CIS 262 Fall 2009: Preparing for Midterm 2

Midterm 2 will be held on Tuesday, November 10 during class. It will be open book: you are allowed to use your own notes and the textbook. All topics covered in lectures from Chapters 5, 6, 7 are relevant. Note: we did not discuss 5.3 (Applications of CFGs), 5.4.4 (Inherent ambiguity), 6.4 (DPDAs), 7.1 (Normal forms), 7.3.1, 7.3.2, 7.3.5 (Homomorphisms and inverse homomorphisms), 7.4.1 (Complexity of CFG and PDA translations), 7.4.2 (Conversion to Chmosky normal form), and 7.4.4 (Testing membership). So you can skip these topics.

Solving following problems would be a good way to prepare for the midterm. Last year's midterm and solutions are also enclosed.

1. Exercise 5.1.5 from textbook
2. A grammar G is called *left-linear* if all the rules are of the form $X \to \varepsilon$ or $X \to Ya$ (X, Y are variables, and a is a terminal). Prove that (1) if G is left-linear then $L(G)$ is regular, and (2) if L is regular then there is a left-linear grammar G such that $L(G) = L$.
3. Let $\Sigma = \{a, b, \#\}$. Draw a PDA P such that $N(P) = \{u\#v \mid u, v \in \{a, b\}^*, u \neq v^R\}$. That is, P should accept (by empty stack) all words that contain exactly one # symbol and the word before # is *not* the reverse of the word following #.
4. Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA. Prove that for all states $p, q \in Q$, for all stack contents $\alpha, \beta, \gamma \in \Gamma^*$, and for all words $u, v \in \Sigma^*$, if $(q, u, \alpha) \vdash^* (p, v, \beta)$ then $(q, u, \alpha\gamma) \vdash^* (p, v, \beta\gamma)$.
5. Exercise 6.2.1
6. Exercise 6.2.4
7. Exercise 6.3.2
8. Exercise 7.2.1
9. Exercise 7.2.2
10. For a grammar G, a variable X is reachable if $S \Rightarrow^* \alpha X \beta$. Describe an algorithm for computing all the reachable variables.

CIS 262: Fall 2009: Solutions to Selected Practice Problems for Midterm 2

Note that the problems marked with $*$ in the textbook have solutions on the book's website. Here are some additional solutions.

Problem 3

Describe a PDA P such that $N(P) = \{u\#v \mid u, v \in \{a,b\}^*, u \neq v^R\}$.

$P = (\{q_0, q_1, q_2\}, \{a,b\}, \{Z_0, a, b\}, \delta, q_0, Z_0)$. Note that $u \neq v^R$ iff either $|u| \neq |v|$ or $|u| = |v|$ but $\exists w$ such that $u = u'aw$ and $v = wbv'$. The following transition function δ captures this property.

- (Push every symbol in the stack till you see #)
  $\delta(q_0, c, X) = \{q_0, cX\}$ for $c \in \{a,b\}$ and $X \in \{a, b, Z_0\}$.
- (Go to "pop" mode after seeing # and match symbols as many as possible)
  $\delta(q_0, \#, X) = \{q_1, X\}$ for $X \in \{a, b, Z_0\}$.
  $\delta(q_1, c, c) = \{q_1, \epsilon\}$ for $c \in \{a,b\}$.
- (Go to "accept" mode if there is a mismatch)
  $\delta(q_1, a, b) = \{q_2, b\}$.
  $\delta(q_1, b, a) = \{q_2, a\}$.
- (Go to "accept" mode if $|u| \neq |v|$)
  $\delta(q_1, \epsilon, c) = \{q_2, c\}$ for $c \in \{a,b\}$ (here $|u| > |v|$).
  $\delta(q_1, c, Z_0) = \{q_2, Z_0\}$ for $c \in \{a,b\}$ (here $|u| < |v|$).
- (Consume the rest of the input symbols and stack symbols in the "accept" mode)
  $\delta(q_2, c, X) = \{q_2, X\}$ for $c \in \{a,b\}$.
  $\delta(q_2, \epsilon, X) = \{q_2, \epsilon\}$ for $X \in \{a, b, Z_0\}$.

Note that Z_0 can get popped if and only if the PDA is in state q_2, i.e. when $u \neq v^R$.

