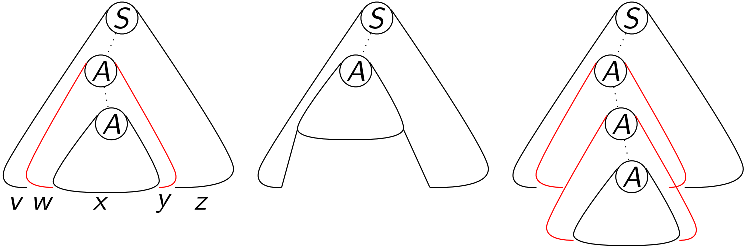
From lecture 12:

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

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



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

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

Theorem (Pumping Lemma for context-free languages)

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

$\exists$ *there exists a constant* $n \geqslant 1$

such that

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

with $|u| \geqslant n$

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

such that

(1) $|wy| \geqslant 1$

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

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

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: $|u| \leqslant 2^h$.

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: $|u| \leqslant 2^h$.

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

(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 $\geqslant 2$ 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 $|u| \geqslant n$ that cannot be pumped up!

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

Suppose that there exists context-free grammar G with $L(G) = L$. Let n be the integer from the pumping lemma.

We prove:

There exists $u \in L$ with $|u| \geqslant n$, such that

for every five strings v, w, x, y and z such that $u = vwxyz$

if

1. $|wy| \geqslant 1$
2. $|wxy| \leqslant n$

then

3. there exists $m \geqslant 0$, such that $vw^m xy^m z$ 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

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

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

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

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 $|u| \geqslant n$.

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

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

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 $u =$

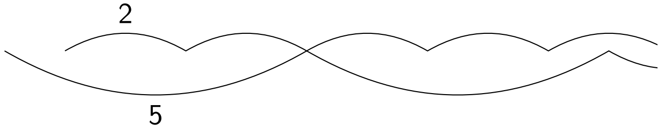
```
main(){int aaa...a;aaa...a=aaa...a;}
```

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.

From lecture 2:

FA $M_i = (Q_i, \Sigma, q_i, A_i, \delta_i) \quad i = 1, 2$

Product construction

construct FA $M = (Q, \Sigma, q_0, A, \delta)$ such that

– $Q = Q_1 \times Q_2$

– $q_0 = (q_1, q_2)$

– $\delta(\ (p, q), \sigma) = (\ \delta_1(p, \sigma), \delta_2(q, \sigma)\ )$

– A as needed

Theorem (2.15 Parallel simulation)

– *$A = \{(p, q) \mid p \in A_1$ or $q \in A_2\}$, then $L(M) = L(M_1) \cup L(M_2)$*

– *$A = \{(p, q) \mid p \in A_1$ and $q \in A_2\}$, then $L(M) = L(M_1) \cap L(M_2)$*

– *$A = \{(p, q) \mid p \in A_1$ and $q \notin A_2\}$, then $L(M) = L(M_1) - L(M_2)$*

Proof. . .

[M] Sect 2.2

From lecture 6:

Regular languages are closed under

– Boolean operations (complement, union, intersection)

– Regular operations (union, concatenation, star)

– Reverse (mirror)

– [inverse] Homomorphism

From lecture 7:
Using building blocks

Theorem

If L_1, L_2 are CFL, then so are $L_1 \cup L_2$, $L_1 L_2$ and L_1^*.

$G_i = (V_i, \Sigma, S_i, P_i)$, having no variables in common.

Construction

$G = (V_1 \cup V_2 \cup \{S\}, \Sigma, S, P)$, new axiom S
– $P = P_1 \cup P_2 \cup \{S \to S_1, S \to S_2\}$ $L(G) = L(G_1) \cup L(G_2)$
– $P = P_1 \cup P_2 \cup \{S \to S_1 S_2\}$ $L(G) = L(G_1) L(G_2)$

$G = (V_1 \cup \{S\}, \Sigma, S, P)$, new axiom S
– $P = P_1 \cup \{S \to SS_1, S \to \Lambda\}$ $L(G) = L(G_1)^*$

[M] Thm 4.9

How about

- $L_1 \cap L_2$
- $L_1 - L_2$
- L_1'

for CFLs L_1 and L_2 ?

