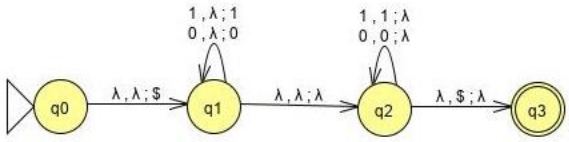
CSE 355: Intro to Theoretical Computer Science
Recitation #10 Solution

1. [5 pts] Given the following PDA state diagram, applied procedure we learned in class (Lemma 2.27, pp.140) to generate the following grammar rules:



(Note: rules with ** will be eliminated later on)

1.1) Rule between q_0 and q_3, use A_{03} to represent the variable.

$A_{03} \longrightarrow A_{12} \mid A_{00}\ A_{03}\ (**) \mid A_{01}A_{13} \mid A_{02}A_{23} \mid A_{03}\ A_{33}\ (**)$

1.2) Rule between q_1 and q_2, use A_{12} to represent the variable.

$A_{12} \longrightarrow 1A_{12}1 \mid 0A_{12}0 \mid A_{10}\ A_{02} \mid A_{11}A_{12}\ (**) \mid A_{12}A_{22}\ (**) \mid A_{13}\ A_{32}\ (**)$

1.3) Rule between q_0 and q_2, use A_{02} to represent the variable.

$A_{02} \longrightarrow\ A_{00}\ A_{02}\ (**) \mid A_{01}A_{12}\ \mid A_{02}A_{22}\ (**) \mid A_{03}\ A_{32}\ (**)$

1.4) Rule between q_1 and q_3, use A_{13} to represent the variable.

$A_{13} \longrightarrow A_{10}\ A_{03}\ \mid A_{11}A_{13}\ \mid A_{12}A_{23} \mid A_{13}A_{33}\ (**)$

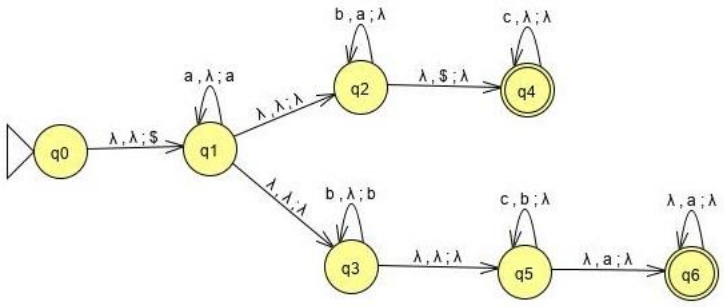
1.5) Rule between q_1 and q_1, use A_{11} to represent the variable.

$A_{11} \longrightarrow \varepsilon\ \mid A_{10}\ A_{01}\ (**) \mid A_{11}A_{11}\ (**) \mid A_{12}A_{21}(**) \mid A_{13}A_{31}\ (**)$

2. [6 pts] Decide whether the following languages are regular, context-free or not, if they are, draw the relevant DFA/NFA or PDA's state diagram; if not, use pumping lemma to prove it.

2.1) $L = \{a^i b^j c^k \mid i = j\ or\ j = k\}$

Yes, this is a context-free language, but not regular. The PDA's state diagram is as follows:



2.2) $L = \{a^i b^j a^i b^j \mid i, j \geq 0\}$

This is NOT a context-free language. We will use pumping lemma to prove it.

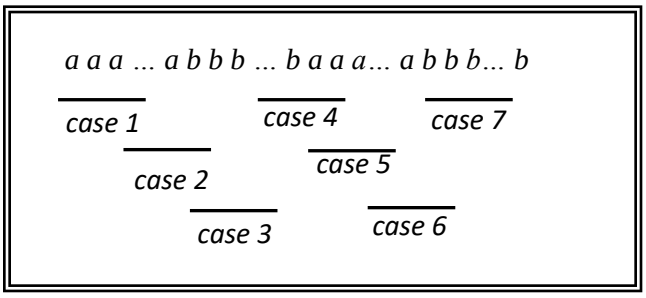
Proof by contradiction:

