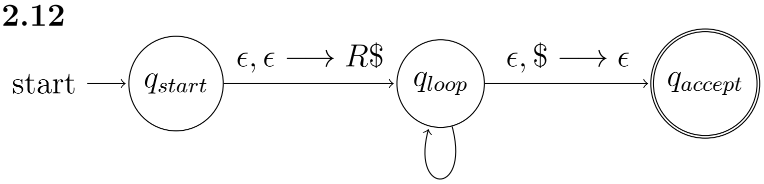
CS 4330 HW5 Solution

2.12



$\epsilon, R \rightarrow XRX$

$\epsilon, R \rightarrow S$

$\epsilon, S \rightarrow aTb$

$\epsilon, S \rightarrow bTa,$

$\epsilon, T \rightarrow XTX,$

$\epsilon, T \rightarrow X,$

$\epsilon, T \rightarrow \epsilon,$

$\epsilon, X \rightarrow a,$

$\epsilon, X \rightarrow b,$

$a, a \rightarrow \epsilon,$

$b, b \rightarrow \epsilon$

2.13

a) $L(G)$ is a set of strings of 0s and s that either contain exactly 2 s and any number of 0s, or contain exactly 1 and the number of 0s to the right of the is twice the number of 0s to the left.

b) Assume for contradiction that $L(G)$ is regular. Suppose p is the pumping length. Consider string $w = 0^p 0^{2p}$. Since $|w| > p$, by pumping lemma, we can split $w = xyz$ such that $|xy| \leq p, y \neq \epsilon$ and $xy^i z \in L(G)$.

Since $|xy| \leq p$, xy cannot contain any and since $|y| > 0$, y must contain at least one 0. Suppose $y = 0^k$ for some $k > 0$. Then $xy^2 z = 0^{p+k} 0^{2p} \notin L(G)$. Thus, $L(G)$ is not regular

2.25

To show that the class of context free languages is closed under the SUFFIX operation, let's create a PDA that recognizes it. Suppose A is a context free language. Let $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ be the PDA that recognizes A.

Let $M' = (Q \cup \hat{Q}, \Sigma, \Gamma, \delta', \hat{q_0}, F)$ be the PDA that recognize SUFFIX(A)

For every $q \in Q$ we will create a corresponding $\hat{q} \in \hat{Q}$. The new start state for M' is $\hat{q_0}$. The transition functions for M' is defined as

$$\delta'(\hat{q}, a, b) = \begin{cases} (\hat{p}, c|(p, c) \in \delta(q, a, b) \\ \\ (p, c)|(p, c) \in \delta(q, a, b) \end{cases}$$

$\delta'(q, a, b) = \delta(q, a, b)$ for $q \in Q$

2.26

Suppose $CFG\ G$ is in Chomsky normal form. Then by definition, all of its production ruls are of the form $A \to BC$ or $A \to a$. Note that each rule of form ABC adds a new variable and each rule of form $A \to a$ converts one variable to a terminal. Since it takes $n-1$ steps to derive start variable into n variables, and then n steps to convert each of the variable into terminals, it takes $2n-1$ steps to derive string of length n.

2.30

a)

$A = \{0^n 1^n 0^n 1^n | n \geq 0\}$

Suppose the given language A is a CFL. Let p be the pumping length. Consider the string $s = 0^p 1^p 0^p 1^p \in A$. As it has the length at least p, it can be written as $s = uvxyz$ satisfying the conditions of the pumping lemma. Since $|vxy| \leq p$, there are 2 possibles cases

1) If vxy is in the first half of s, then $0^p 1^p$ is the suffix of $uv^0 xy^0 z$. However, $uv^0 xy^0 z$ has length less than $4p$, so it cannot be of the form $0^n 1^n 0^n 1^n$. Similarly, the argument holds if vxy

is in the second half of s.