Problem 4

Let $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA. Prove that for all states $p, q \in Q$, for all stack contents $\alpha, \beta, \gamma \in \Gamma^*$, and for all words $u, v \in \Sigma^*$, if $(q, u, \alpha) \vdash^* (p, v, \beta)$ then $(q, u, \alpha\gamma) \vdash^* (p, v, \beta\gamma)$.

We prove the claim by induction on the number of steps n in the sequence of IDs. More precisely, fix q, u, α, γ. We want to show that for all n, for all states p, for all words v, and for all stack contents β, if $(q, u, \alpha) \vdash^n (p, v, \beta)$ then $(q, u, \alpha\gamma) \vdash^n (p, v, \beta\gamma)$.

The base case is trivially true for $n = 0$, i.e. if $(q, u, \alpha) = (p, v, \beta)$, or $q = p, u = v, \alpha = \beta$ then $(q, u, \alpha\gamma) = (p, v, \beta\gamma)$.

Inductive case. Assume: for all states all states p, for all words v, and for all stack contents β, if $(q, u, \alpha) \vdash^n (p, v, \beta)$ then $(q, u, \alpha\gamma) \vdash^n (p, v, \beta\gamma)$. Consider a state p, word v and stack β. Assume $(q, u, \alpha) \vdash^{n+1} (p, v, \beta)$. We want to prove that $(q, u, \alpha\gamma) \vdash^{n+1} (p, v, \beta\gamma)$.

We know that (q, u, α) goes to (p, v, β) in $n+1$ steps. Suppose the ID after n steps is (q', u', α'). That is, $(q, u, \alpha) \vdash^n (q', u', \alpha') \vdash (p, v, \beta)$. By induction hypothesis, $(q, u, \alpha\gamma) \vdash^n (q', u', \alpha'\gamma)$. Note here that our induction hypothesis included the universal quantifier over the components of the end ID, which allowed us to use it correctly.

Let's look at the transition used in the last step $(q', u', \alpha') \vdash (p, v, \beta)$ Suppose $u' = aw$, where $a \in \Sigma \cup \{\epsilon\}$ and $\alpha' = Z\alpha''$. Note that α' has to be non-empty (otherwise the PDA cannot take the last step). Since $(q', u', \alpha') \vdash (p, v, \beta)$, $\delta(q', a, Z) \ni (p, \eta)$, where $v = u'$ or $v = w$ (depending on $a = \epsilon$ or not respectively) and $\beta = \eta\alpha''$. Then $(q', u', \alpha'\gamma) = (q', u', Z\alpha''\gamma) \vdash (p, v, \eta\alpha''\gamma)$ (using $(p, \eta) \in \delta(q', a, Z)$ for both the cases $v = u'$ or $v = w$) $= (p, v, \beta\gamma)$. That is, we have proved $(q, u, \alpha\gamma) \vdash^{n+1} (p, v, \beta\gamma)$.

Exercise 7.2.1.e

Prove that $L = \{a^n b^n c^i \mid n \leq i \leq 2n\}$ is not a CFL.

Proof by contradiction. Assume that L is a CFL. Let n be the constant of the Pumping Lemma for L. Consider the word $z = a^n b^n c^{2n}$. z is in L, and $|z| \geq n$. The Pumping Lemma tells us that z can be written as $uvwxy$ such that $|vwx| \leq n$, vx is non-empty, and for all i, uv^iwx^iy is in L.

First, note that v can contain only one type of symbols, that is, v contains only a's or only b's or only c's (reason: if v contains two different symbols, then repeating v will destroy the pattern "a's followed by b's followed by c's", that is, uv^2wx^2y is not in $a^*b^*c^*$, and hence, not in L). By same reason, x contains only one type of symbols.

Case x does not contain any c's. In this case uwy has $2n$ c's, and $|uwy| < 3n$, so uwy cannot be in L (number of c's exceeds the sum of a's and b's).

Case x contains some c's and v is empty. In this case, uv^2wx^2y has n a's n b's and $> 2n$ c's. Hence, not in L.

Case x contains some c's and v is non-empty. In this case, since v contains only one type of symbol, say a's (case when v has only b's is similar). Now uwy has n b's but less than n a's, so not in L.

So in each case, some pumping is not L. Hence, a contradiction.

CIS 262: Fall 2008: Midterm 2, November 13, 12.00–1.20pm

Please write your answers succinctly and rigorously.