Step #1: Let us assume L is context-free, then according to pumping lemma, there must exist a pumping length p such that, for any string $\omega \in L$, as long as $|\omega| \geq p$, it can be broken into five pieces(strings), $\omega = uvxyz$, such that:
1) $v, y \; cannot \; be \; both \; \varepsilon$
2) $|vxy| \leq p$
3) for all $i \geq 0, the \; string \; uv^i xy^i z \; must \; also \in L$

Step #2: We will pick one special string $\omega \in L$ and make sure $|\omega| \geq p$ so that we can apply above rules. The special string we pick is $\omega = a^p b^p a^p b^p$, notice $\omega \in L \; and \; |\omega| = 4p \; which \; is > p$, so we should be able to apply pumping lemma to this string.

Step #3: We will get a contradiction from the string we picked in step #2. According to pumping lemma, $\omega = a^p b^p a^p b^p$ can be broke into five strings, $\omega = uvxyz$, such that:
1) $v, y \; cannot \; be \; both \; \varepsilon$
2) $|vxy| \leq p$
3) for all $i \geq 0, the \; string \; uv^i xy^i z \; must \; also \in L$

From above condition #2, since substring vxy length $\leq p$, it might fall into one of the following 7 areas of ω , we will analysis each of the following 7 cases.



Case 1:
For this case, where sub-string vxy contains all a_s from the first half of the original string ω, and $|vxy| \leq p$, then apply condition #3, if we pump v and y multiple times, the resulting string will be $a^{p+i} b^p a^p b^p$, $i > 0, must$ also $\in L$. But this cannot be true since the resulting string is not of the form $a^i b^j a^i b^j$ anymore, so we get a contradiction here.

Case 2:
For this case, where sub-string vxy might contain both a_s and b_s, but vxy falls into the first half of the original string ω. (Note: you can't assume where vxy locates). When applying condition #3, if we pump v and y multiple times, the resulting string will be of the form $a^{p+i} b^{p+j} a^p b^p, \; i, \; j \geq 0 \; but \; cannot \; be \; both \; 0, must$ also $\in L$. But this cannot be true since the resulting string's first half is different from the second form, *i.e.* the resulting string is not of the form $a^i b^j a^i b^j$ anymore, so we get a contradiction here.

Case 3:
For this case, where sub-string vxy contains all b_s from the first half of the original string ω and $|vxy| \leq p$, then apply condition #3, if we pump v and y multiple times, the resulting string will be $a^p b^{p+i} a^p b^p$, $i > 0, must$ also $\in L$. But this cannot be true since the resulting string is not of the form $a^i b^j a^i b^j$ anymore, so we get a contradiction here.

Case 4:
For this case, where sub-string vxy might contain both b_s and a_s, but vxy falls between the first and second half of the original string ω. When applying condition #3, if we pump *v* and *y* multiple times, the resulting string will be of the form $a^p b^{p+i} a^{p+j} b^p, i,\ j \geq 0\ but\ cannot\ be\ both\ 0, must$ also $\in\ L$. But this cannot be true since the resulting string's first half is different from the second form, *i.e.* the resulting string's first half contains different number of b_s and a_s from the second half, so we get a contradiction here.

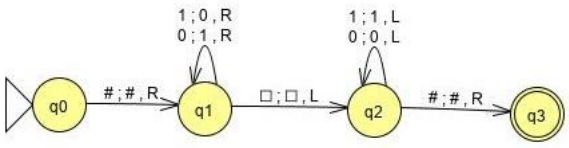
Case 5:
The argument for case #5 is similar to case #1. For this case, where sub-string vxy contains all a_s from the second half of the original string ω, and $|vxy| \leq p$, then apply condition #3, if we pump *v* and *y* multiple times, the resulting string will be $a^p b^p a^{p+i} b^p,\ i > 0, must$ also $\in\ L$. But this cannot be true since the resulting string is not of the form $a^i b^j a^i b^j$ anymore, so we get a contradiction here.

