarrow aababb$

(2') $aabaB\underline{B} \Rightarrow aaba\underline{B}bS \Rightarrow aababSb\underline{S} \Rightarrow aabab\underline{S}b \Rightarrow aababb$

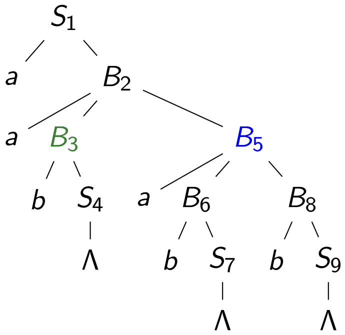
[M] E 4.8

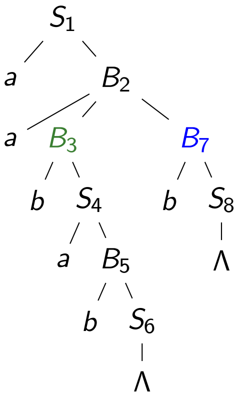
$$AeqB = \{\ x \in \{a, b\}^* \mid n_a(x) = n_b(x)\ \}$$

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



From lecture 8:



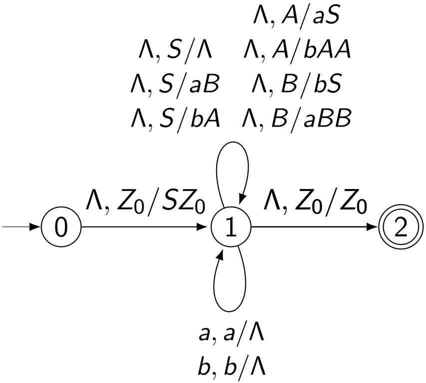


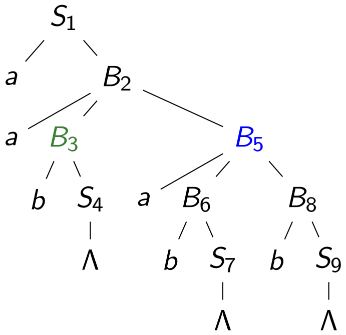
$S \Rightarrow aB \Rightarrow aaBB \Rightarrow aabSB \Rightarrow aabB \Rightarrow aabaBB \Rightarrow aababSB \Rightarrow aababB \Rightarrow aababbS \Rightarrow aababb$

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

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

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





$q_0 \quad aababb \quad Z_0$

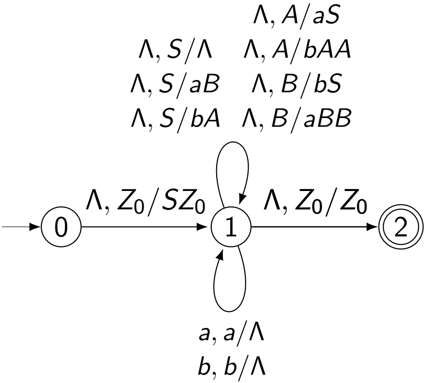
$\dots$

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

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

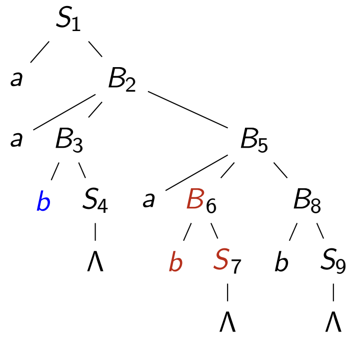
$A \to aS \mid bAA$

$B \to bS \mid aBB$



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

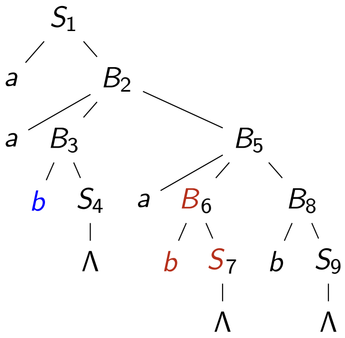
Top-down = expand-match



preorder: leftmost

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

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



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_0	
q_1	$aababb$	$S\ Z_0$	$1 : S \to aB$
q_1	$aababb$	$aB\ Z_0$	match a
q_1	$a\ ababb$	$B\ Z_0$	$2 : B \to aBB$
q_1	$a\ ababb$	$aBB\ Z_0$	match a
q_1	$aa\ babb$	$BB\ Z_0$	$3 : B \to bS$
q_1	$aa\ babb$	$bSB\ Z_0$	match b
q_1	$aab\ abb$	$SB\ Z_0$	$4 : S \to \Lambda$
q_1	$aab\ abb$	$B\ Z_0$	$5 : B \to aBB$
q_1	$aab\ abb$	$aBB\ Z_0$	match a
q_1	$aaba\ bb$	$BB\ Z_0$	$6 : B \to bS$
q_1	$aaba\ bb$	$bSB\ Z_0$	match b
q_1	$aabab\ b$	$SB\ Z_0$	$7 : S \to \Lambda$
q_1	$aabab\ b$	$B\ Z_0$	$8 : B \to bS$
q_1	$aabab\ b$	$bS\ Z_0$	match b
q_1	$aababb$	$S\ Z_0$	$9 : S \to \Lambda$
q_1	$aababb$	Z_0	
q_2	$aababb$	Z_0	