Example

AnBnCn is intersection of two context-free languages.

[M] E 6.10

Hence, CFL is not closed under intersection

Example

AnBnCn is intersection of two context-free languages.

[M] E 6.10

Hence, CFL is not closed under intersection

$L_1 \cap L_2 = (L_1' \cup L_2')'$

Hence, CFL is not closed under complement

$L_1' = \Sigma^* - L_1$

Hence, CFL is not closed under setminus

Example

Complement of XX
$= \{ x \in \{a, b\}^* \mid |x| \text{ is odd} \} \cup \{ x\, y \mid x, y \in \{a, b\}^*, |x| = |y|, x \neq y \}$
is context-free

[M] E 6.11

Indeed, CFL is not closed under complement

Example

Complement of *AnBnCn* is context-free.
Complement of $\{x \in \{a, b, c\}^* \mid n_a(x) = n_b(x) = n_c(x)\}$ is context-free.

[M] E 6.12

Example

Complement of $AnBnCn$ is context-free.
Complement of $\{x \in \{a, b, c\}^* \mid n_a(x) = n_b(x) = n_c(x)\}$ is context-free.

$AnBnCn = L_1 \cap L_2 \cap L_3$, with
$L_1 = \{a^i b^j c^k \mid i \leqslant j\}$
$L_2 = \{a^i b^j c^k \mid j \leqslant k\}$
$L_3 = \{a^i b^j c^k \mid k \leqslant i\}$
[M] E 6.12

Example

Complement of *AnBnCn* is context-free.
Complement of $\{x \in \{a, b, c\}^* \mid n_a(x) = n_b(x) = n_c(x)\}$ is context-free.

$\{x \in \{a, b, c\}^* \mid n_a(x) = n_b(x) = n_c(x)\} = A_1 \cap A_2 \cap A_3$, with
$A_1 = \{x \in \{a, b, c\}^* \mid n_a(x) \leqslant n_b(x)\}$
$A_2 = \{x \in \{a, b, c\}^* \mid n_b(x) \leqslant n_c(x)\}$
$A_3 = \{x \in \{a, b, c\}^* \mid n_c(x) \leqslant n_a(x)\}$
[M] E 6.12

Example

$L_1 = \{\ a^{2n} b^n \mid n \geqslant 1\ \}^*$

$a^{16} b^8 a^8 b^4 a^4 b^2 a^2 b^1$

$L_2 = a^* \{\ b^n a^n \mid n \geqslant 1\ \}^* \{\ b\ \}$

From lecture 2:

FA $M_i = (Q_i, \Sigma, q_i, A_i, \delta_i)$ $\quad i = 1, 2$

Product construction

construct FA $M = (Q, \Sigma, q_0, A, \delta)$ such that

– $Q = Q_1 \times Q_2$

– $q_0 = (q_1, q_2)$

– $\delta(\ (p, q), \sigma) = (\ \delta_1(p, \sigma), \delta_2(q, \sigma)\ )$

– A as needed

Theorem (2.15 Parallel simulation)

– $A = \{(p, q) \mid p \in A_1$ *or* $q \in A_2\}$, *then* $L(M) = L(M_1) \cup L(M_2)$

– $A = \{(p, q) \mid p \in A_1$ *and* $q \in A_2\}$, *then* $L(M) = L(M_1) \cap L(M_2)$

– $A = \{(p, q) \mid p \in A_1$ *and* $q \notin A_2\}$, *then* $L(M) = L(M_1) - L(M_2)$

Proof. . .

[M] Sect 2.2

Theorem

If L_1 is a CFL, and L_2 in REG, then $L_1 \cap L_2$ is CFL.

