CS340: Theory of Computation
Assignment 3 Solutions

Yatharth Goswami
Roll No. 191178

October 6, 2021

I Problem 1 Solution

In this problem, we were asked to devise a way to identify if the two strings a and b provided to us are equal. For this we were asked to check if the two strings have the same length or not.

(a) For this part we first define the language to check if the length of strings a and b are equal. For this we first introduce an extra symbol # to the alphabet, therefore $\Sigma = \{0, 1, \#\}$. Now, we define the language, for this problem as $L = \{w_1 \# w_2 \mid |w_1| = |w_2|,\ w_1, w_2 \in \{0,1\}^*\}$. So, in order to find if two strings a and b have the same length we will check if the string $a\#b$ belongs to the language L. Now, we need to design PDA which accepts this language. Below is the PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, F)$ which accepts the language L, with

- $Q = \{q_0, q_1, q_2, q_3\}$
- $\Sigma = \{0, 1, \#\}$
- $\Gamma = \{\$, A\}$
- $F = \{q_3\}$

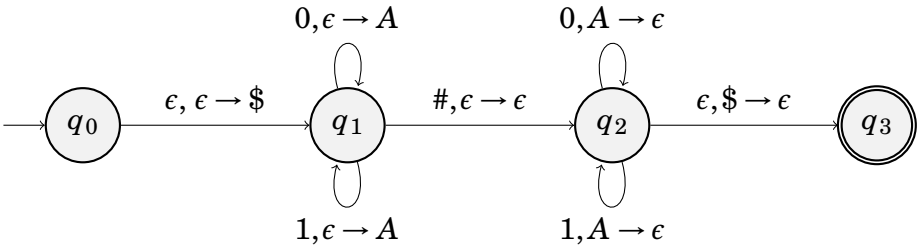

Figure 1: PDA for 1(i)

(b) Since we already know that $|a| = |b|$, we just need to check if the corresponding characters are also the same. Consider this language $L = \{ww \mid w \in \{0,1\}^*\}$. If $|a| = |b|$, then $a = b$, iff $ab \in L$.

(c) We will claim that $L = \{ww \mid w \in \{0,1\}^*\}$ is not a CFL and hence a PDA cannot be designed for it. We will prove using the contrapositive form of the pumping lemma, which says that $\forall p \geq 0$, $\exists\ w \in L$, s.t. $|w| \geq p$ s.t. $\forall w = uvxyz$ where $|vxy| \leq p$ and $|vy| > 0$, $\exists\ i \geq 0$, s.t. $uv^i xy^i z \notin L$, then L is not a CFL.

- Given a $p \geq 0$ by the opponent.
- We will choose the string $w = 0^p 1^p 0^p 1^p$.
- Opponent gives a partition $w = uvxyz$ with the above given constraints.
- We need to now come up with an $i \geq 0$ such that $uv^i xy^i z \notin L$. For this, let us consider the various possible cases.
  - It might be the case that vxy appears completely in the first part of string (i.e. in the first $2p$ length). In this case, we will take $i = 2$, and consider the string uv^2xy^2z. Notice that in this string since the length $|vy| > 0$ and $\leq p$. The length of uv^2xy^2z increases by at most p. Hence the midpoint of the string moves at most by $p/2$. We now say that string y can be of two forms $1^i \mid i \geq 0$ or $0^i 1^j \mid i, j \geq 0$. Notice in both the cases the string yz starts with p number of 1s. Hence the first block of 1s (in $0^p 1^p 0^p 1^p$) will remain intact in uv^2xy^2z as well. Now, since the first block of 1s will be moving to the right since we

are adding another v and y string in the first half, therefore if we consider the first block of 1s (in $0^p1^p0^p1^p$) to move by $k = |vy| > 0$, then midpoint will only move by $k/2 \leq p/2$. Hence, we can be sure that the second half will start with a 1. And since the first half starts with a 0, this string cannot belong to L.