Theorem

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

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

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

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

[M] Th 5.18

One leftmost derivation step:

$$y_i A_i \alpha_i \Rightarrow y_i \beta_i \alpha_i = y_i x_{i+1} A_{i+1} \alpha_{i+1} \qquad \text{with } y_i, x_{i+1} \in \Sigma^*$$

With $y_i = x_0 x_1 \dots x_i$:

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

Complete leftmost derivation:

$$\begin{array}{rcl} S & = & x_0 A_0 \alpha_0 \\ & \Rightarrow & x_0 x_1 A_1 \alpha_1 \\ & \Rightarrow & x_0 x_1 x_2 A_2 \alpha_2 \\ & \Rightarrow & \dots \\ & \Rightarrow & x_0 x_1 x_2 \dots x_m A_m \alpha_m \\ & \Rightarrow & x_0 x_1 x_2 \dots x_m \beta_m \alpha_m = x \end{array}$$

Use induction on i to prove that for $i = 0, 1, \ldots, m$, in $NT(G)$,

$$(q_0, x, Z_0) = (q_0, x_0x_1 \ldots x_m\beta_m\alpha_m, Z_0) \vdash^* (q_1, x_{i+1} \ldots x_m\beta_m\alpha_m, A_i\alpha_i Z_0)$$

That is, $NT(G)$ can perform steps that read as input $x_0x_1 \ldots x_i$ and leave $A_i\alpha_i Z_0$ on stack.

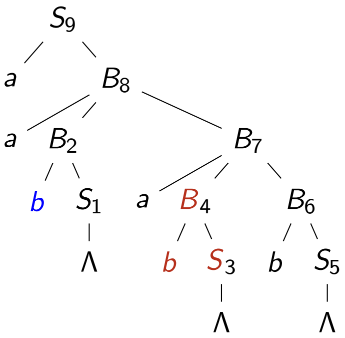
Then prove that

$$(q_1, \beta_m\alpha_m, A_m\alpha_m Z_0) \vdash^* (q_2, \Lambda, Z_0)$$

$$S \rightarrow abc$$

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

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



	stack^r	input	
q_0	Z_0	$aababb$	shift a
...			

Bottom-up = shift-reduce

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

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

$A \to aS \mid bAA$

$B \to bS \mid aBB$

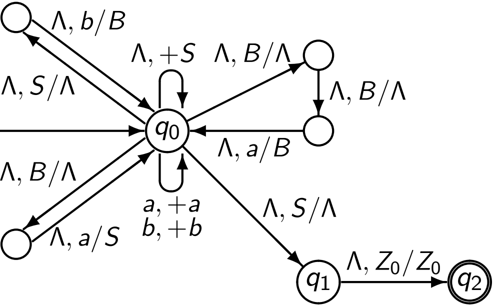
	stack^r	input	
q_0	Z_0	$aababb$	shift a
q_0	$Z_0\ a$	$a\ ababb$	shift a
q_0	$Z_0\ aa$	$aa\ babb$	shift b
q_0	$Z_0\ aab$	$aab\ abb$	$1: S \to \Lambda$
q_0	$Z_0\ aabS$	$aab\ abb$	$2: B \to bS$
q_0	$Z_0\ aaB$	$aab\ abb$	shift a
q_0	$Z_0\ aaBa$	$aaba\ bb$	shift b
q_0	$Z_0\ aaBab$	$aabab\ b$	$3: S \to \Lambda$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

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

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

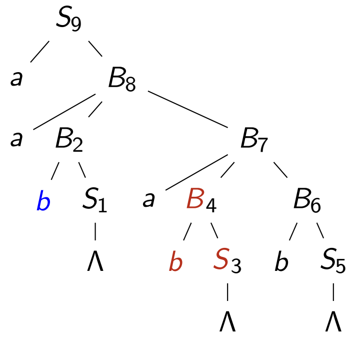
$A \to aS \mid bAA$

$B \to bS \mid aBB$



+ states/transitions for other productions

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

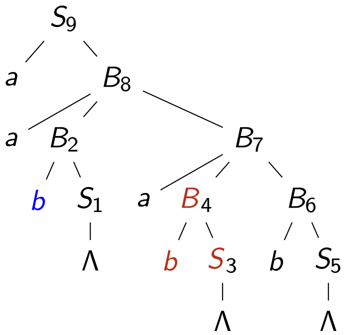


postorder: rightmost

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

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

Bottom-up = shift-reduce



postorder: rightmost, in reverse

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

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

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 $|\alpha|$ 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 $A \to \alpha$ 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.

$\Gamma = \ldots$

[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* $L(G)$.

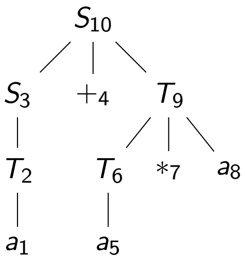
The details of the proof of this result do not have to be known for the exam.

[M] Th 5.23

Example: algebraic expressions

shift-reduce

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



stack [reverse]	input
Z_0	$a + a * a$
$Z_0\ a_1$	$+a * a$
$Z_0\ T_2$	$+a * a$
$Z_0\ S_3$	$+a * a$
$Z_0\ S_3\ +_4$	$a * a$
$Z_0\ S_3\ +_4\ a_5$	$*a$
$Z_0\ S_3\ +_4\ T_6$	$*a$
$Z_0\ S_3\ +_4\ T_6\ *_7$	a
$Z_0\ S_3\ +_4\ T_6\ *_7\ a_8$	
$Z_0\ S_3\ +_4\ T_9$	
$Z_0\ S_{10}$	
$-$