Homework 4! (probably Tuesday)

From lecture 11:

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

Definition

Language accepted by M by *empty stack*

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

[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

From lecture 11:

Exercise 5.21.

Prove the converse of Theorem 5.28:

If there is a PDA $M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$ accepting L by empty stack (that is, $x \in L$ if and only if $(q_0, x, Z_0) \vdash_M^* (q, \Lambda, \Lambda)$ for some state q), then there is a PDA M_1 accepting L by final state (i.e., the ordinary way).

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

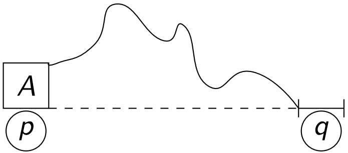
If $L = L_e(M)$ is the empty stack language of PDA M, then there exists a CFG G such that $L = L(G)$.

[M] Th 5.29

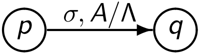
$M = (Q, \Sigma, \Gamma, q_0, Z_0, A, \delta)$

triplet construction

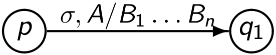
nonterminals $[p, A, q]$ $p, q \in Q,\ A \in \Gamma$
$[p, A, q] \Rightarrow_G^* w$ iff $(p, w, A) \vdash_M^* (q, \Lambda, \Lambda)$

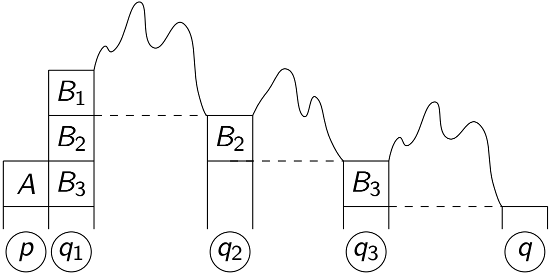


– productions



$[p, A, q] \rightarrow \sigma$ for $(q, \Lambda) \in \delta(p, \sigma, A)$



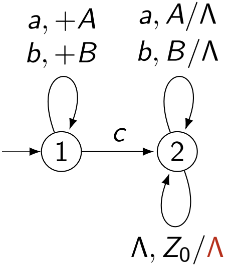


$[p, A, q] \rightarrow \sigma\ [q_1, B_1, q_2][q_2, B_2, q_3] \cdots [q_n, B_n, q]$
for $(q_1, B_1 \cdots B_n) \in \delta(p, \sigma, A)$, and $q, q_2, \ldots, q_n \in Q$

$S \rightarrow [q_0, Z_0, q]$ for all $q \in Q$

N.B.: σ may also be Λ

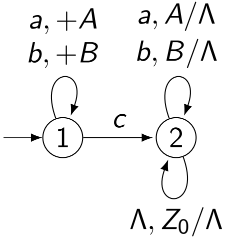
Construction from PDA to CFG, and the intuition behind it, must be known for the exam.
The details of the proof that $L(G) = L_e(M)$ do not have to be known for the exam.



check stack

$L_e(M) = SimplePal = \{\ wcw^r \mid w \in \{a,b\}^*\ \}$

12 transitions $\Rightarrow$ 33 (+2) productions (!)



$X \in \{A, B, Z_0\}$

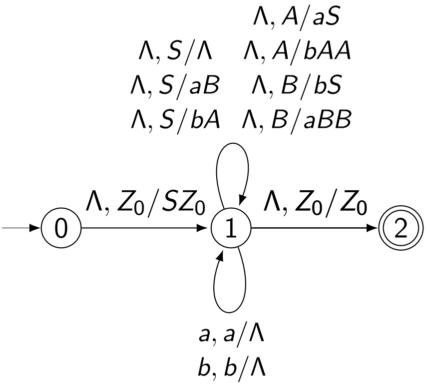
	$S \to [1,Z_0,1] \mid [1,Z_0,2]$
$\delta(1,a,X) = \{(1,AX)\}$	$[1,X,1] \to a\,[1,A,1][1,X,1]$
	$[1,X,1] \to a\,[1,A,2][2,X,1]$
	$[1,X,2] \to a\,[1,A,1][1,X,2]$
	$[1,X,2] \to a\,[1,A,2][2,X,2]$
$\delta(1,b,X) = \{(1,BX)\}$	$\dots$
$\delta(1,c,X) = \{(2,X)\}$	$[1,X,1] \to c\,[2,X,1]$
	$[1,X,2] \to c\,[2,X,2]$
$\delta(2,a,A) = \{(2,\Lambda)\}$	$[2,A,2] \to a$
$\delta(2,b,B) = \{(2,\Lambda)\}$	$[2,B,2] \to b$
$\delta(2,\Lambda,Z_0) = \{(2,\Lambda)\}$	$[2,Z_0,2] \to \Lambda$

not 'live'

From lecture 11:

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

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



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

Section 6

Context-Free and Non-Context-Free Languages

6 Context-Free and Non-Context-Free Languages
- Pumping Lemma
- Decision problems

From lecture 2:
Regular language is language accepted by an FA.