- It might be the case that vxy appears completely in the second part of string (i.e. in the second $2p$ length). We In this case, we will take $i = 2$, and consider the string uv^2xy^2z. Similar to the above explanation in this case the midpoint will be move by a distance $x/2$ where $x = |vy|$ to the right. Since $x/2 \leq p/2$ and second block of 0s have length p, we can be sure that the first half will end with a 0, while the second half will end with a 1. Hence, this string cannot belong to L.
- In the third case vxy lies entirely in the middle segment of w (contains at least one character from the first $2p$ and at least one character from the last $2p$ length). Note that in this case, vxy can only be a part of the middle 1^p0^p segment since $|vxy| \leq p$. In this case, we will take $i = 0$. Consider the string uxz, this string will be of the form $0^p1^x0^y1^p$ where $x \leq p$ and $y \leq p$ and both cannot be simultaneously p since $|vy| > 0$. For the string to have two equal halfs, the second half should start with p zeros and first half should end with p ones, which is not possible. Hence, this string also cannot lie in L.

Hence by enumerating all possible cases, we showed that for each choice of partition, there exists an i for which $uv^ixy^iz \notin L$. Hence, using pumping lemma we can say that L is not a CFL and hence there will not exist any PDA which accepts L.

II Problem 2 solution

In this problem we were asked to show that $L = \{a^n b^j c^k \mid k = jn, j, k, n \geq 0\}$ is not context free. We will employ the contrapositive form of pumping lemma for proving this which says that $\forall p \geq 0$, $\exists\, w \in L$, s.t. $|w| \geq p$ s.t. $\forall w = uvxyz$ where $|vxy| \leq p$ and $|vy| > 0$, $\exists\, i \geq 0$, s.t. $uv^i xy^i z \notin L$, then L is not a CFL.

- Opponent gives us a $p \geq 0$.
- We choose the string $w = a^p b^p c^{p^2}$.
- He gives us a partition $w = uvxyz$ with the above given constraints.
- Now there can be different cases possible. Say, $|vy| = k > 0$
  - vxy contains only 1 symbol. In this case, suppose first if vxy is contained inside only a or b. Without loss of generality, assume that vxy is contained completely inside the block of a's. In this case, we will choose $i = 0$ and the string $uv^0xy^0z = a^{p-k}b^p c^{p^2}$ which doesn't belong to L since $p*(p-k) \neq p^2$ for $k > 0$. In the other case, if vxy lies entirely in the block of c's. We will again choose $i = 0$ and string $uv^0xy^0z = a^p b^p c^{p^2-k}$ again will not lie in L since $p^2 \neq p^2 - k$ for $k > 0$.
  - vxy contains of 2 symbols. In this case, suppose first that vxy contains both a and b. In this case, we choose $i = 0$ and we get the string $uv^0xy^0z = a^l b^m c^{p^2}$ where both $l \leq p$ and $m \leq p$ and both not simultaneously equal to p since we are removing vy with $|vy| > 0$. Therefore $l * m < p^2$ and hence this string will not lie in L.

    Now, consider the case when vxy lies between blocks of b and c. In this case, choosing $i = 0$, will give $uv^0xy^0z = a^p b^{p-k_1} c^{p^2-k_2}$. We also know that $k_1 + k_2 = k > 0$ and hence both can't be simultaneously 0. We claim that this string will not belong to set L. Suppose, on contrary the string $a^p b^{p-k_1} c^{p^2-k_2}$ belongs to L, then we have the relation.

    $$p*(p-k_1) = p^2 - k_2$$
    $$k_2 = p*k_1$$

    From the above equation, we get that if $k_1 = 0$ then $k_2 = 0$ but this can't be true since $k_1 + k_2 = k > 0$. In all the other cases, $k_2 + k_1 = k_1 * (p+1)) = k$. If we choose k_1 anything other than 0, we get that $k > p$, which can't be true since $k \leq |vxy| \leq p$. Hence, we arrive at a contradiction and the original string is not present in L.
  - vxy cannot contain all the 3 symbols since then $|vxy| > p$ since it will have to contain all the b's in the middle compulsorily. Hence, this case is not possible.

  Therefore we proved that for each possible partition $uvxyz$ we can choose $i = 0$ and get a string uxz which is not in L. Hence L is not context free.

III Problem 3 Solution

In this problem, we were asked to provide the Context-free-grammar that will accept the provided language.