2) If vxy straddles in the middle, then as $|vxy| \leq p$. 01^p is the suffix of uv^0xy^0z. However, uv^0xy^0z has length less than $4p$, so it cannot be of the form $0^n1^n0^n1^n$.

d)

$D = \{t_1\#t_2\#\Delta\Delta\Delta\#t_k|k \geq 2, eacht_i \in a, b^*, andt_i = t_j forsomei \neq j\}$

Suppose the given language D is a CFL. Let p be the pumping length. Consider the string $s = 0^p1^p\#0^p1^p \in D$. As it has the length at least p, it can be written as $s = uvxyz$ satisfying the conditions of the pumping lemma. Since $|vxy| \leq p$, there are 2 possibles cases

If vxy contains # character, then pumping uv^0xy^0z down will remove the # character from s, which will no longer in D.

If vxy doesn't contain #, then vyx can only be on one side before or after #. Then, uv^2xy^2z will result in string that is not in D, which is a contradiction.

2.31

Let B be the language of all palindromes over $\{0, 1\}$ containing an equal number of 0s and 1s.Suppose the given language B is a CFL. Let p be the pumping length. Consider the string $s = 0^p1^p0^p1^p \in B$. From exercise 2.32 a, we proved that s violates pumping lemma. Thus, B is not context free language.

2.32

Suppose C is a CFL. Let p be the pumping length and $s = 1^p3^p2^p4^p \in C$. Since $|s| \geq p$, we can write $s = uvxyz$ satisfying the condition of the pumping lemma. Since $|vxy| \leq p$, there are 3 possibles cases

1) If vxy is substring of 1^p3^p, then in uv^2xy^2z, either the number of 1s is greater than the number of 2s, or the number of 3s is greater than number of 4, or both. Thus, $uv^2xy^2z \notin C$, which is a contradiction.

2) If vxy is substring of 2^p4^p, then similar to 1), $uv^2xy^2z \notin C$, which is a contradiction.

3) If vxy is a substring of 3^p2^p then in uv^2xy^2z, either the number of 2s is greater than the number of 1s or the number of 3s is greater than number of 4s, or both. Thus, $uv^2xy^2z \notin C$, which is a contradiction.

2.44

To prove that $A \diamond B$ is context free, let's create a PDA M_{AB} that recognize it. Let M_A and M_B be DFAs that recognize A and B, respectively. Let M_{AB} be a PDA as follows

1. Push a \$ to the stack and go to the start state of M_A

2. For every symbol from A, we pushes x to the stack.

3. If we ever transition to the accept state of A, create an ϵ-transition to start state of B. By doing this, we can nondeterministically guess the end of string in A.

4. For ever character read from B, we pop a character in the stack

5. If \$ is read, it means we are at the bottom of the stack. Therefore, we will accept. Otherwise, reject.

2.39 Refer to Problem 1.42 for the definition of the shuffle operation. Show that the class of context-free languages is not closed under shuffle.

Proof: Let $L_1 = \{ a^i b^j c^k \mid 0 \leq i < j, k \geq 0 \}$ and $L_2 = \{ a^i b^j c^k \mid 0 \leq j < k, i \geq 0 \}$. They are both CF as L_1 and L_2 can be generated by CFGs $G_1 = (\{S_1, A\}, \{a, b, c\}, S_1, P_1)$ and $G_2 = (\{S_2, B\}, \{a, b, c\}, S_2, P_2)$, where $P_1 = \{ S_1 \to S_1c \mid A;\ A \to aAb \mid aAbb \mid b \}$ and $P_2 = \{ S_2 \to aS_2 \mid B;\ B \to bBc \mid bBcc \mid c \}$, respectively.

If shuffle(L_1, L_2) is CF, then shuffle($L_1 \cup L_2$) $\cap$ a*b*c* should be CF, too. However, shuffle($L_1 \cup L_2$) $\cap$ a*b*c* = C = $\{ a^i b^j c^k \mid 0 \leq i < j < k \}$, a language proved not CF in Example 2.37 (page 128). The contradiction comes from the assumption that shuffle(L_1, L_2) is CF. So shuffle(L_1, L_2) is not CF.