1. Consider the grammar G with a single variable R, which is also the start variable, over the alphabet $\{0, 1, +, *, (,)\}$:

$$R \rightarrow 0 \mid 1 \mid R* \mid R+R \mid (R)$$

   (a) Show a left-most derivation of the word $0 + (1 * +0)$ — 5pts
   (b) Show a parse tree for the word $(0 + 1) * +1$ — 5pts
   (c) Is G in Chomsky Normal Form? — 5pts
   (d) Is G ambiguous? Justify your answer in one or two sentences, no proof is needed. — 7pts
   (e) If $L(G)$ regular? Justify your answer in one or two sentences, no proof is needed. — 8pts

2. Give a PDA for the following language: — 30pts

$$\{ 0^i 1 0^j 1 0^k \mid k < i + j \}$$

   Your PDA may use acceptance by final state or acceptance by empty stack, make sure to state your choice clearly. There is no need to give a proof that your PDA indeed accepts the above language, but explain how your PDA works in a few sentences.

3. For each of the statements below, state whether the statement is True or False. In either case, give a short justification (one or two sentences, or a counter-example). — 30pts

   (a) For a PDA A, $L(A) = N(A)$.
   (b) Context-free languages are closed under intersection.
   (c) If A is an NFA then $L(A)$ is a context-free language.

4. State the Pumping Lemma for CFLs, and prove that the following language is not a CFL using it: — 30pts

$$\{ a^i b^j c^k \mid i = k \text{ and } j > i \}$$

5. For a language L, the set of its prefixes, denoted $Pref(L)$, contains all the prefixes of the words in L. That is, a word x belongs to $Pref(L)$ precisely when there exists another word y such that xy is in L.

   (a) Consider the language L, over the alphabet $\{(,)\}$ specified by the grammar:

$$S \rightarrow \varepsilon \mid (S)S$$

      i. Describe $Pref(L)$ in words. — 5pts
      ii. Give a grammar for $Pref(L)$. No proof is needed. — 5pts

   (b) Let G be a context-free grammar over alphabet Σ with variables V, start variable S, and productions P.

      i. Describe how to construct a set P' of productions to obtain a grammar G' such that $L(G') = Pref(L(G))$. Note: G' has same variables V and same start variable S. — 10pts
      ii. Prove that G' accepts a word if and only if it belongs to $Pref(L(G))$. — 10pts

CIS 262: Fall 2008: Midterm 2 Solutions

Problem 1

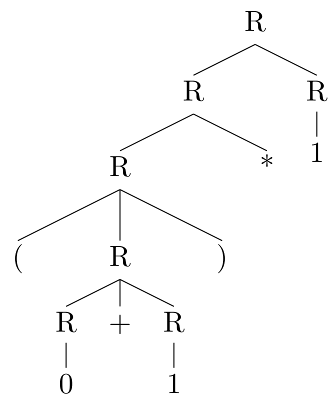
Consider the grammar G with a single variable R, which is also the start variable, over the alphabet $\{0, 1, +, *, (,)\}$:

$$R \rightarrow 0 \mid 1 \mid R* \mid R+R \mid (R)$$

1. Show a left-most derivation of the word $0 + (1 * +0)$

   $R \Rightarrow_{lm} R{+}R \Rightarrow_{lm} 0{+}R \Rightarrow_{lm} 0{+}(R) \Rightarrow_{lm} 0{+}(R{+}R) \Rightarrow_{lm} 0{+}(R{*}{+}R) \Rightarrow_{lm} 0{+}(1{*}{+}R) \Rightarrow_{lm} 0 + (1 * +0)$

2. Show a parse tree for the word $(0 + 1) * +1$

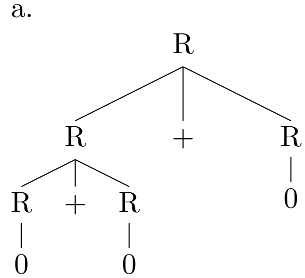
3. Is G in Chomsky Normal Form?

   No. Rules such as $R \rightarrow (R)$ are not allowed in Chomsky Normal Form.

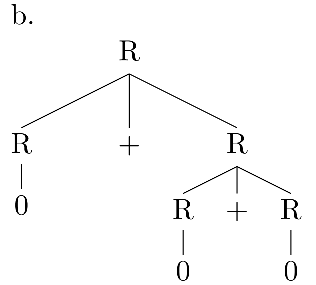
4. Is G ambiguous?

   Yes. For example, there are two different parse trees for $0 +\ 0 +\ 0$:

   a.

   b.