(a) $L_1 = \{a^i b^j c^k \mid i = j \text{ or } i \neq k \, i, j, k \geq 0\}$

Let us define a grammar $G = (V, \Sigma, P, S)$ in this way

- $V = \{S, S_1, S_2, S_3, S_4, S_5, T_1, T_2\}$
- $\Sigma = \{a, b, c\}$
- Set of Production rules P:
  $S \rightarrow S_1 \mid S_2$
  $S_1 \rightarrow T_1 T_2$
  $T_1 \rightarrow aT_1b \mid \epsilon$
  $T_2 \rightarrow cT_2 \mid \epsilon$
  $S_2 \rightarrow aS_2c \mid S_3 \mid S_4$
  $S_3 \rightarrow S_3c \mid S_5c$
  $S_4 \rightarrow aS_4 \mid aS_5$
  $S_5 \rightarrow bS_5 \mid \epsilon$
- Start State = S

Description of each non-terminal

- S generates all strings in L.
- S_1 generates all strings of the form $a^i b^j c^k \mid i = j, k \geq 0$.
- T_1 generates all strings of the form $a^n b^n \mid n \geq 0$.
- T_2 generates all strings of the type $c^n \mid n \geq 0$.
- S_2 generates all strings of the form $a^i b^j c^k$ such that $i \neq k$ and $i, j, k \geq 0$.
- S_3 generates all strings of the form $b^i c^j \mid i \geq 0, \; j > 0$.
- S_4 generates all strings of the form $a^i b^j \mid i > 0, \; j \geq 0$.
- S_5 generates all strings of the form $b^i \mid i \geq 0$.

(b) Given a grammar $G = (V, \Sigma, P, S)$ with the set of production rules as
$S \rightarrow BSB \mid B \mid \epsilon$
$B \rightarrow 00 \mid \epsilon$
Let us apply the algorithm to convert into CNF.

- In the first step of conversion to CNF, we will remove any instance of start state on the right hand side. We get the new set of production rules as follows by introducing a new start state S_0
  $S_0 \rightarrow S$
  $S \rightarrow BSB \mid B \mid \epsilon$
  $B \rightarrow 00 \mid \epsilon$
- Removing ϵ transitions. First we remove the transition $B \rightarrow \epsilon$
  $S_0 \rightarrow S$
  $S \rightarrow BSB \mid B \mid BS \mid SB \mid \epsilon$

$B \to 00$
Note that $S \to S$ will also be added above but it is equivalent to just S. Hence we can remove it as well.
Now, removing the transition $S \to \epsilon$
$S_0 \to S \mid \epsilon$
$S \to BSB \mid B \mid BS \mid SB$ $B \to 00$

- Removing the single variable rule. Removing the transition $S \to B$ first
  $S_0 \to S \mid \epsilon$
  $S \to BSB \mid BS \mid SB \mid 00$
  $B \to 00$
  Now, removing the transition $S_0 \to S$.
  $S_0 \to BSB \mid BS \mid SB \mid 00 \mid \epsilon$
  $S \to BSB \mid BS \mid SB \mid 00$
  $B \to 00$

- Shortening the RHS. Removing the rule $S_0 \to BSB$ and the rule $S \to BS$, by introducing a new variable $V \to BS$.
  $S_0 \to VB \mid BS \mid SB \mid 00 \mid \epsilon$
  $S \to VB \mid BS \mid SB \mid 00$
  $V \to BS$
  $B \to 00$

- Adding variables. In this step we will remove the rules like $B \to 00$, $S_0 \to 00$ and $S \to 00$, using two new variables U_1 and U_2.

  $S_0 \to VB \mid BS \mid SB \mid U_1U_2 \mid \epsilon$
  $S \to VB \mid BS \mid SB \mid U_1U_2$
  $V \to BS$
  $B \to U_1U_2$
  $U_1 \to 0$
  $U_2 \to 0$

The Chomsky normal form gives a grammar $G' = (V', \Sigma, P', S_0)$, with the final set of production rules P' as:

$S_0 \to VB \mid BS \mid SB \mid U_1U_2 \mid \epsilon$
$S \to VB \mid BS \mid SB \mid U_1U_2$
$V \to BS$
$B \to U_1U_2$
$U_1 \to 0$
$U_2 \to 0$

and $V' = \{S_0, S, B, U_1, U_2, V\}$.