[M] Thm 6.13

product construction

PDA $M_1 = (Q_1, \Sigma, \Gamma, q_1, Z_1, A_1, \delta_1)$
FA $M_2 = (Q_2, \Sigma, q_2, A_2, \delta_2)$

$Q = Q_1 \times Q_2 \qquad q_0 = \langle q_1, q_2 \rangle \qquad A = A_1 \times A_2$

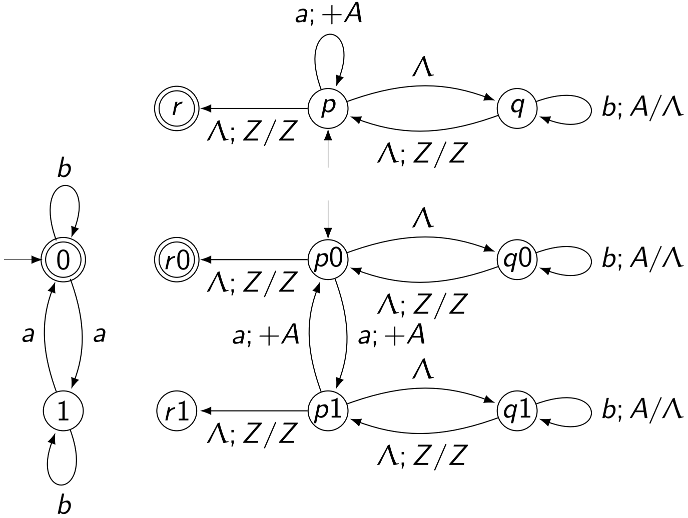
$\delta(\langle p, q \rangle, \sigma, X) \ni (\langle p', q' \rangle, \alpha)$
whenever $\delta_1(p, \sigma, X) \ni (p', \alpha)$ and $\delta_2(q, \sigma) = q'$

$\delta(\langle p, q \rangle, \Lambda, X) \ni (\langle p', q \rangle, \alpha)$
whenever $\delta_1(p, \Lambda, X) \ni (p', \alpha)$ and $q \in Q_2$

The inductive proof that this construction works does not have to be known for the exam.

Also CFG proof

$\{ \ a^n b^n \mid n \geqslant 1 \ \}^* \cap \{ \ w \in \{a, b\}^* \mid n_a(x) \text{ even } \}$



Non-determinism of PDA

- enables $L(M_1) \cup L(M_2)$
- 'prevents' $L(M_1)'$ (also Λ-transitions)

If L is accepted by DPDA without Λ-transitions, then so is L'

Even: if L is accepted by DPDA, then so is L'

Hence, if L is CFL and L' is not, then there is no DPDA for L

"given a CFL L, does it have property ... ?" yes/no
input CFG G

Given CFG G [G_1 and G_2]
– and given a string x, is $x \in L(G)$? membership problem
convert G to ChNF, and try all derivations of length $2|x| - 1$
(special case if $x = \Lambda$)
Cocke, Younger, and Kasami (1967)
Earley (1970)

– is $L(G) \neq \varnothing$? non-emptiness

  is S useful?

  pumping lemma

– is $L(G)$ infinite?

  pumping lemma

– is $L(G_1) \cap L(G_2)$ nonempty?
– is $L(G) = \Sigma^*$?
– is $L(G_1) \subseteq L(G_2)$?

$$L(G) = \Sigma^*, \text{ if and only if } \Sigma^* \subseteq L(G)$$

All undecidable

Given context-free L and regular R

– is $R \subseteq L$?

– is $L \subseteq R$?

ABOVE

$R \subseteq L$?

Special case $R = \Sigma^*$

$\Sigma^* \subseteq L$ iff $L = \Sigma^*$ undecidable

$L \subseteq R$?

iff $L \cap R' = \varnothing$

regular languages are closed under complement

CFL closed under intersection with regular languages

emptiness context-free decidable

Section 7

Course Computability

Contents

END.

Thanks to HJH for the slides