5. If $L(G)$ regular?

   No. Every word in $L(G)$ has same number of left brackets and right brackets, which is not a regular requirement. The fact that $L(G)$ is not regular can be proved using Pumping Lemma, and considering the word $(^n0)^n$.

Problem 2

Give a PDA for the following language:

$$\{\ 0^i10^j10^k \ \mid\ k < i + j\ \}$$

The following PDA is accepting L above by final state: $P = (\{q_0, ..q_3\}, \{0, 1\}, \{0, \$\}, \delta, q_0, \$, \{q_3\}\}$
The δ function is defined as:

- $\delta(q_0, 0, s) = \{(q_0, 0s)\}$
- $\delta(q_0, 1, s) = \{(q_1, s)\}$
- $\delta(q_1, 0, s) = \{(q_1, 0s)\}$
- $\delta(q_1, 1, s) = \{(q_2, s)\}$
- $\delta(q_2, 0, 0) = \{(q_2, \epsilon)\}$
- $\delta(q_2, \epsilon, 0) = \{(q_3, \epsilon)\}$

In the above, s ranges over stack symbols $\{0, \$\}$.

The idea: in state q_0, push the symbol 0 to the stack for each of the first block of 0's. In case of 1, more to q_1 that pushes 0 to the stack for each of the second block of 0's. After the second 1, there are $i + j$ 0's in the stack, and the PDA is in state q_2.

State q_2 pops 0s from the stack as long as the next symbol in w is 0. We can move non-deterministically to the accepting state, q_3 if the next symbol in the stack is 0. In case $k \geq i + j$, we get to the initial stack symbol, \$, so we cannot get to the accepting state, q_3.

Problem 3

For each of the statements below, state whether the statement is True or False.

1. For a PDA A, $L(A) = N(A)$.

   False. For example, consider a PDA A with two states: $\{q_0, q_1\}$, where q_0 is the initial state, and q_1 is accepting, and the only transition is $\delta(q_0, 0, X) = \{(q_1, aX)\}$. $N(A) = \emptyset$ since A never has an empty stack, but $L(A) = \{0\}$.

2. Context-free languages are closed under intersection.

   False. For example $L_1 = \{0^i10^j10^k | i = j\}$ and $L_2 = \{0^i10^j10^k | j = k\}$ are both CFLs. However, $L_1 \cap L_2 = \{0^i10^j10^k | i = j = k\}$, which is not a CFL.

3. If A is an NFA then $L(A)$ is a context-free language.

   True. If A is an NFA then $L(A)$ is a regular language, and a regular language is also a CFL.

Problem 4

State the Pumping Lemma for CFLs, and prove that the following language is not a CFL using it:

$$\{ \ a^i b^j c^k \ \mid \ i = k \text{ and } j > i \ \}$$

The pumping lemma: If L is a CFL, there exists some n such that for all words z such that $z \in L$ and $|z| \geq n$ we can split z into $z = uvwxy$ where:

- $|vwx| \leq n$
- $|vx| > 0$
- for all $i \geq 0$, $uv^iwx^iy \in L$

Suppose $L = \{ \ a^i b^j c^k \ \mid \ i = k \text{ and } j > i \ \}$ is a CFL, and assume n is the pumping lemma constant. Consider $z = a^n b^{n+1} c^n \in L$, $|z| > n$, then we should be able to split $z = uvwxy$ as defined above. Note that vwx cannot contain both a's and c's since there are $n+1$ b's between the a's and the b's.

Consider the following cases:

1. v contains one or more a symbols. In this case neither v nor x contains c's, and therefore uwy would contain a smaller number of a's then c's and therefore not in L.
2. x contains one or more c symbols. This case is symmetrical to the previous case.
3. Otherwise, v does not contain any a's and x does not contain any c's. This can only happen when vwx contains only b's, and since $|vx| > 0$, the number of b's in $z' = uwy$ decreases without affecting the number of a's and c's, so $\#b(z') \leq \#a(z')$, and hence $z' \notin L$.

Since there is no way to split z so that the pumping lemma would hold, L is not a CFL.

Problem 5

For a language L, the set of its prefixes, denoted *Pref*(L), contains all the prefixes of the words in L.

1. Consider the language L, over the alphabet $\{(,)\}$ specified by the grammar:

$$S \ \rightarrow \ \varepsilon \ \mid \ (S)S$$

   (a) Describe *Pref*(L) in words.