Case 6:
The argument for case #6 is similar to case #2. For this case, where sub-string vxy might contain both a_s and b_s, but vxy falls into the second half of the original string ω. (Note: you can't assume where vxy locates). When applying condition #3, if we pump *v* and *y* multiple times, the resulting string will be of the form $a^p b^p a^{p+i} b^{p+j},\ i,\ j \geq 0\ but\ cannot\ be\ both\ 0, must$ also $\in\ L$. But this cannot be true since the resulting string's first half is different from the second form, *i.e.* the resulting string is not of the form $a^i b^j a^i b^j$ anymore, so we get a contradiction here.

Case 7:
The argument for case #7 is similar to case #3. For this case, where sub-string vxy contains all b_s from the second half of the original string ω and $|vxy| \leq p$, then apply condition #3, if we pump *v* and *y* multiple times, the resulting string will be $a^p b^p a^p b^{p+i},\ i > 0, must$ also $\in\ L$. But this cannot be true since the resulting string is not of the form $a^i b^j a^i b^j$ anymore, so we get a contradiction here.

Step #4: from the arguments we did in step #3, we conclude that our assumption in step #1 must be wrong and *L* cannot be context-free!

3. [3 pts] Assume there is a special symbol # placed at the left end of the Turing machine's tape (to mark the end of the type). Given the following TM's state diagram, explain in English what this TM does?

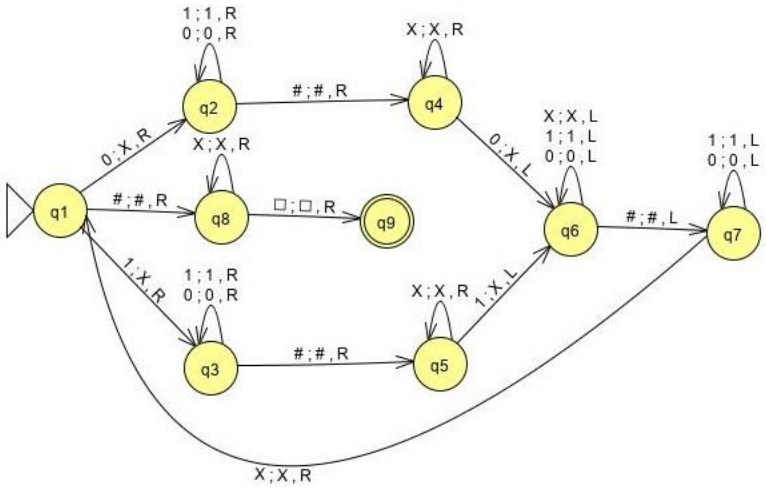


Given a binary number, this Turing machine flips all its bits, *i.e.* changing 0 to 1 or 1 to 0 and at the end, move its reading/writing head to position right to # symbol.

4. [6 pts] Given the following TM state diagram, give the sequence of configurations for the following two strings:

4.1) `11`
$\boldsymbol{q_1 11 \vdash X q_3 1 \vdash X 1 q_3 \vdash X 1 \sqcup q_3}$ **Rejected!**

4.2) `01#01`
$q_1 01\#01 \vdash X q_2 1\#01 \vdash X 1 q_2 \#01 \vdash X 1 \# q_2 01 \vdash X 1 \# 0 q_4 1 \vdash X 1 \# q_6 X 1 \vdash X 1 q_7 \# X 1 \vdash X q_7 1 \# X 1 \vdash X 1 q_1 \# X 1 \vdash X X \# q_3 X 1 \vdash X X \# X q_5 1 \vdash X X \# X 1 q_5 \vdash X X \# X q_6 X \vdash X X \# q_6 X X \vdash X X q_7 \# X X \vdash X X \# q_1 X X \vdash X X \# X q_8 X \vdash X X \# X X q_8 \vdash X X \# X X \sqcup q_9$ **Accepted!**



4.3) Use set notation to describe the language accepted by this Turing machine.

$L = \{\omega\#\omega \mid \omega \in \{0,1\}^{\star}\}$

***i.e.* odd length palindromes with a # symbol in the middle.**