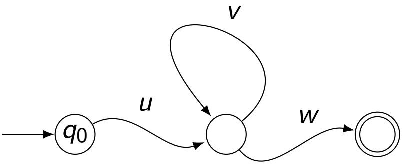
Theorem

Suppose L is a language over the alphabet Σ. If L is accepted by a finite automaton M, and if n is the number of states of M, then

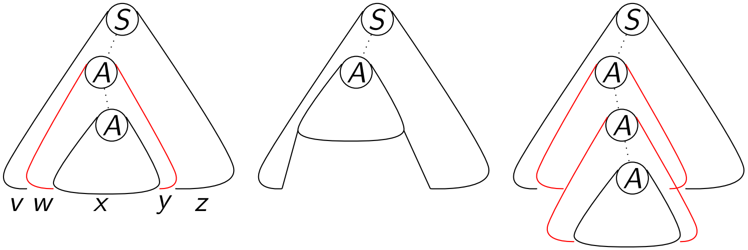
$\forall$ *for every* $x \in L$
satisfying $|x| \geq n$
$\exists$ *there are three string* u, v, *and* w,
such that $x = uvw$ *and the following three conditions are true:*
(1) $|uv| \leq n$,
(2) $|v| \geq 1$
$\forall$ *and (3) for all* $i \geq 0$, uv^iw *belongs to* L

[M] Thm. 2.29

From lecture 2:

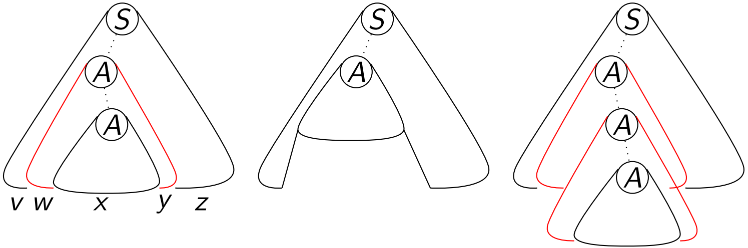


[M] Fig. 2.28


S
A
A
v
w
x
y
z
S
A
S
A
A
A

$S \Rightarrow^* vAz \Rightarrow^* vwAyz \Rightarrow^* vwxyz, \; v, w, x, y, z \in \Sigma^*$

$S \underset{(1)}{\Rightarrow^*} vAz, \; A \underset{(2)}{\Rightarrow^*} wAy, \; A \underset{(3)}{\Rightarrow^*} x$



$S \underset{(1)}{\Rightarrow^*} vAz \underset{(3)}{\Rightarrow^*} vxz$

$S \underset{(1)}{\Rightarrow^*} vAz \underset{(2)}{\Rightarrow^*} vwAyz \underset{(2)}{\Rightarrow^*} vwwAyyz \underset{(3)}{\Rightarrow^*} vwwxyyz$

Theorem (Pumping Lemma for context-free languages)

$\forall$ *for every context-free language* L

$\exists$ *there exists a constant* $n \geq 2$

such that

$\forall$ *for every* $u \in L$

with $|u| \geq n$

$\exists$ *there exists a decomposition* $u = vwxyz$

such that

(1) $|wy| \geq 1$

(2) $|wxy| \leq n$,

$\forall$ *(3) for all* $m \geq 0$, $vw^m xy^m z \in L$

[M] Thm. 6.1

Example

AnBnCn is not context-free.

[M] E 6.3

Theorem (Pumping Lemma for context-free languages)

$\forall$ *for every context-free language* L

$\exists$ *there exists a constant* $n \geq 2$

such that

$\forall$ *for every* $u \in L$

with $|u| \geq n$

$\exists$ *there exists a decomposition* $u = vwxyz$

such that

(1) $|wy| \geq 1$

(2) $|wxy| \leq n$,

$\forall$ *(3) for all* $m \geq 0$, $vw^m xy^m z \in L$

If $L = L(G)$ with G in ChNF, then $n = 2^{|V|}$.

Proof. . .

