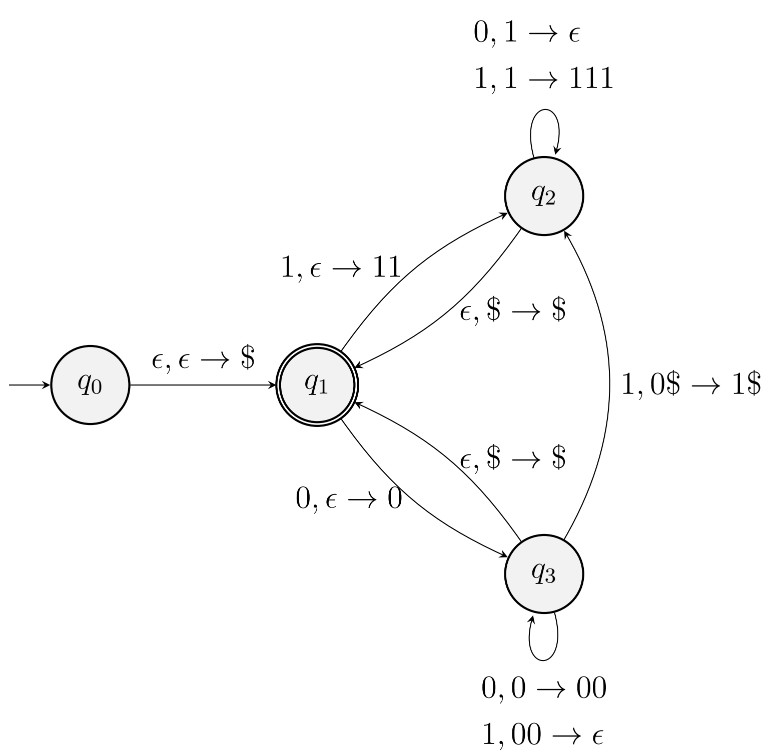
COMS 3261, Computer Science Theory (Fall 2023): Assignment 2 Solutions

Instructions

- All problems 6 are worth 10 points.
- Submit your solutions in pdf format. Late homeworks will **not** be accepted.
- You can discuss with TAs, the prof, and other students, but please acknowledge them at the beginning of each problem. All of your solutions must be written in your own words.

Problems

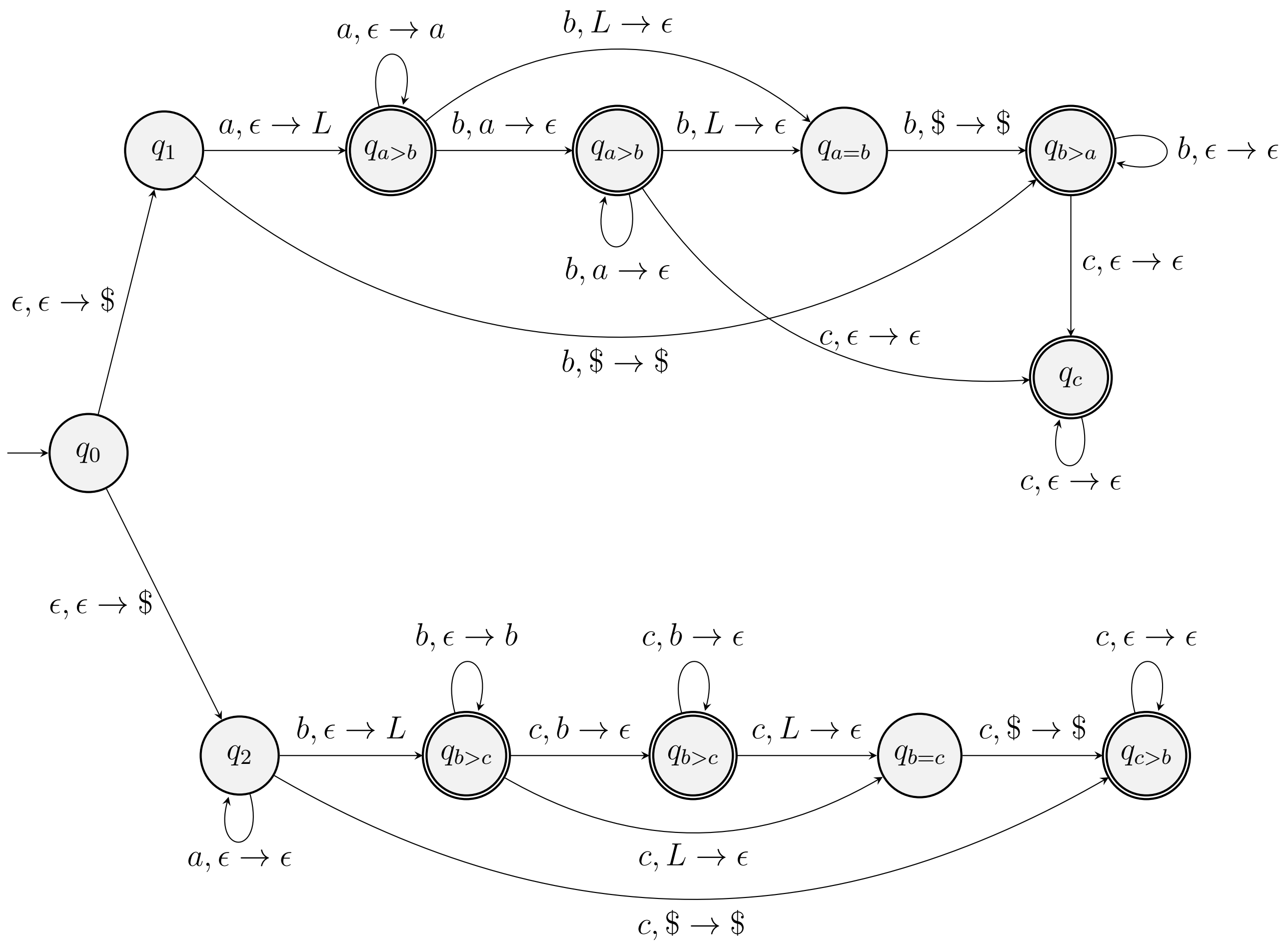
1. a. At any given time, the stack should have either all 0's or all 1's. If it is just 0's on the stack, then the number of 0's in the stack equals to the number of 0's so far minus twice the number of 1's seen so far. If it is just 1's on the stack, the number of 1's on the stack equals twice the number of 1's seen so far minus the number of 0's seen so far. Thus, we can construct the following PDA:



b. The language can be written as $L_1 \cup L_2 \cup L_3 \cup L_4$, where:

$$L_1 = \{a^i b^j c^k \mid i < j\}$$
$$L_2 = \{a^i b^j c^k \mid i > j\}$$
$$L_3 = \{a^i b^j c^k \mid j < k\}$$
$$L_4 = \{a^i b^j c^k \mid j > k\}$$

For lanauge $L_1 \cup L_2$, we push an a every time we read an a, then we pop an a every time we read a b. In order to check when the number of b's is equal to the number of a's, we push a special symbol L on the first a. The same is done for $L_3 \cup L_4$ for the symbols b and c. Thus, we can construct the following PDA:



2. a. (Extra Credit) We can construct the following CFG:

$$S \to A \mid B \mid AB \mid BA$$
$$A \to a \mid aAa \mid aAb \mid bAb \mid bAa$$
$$B \to b \mid aBa \mid aBb \mid bBb \mid bBa$$

We claim that AB and BA never generates a string of the form ww. A generates string of form $(a+b)^k a(a+b)^k$ with length $2k+1$, $k \geq 0$. B generates strings of form $(a+b)^k b(a+b)^k$, $k \geq 0$. Thus, the string generated by AB has the form $(a+b)^k a(a+b)^k (a+b)^{k'} b(a+b)^{k'}$ with length $2k+1+2k'+1$. This is not of the form ww since positions $k+1$ and $\frac{2k+1+2k'+1}{s} + k + 1 = (k+k'+1) + k + 1$ differ.

Using the claim above, we can show this grammar generates exactly the string not of form ww. String $x \in \{ww\}$ if and only if one of the following conditions hold:

(1) $|x|$ is odd

(2) $|x|$ is even and $\exists i$ such that $x_i \neq x_{\frac{|x|}{2}+i}$

Our grammar generates all x satisfying (1), since rules $S \to A$, $S \to B$ and A generates all odd length strings with a in the middle, B generates all odd length strings with b in the middle. For proving (2), let string $x = x^1x^2$, $|x^1| = |x^2| = n$. Since $x^1 \neq x^2$, let them differ in position i, say $x^1_i = a$ and $x^2_i = b$. Let s_1 be the substring of x^1x^2 of length $2i-1$, let s_2 be the remaining part of x^1x^2 with length $2n - (2i-1) = 2(n-i)+1$. Since s_1 has odd length and its middle symbol is an a, $A \stackrel{*}{\Rightarrow} s_1$. Similarly, since s_2 has odd length and middle symbol is a b, $B \stackrel{*}{\Rightarrow} s_2$. Thus $S \to AB \stackrel{*}{\Rightarrow} s_1s_2 = x^1x^2$. The same could be proved for $x^1_i = b$ and $x^2_i = a$, $S \to BA \stackrel{*}{\Rightarrow} s_1s_2 = x^1x^2$.