   A language over the alphabet $\{(,)\}$, such that for any $w \in$ *Pref*(L), any prefix of w does not contain more right parentheses than left parentheses.

   (b) Give a grammar for *Pref*(L).

$$S \ \rightarrow \ \varepsilon \ \mid \ (S)S \mid \ (S$$

2. Let G be a context-free grammar over alphabet Σ with variables V, start variable S, and productions P.

   (a) Describe how to construct a set P' of productions to obtain a grammar G' such that $L(G') = \mathit{Pref}(L(G))$. Note: G' has same variables V and same start variable S.

   This question is not solvable without adding variables to V', the set of variables for G'. We define V' as: $V' = V \cup \{X' | X \in V\}$, where for each production rule $X \to x_0...x_k \in P$, P' contains the production rules:

   - $X \to x_0...x_k \in P'$ (the same as in P)
   - For every $j \leq k$, $X' \to x_0...x_{j-1}x'_j \in P'$ where $x'_j = a$ if $x_j = a \in \Sigma$ (a terminal), and $x'_j = Y'$ if $x_j = Y \in V$

   The start variable for G' is S' (the primed variable corresponding S).

   (b) Prove that G' accepts a word if and only if it belongs to $\mathit{Pref}(L(G))$.

   First, note that for every $X \in V$ and $w \in \Sigma^*$, $X \Rightarrow^*_G w$ iff $X \Rightarrow^*_{G'} w$ (this is trivial, since the production rules for all the variables in X in G' are identical to those in G).

   In order to prove the desired property of primed variables, we prove a stronger property by induction on the derivation of w: for every $X \in V$, $X' \Rightarrow^*_{G'} w$ iff for some $a \in \Sigma^*$, $X \Rightarrow^*_G wa$ (that is, X' derives all prefixes of words derivable from X).

   $\Leftarrow$ Our induction hypothesis is that for every $m \leq n$, if for some word a, $X \Rightarrow^m_G wa$ then for some $k \leq m$ $X' \Rightarrow^k_{G'} w$. We want to prove the same holds for $n+1$.

   Base If $X \Rightarrow w'$ is a one-step derivation, then the derivation rule we used contains only terminals. In this case, from the definition of P' we can see that for every w such that $X \to w$, X' can be written to all prefixes of w'.

   Step In case $X \Rightarrow^{n+1} w'$, we need to show that if we split w' into $w' = wa$ then $X' \Rightarrow^k w$ for some $k \leq n+1$.

   Assuming the first production rule used for the derivation of X is $X \to x_0...x_k$, consider the smallest i such that $x_0...x_i \Rightarrow^n w''$ and $\exists b.w'' = wb$.

   - If x_i is a terminal then $w'' = w$ (otherwise, we could remove x_i and get a shorter string), and since P' contains the rule $X' \to x_0...x_i$, it follows from the induction hypothesis, that $X' \Rightarrow x_0...x_i \Rightarrow^n w$.
   - If x_i is a variable Y, then $w'' = w_1w_2w_3$ such that $w_1w_2 = w$ and $Y \Rightarrow^m w_2w_3$ (for some $m \leq n$). By the induction hypothesis, $Y' \Rightarrow w_2$, and therefore $X' \Rightarrow x_0...x_{i-1}Y' \Rightarrow^k w$ (where $k \leq n$).

   $\Rightarrow$ Our induction hypothesis is that for every $m \leq n$, if $X' \Rightarrow^m_{G'} w$, then for some word a, $X \Rightarrow_G wa$ We want to prove the same holds for $n+1$.

   Base For one step of derivation, if $X' \Rightarrow w$, the rule used is in the form $X' \to x_0...x_i$ such that for some $j \geq i$ there the rule $X \to x_0...x_i...x_j$ in P.

   Step In case $X' \Rightarrow^{n+1}_{G'} w$, the first rule used is in the form $X' \to x_0...x'_i$, and for some $j \geq i$ there is the rule $X \to x_0...x_i...x_j$ in P.

   If x_i is a terminal, then we can take a such that $x_{i+1}...x_j \Rightarrow^* a$, and therefore $X \Rightarrow^* wa$

   If x_i is a variable, and $x'_i \Rightarrow^* w''$ then according to the induction hypothesis, there is a b such that $x_i \Rightarrow w''b$. We can select a such that $bx_{i+1}x_j \Rightarrow^* a$, and in this case $X \Rightarrow^* wa$.