[M] Thm. 6.1

From lecture 9:

Definition

CFG in *Chomsky normal form*

productions are of the form

– $A \to BC$ variables A, B, C

– $A \to \sigma$ variable A, terminal σ

Theorem

For every CFG G there is CFG G_1 in CNF such that $L(G_1) = L(G) - \{\Lambda\}$.

[M] Def 4.29, Thm 4.30

Theorem (Pumping Lemma for context-free languages)

[M] Thm. 6.1

Proof

Let G be CFG in Chomsky normal form with $L(G) = L - \{\Lambda\}$.

Derivation tree in G is binary tree
(where each parent of a leaf node has only one child).

Height of a tree is number of edges in longest path from root to leaf node.

At most 2^h leaf nodes in binary tree of height h: $|u| \leq 2^h$.

Theorem (Pumping Lemma for context-free languages)

[M] Thm. 6.1

Proof (continued)

At most 2^h leaf nodes in binary tree of height h: $|u| \leq 2^h$.

Let p be number of variables in G,
let $n = 2^p$
and let $u \in L(G)$ with $|u| \geq n$.

(Internal part of) derivation tree of u in G has height at least p.
Hence, longest path in (internal part of) tree contains at least $p + 1$ (internal) nodes.

Consider final portion of longest path in derivation tree.
(leaf node $+$ $p + 1$ internal nodes),
with ≥ 2 occurrences of a variable A.

Pump up derivation tree, and hence u.

Application of pumping lemma:
mainly to prove that a language L cannot be generated by a context-free grammar.

How?
Find a string $u \in L$ with $|u| \geq n$ that cannot be pumped up!

What is n?
What should u be?
What can v, w, x, y and z be?
What should m be?

Suppose that there exists context-free grammar G with $L(G) = L$.
Let $n \geq 2$ be the integer from the pumping lemma.
We prove:
There exists $u \in L$ with $|u| \geq n$, such that
for every five strings v, w, x, y and z such that $u = vwxyz$
if
1. $|wy| \geq 1$
2. $|wxy| \leq n$

then
3. there exists $m \geq 0$, such that vw^mxy^mz does not belong to L

Example

AnBnCn is not context-free.

[M] E 6.3

$u = a^n b^n c^n$

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

Example

XX is not context-free.

[M] E 6.4

Applying the Pumping Lemma

Example

AnBnCn is not context-free.

[M] E 6.3

$u = a^n b^n c^n$

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

Example

XX is not context-free.

[M] E 6.4

$u = a^n b^n a^n b^n$

$\{ a^i b^j a^i b^j \mid i, j \geq 0 \}$

Example

$\{ x \in \{a, b, c\}^* \mid n_a(x) < n_b(x) \text{ and } n_a(x) < n_c(x) \}$ is not context-free.

[M] E 6.5

ABOVE

$L = \{ x \in \{a, b, c\}^* \mid n_a(x) < n_b(x) \text{ and } n_a(x) < n_c(x) \}$ is not context-free.

Proof by contradiction.

Suppose L is context-free, then there exists a pumping constant n for L.

Choose $u = a^n b^{n+1} c^{n+1}$. Then $u \in L$, and $|u| \geq n$.

This means that we can pump u within the language L.

Consider a decomposition $u = vwxyz$ that satisfies the pumping lemma, in particular $|wxy| \leq n$.

Case 1: wy contains a letter a. Then wy cannot contain letter c (otherwise $|wxy| > n$). Now $u_2 = vw^2xy^2z$ contains more a's than u, so at least $n+1$, while u_2 still contains $n+1$ c's. Hence $u_2 \notin L$.

Case 2: wy contains no a. Then wy contains at least one b or one c (or both). Then $u_0 = vw^0xy^0z = vxz$ has still n a's, but less than $n+1$ b's or less than $n+1$ c's (depending on which letter is in wy). Hence $u_0 \notin L$.

These are two possibilities for the decomposition $vwxyz$, in both cases we see that pumping leads out of the language L.

Hence u cannot be pumped.

Contradiction; so L is not context-free.

Example

The Set of Legal C Programs is Not a CFL

[M] E 6.6

Choose $u =$

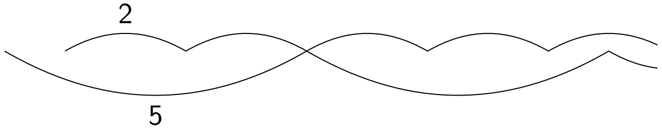
`main(){int aaa...a;aaa...a=aaa...a;}`

where `aaa...a` contains $n + 1$ a's

Lemma (⊠)

$L \subseteq \{a\}^*$ context-free, then L regular.

[M] Exercise 6.23



This exercise does not have to be known for the exam.