b. The language can be considered as the union of the following:

$$
\begin{aligned}
L_1 &= \{a^i b^i \mid i < j\} \\
L_2 &= \{a^i b^j \mid j > i\} \\
L_3 &= b(a+b)^* + a(a+b)^* b(a+b)^* a(a+b)^*
\end{aligned}
$$

We can construct a CFG for each of the above languages:

$$
\begin{aligned}
L_1 :& S_1 \rightarrow aS_1b \mid S_1b \mid b \\
L_2 :& S_2 \rightarrow aS_2b \mid aS_2 \mid a \\
L_3 :& S_3 \rightarrow bA \mid aAbAaA \\
& A \rightarrow \epsilon \mid aA \mid bA
\end{aligned}
$$

Then we can define the CFG for the complement of $\{a^n b^n \mid n \geq 0\}$ as $V = \{S, S_1, S_2, S_3, A\}$, and S is the start variable:

$$
\begin{aligned}
S &\rightarrow S_1 \mid S_2 \mid S_3 \\
S_1 &\rightarrow aS_1b \mid S_1b \mid b \\
S_2 &\rightarrow aS_2b \mid aS_2 \mid a \\
S_3 &\rightarrow bA \mid aAbAaA
\end{aligned}
$$

3. a. Assume $L = \{a^i b^j \mid j = i^2\}$ is context free. Let p be the pumping length, choose string $w = a^p b^{p^2}$. There must exist some partition $w = uvxyz$, $|vxy| \le p$, $|vy| \ge 1$, such that $uv^m xy^m z \in L$, where $m \ge 0$. There are 4 possible partitions for w:

Case 1) vxy are all a's
Suppose $vxy = a^n a^k a^m$, and $n + m > 0$. We can pump up: $w' = uv^2xy^2z = a^{p+n+m} b^{p^2}$, but $(p + n + m)^2 \ne p^2$, thus $w' \notin L$.

Case 2) vxy are all b's
Suppose $vxy = b^n b^k b^m$, and $n + m > 0$. We can pump up: $w' = uv^2xy^2z = a^p b^{p^2+n+m}$, but $p^2 \ne p^2 + n + m$, thus $w' \notin L$.

Case 3) Either v has both a's and b's, or y has both a's and b's
Suppose $v = a^n b^m$ and $xy = b^k$. We can pump up: $w' = uv^2xy^2z$ since v^2 is of form $a^n b^m a^n b^m$ but all strings in L must have form a^*b^*, $w' \notin L$. Similarly, suppose $vx = a^k$ and $y = a^n b^m$. Since y^2 is of the form $a^n b^m a^n b^m$ but all strings in L must have form a^*b^*, $w' \notin L$.

Case 4) v has just a's and y has just b's
Let $u = a^i$, $v = a^j$, $x = a^{p-i-j} b^k$, $y = b^l$, $z = b^{p^2-k-l}$, $j + l \ge 1$. Choose $w' = uv^{p^2+1} xy^{p^2+1} z$, then w' has $p + p^2 i$ a's and $p^2 + p^2 j$ b's. However, since $p > 0$ and $j \le p$, $(p + p^2 i)^2 = p^2 + p^4 i^4 + 2p^3 i > p^2 + p^2 j$, $w' \notin L$.

We find a contradiction, therefore L is not context free.

b. Assume $L = \{a^i \mid i \text{ is prime}\}$ is context free. Let p be the pumping length, choose string $w = a^{p'}$, where $p' > p$ and p' is a prime number. There must exist some partition $w = uvxyz$, $|vxy| \le p$, $|xy| \ge 1$, such that $uv^m xy^m z \in L$, where $m \ge 0$. Suppose $|vy| = k \ge 1$, we can pump up: $m = p' + 1$, then $w' = uv^{p'+1} xy^{p'+1} z = uvv^{p'} xyy^{p'} z$. Since w' is in form a^*, $w' = uvxyzv^{p'} x^{p'} = a^{p'} a^{kp'} = a^{p'+kp'}$. However, $p'(k + 1)$ is not a prime number, therefore $w' \notin L$, and L is not context free.

4. We can prove that CFL's are not closed under complement by providing a counter example. Let $L = \{a^i b^j c^k \mid \text{either } i \neq j \text{ or } j \neq k\}$, let $L_0 = \{w \mid w \neq a^* b^* c^*\}$. We know that both L and L_0 are CFL's. $L_1 = \{a^n b^n c^n \mid n \geq 0\}$ is not CFL. We can write $\overline{L_1} = L \cup L_0$. Assume CFLs are closed under complement, then since CFLs are also closed under union and L and L_0 are both CFLs, $\overline{L_1}$ must be CFL and L_1 must also be CFL. However, we know L_1 is not a CFL, so there is a contradiction. Therefore CFLs are not closed under complement.