Kaushal Banthia
19CS10039

① (a) $S \rightarrow ab\,S\,cB \mid \varepsilon$

$B \rightarrow bB \mid b$

This ~~language~~ grammar generates languages as follows

Eg ⇒ $ab\,(ab\,(ab\,S\,cB)\,cB)\,cB$

~~$\Rightarrow ab\,ab\,ab\,cB\,cB\,cB$~~ ⇒ $ab\,ab\,ab \ldots ab\,cB\,cB \ldots cB$

$\Rightarrow (ab)^n (cB)^n$

$\Rightarrow (ab)^n (cb^m)^n$

Thus $L = \{(ab)^n (cb^m)^n \mid n \geq 1,\ m \geq 1\}$

Also, since there is only one ~~way~~ way of generating here, it is Non Ambiguous

(b) $L_1 = \{a^l b^m c^n \mid l < m \text{ and } l < n\}$

Given a k, choose $z = a^k b^k c^k$ $\quad |z| = 3k \geq k$.

Let $z = uvwxy$ such that $vx \neq \varepsilon$ and $|vwx| \leq k$.

~~choose~~ choose i ~~=1~~. ⇒ $z' = uv^i w x^i y$.

To show $z' \notin L_1$

Case 1: v has at least one a and at least one b.
or x has at least one b and at least one c.

⇒ z' (for $i = 2$) is not of the form $a^k b^k c^k$, since v^2 will cause a $(\ldots abab\ldots)$ occurence in it, or $(\ldots aabaab\ldots)$ or $(\ldots abbabb\ldots)$ etc.

~~Case 2: v has only a's and x has only c's~~

Case 2: v and x both belong to the same alphabets

If they are contained in b's or c's then for $i=0$ there are more a's than what can be accepted.

If they are contained in a's, then for $i = 2$, there are more a's than what can be accepted.

Case 3: v and x belong to a, b respectively or b, c respectively

For a, b ⇒ we have $i=0$ if $u=\varepsilon$ ⇒ #b < #a

or

$i=2$. if $u \neq \varepsilon$ ⇒ #c < #a.

For b, c ⇒ we have. $i=0$ ⇒ #b or/and #c < #a.

Thus, by all 3 cases $z' \notin L_1$

Thus, L_1 is not context free.

(2) (a) We can have the following productions

~~S → aSa | bSb | R~~ ⇒ S → aSa | bSb | R

R → aTb | bTa

~~T → AT | BT | a | b~~ T → a|b|ε|aTb|bTa|aTa|bTb.

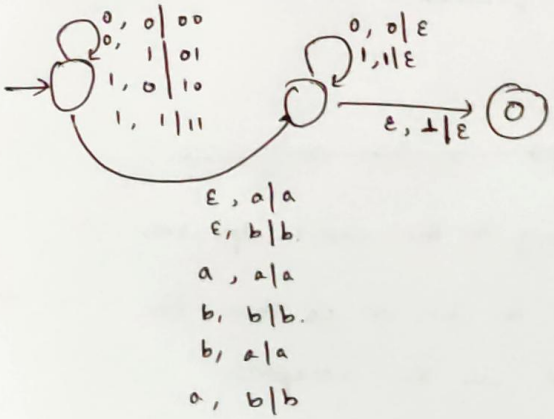
ε can be removed as follows

⇒ S → aSa | bSb | R

R → ab | ba | aTb | bTa

T → a|b|ab|ba|aa|bb|aTb|bTa|aTa|bTb.

(b) Now, we need a PDA which accepts all palindromes



Acceptance by empty stack

Kaushal Banthia
19CS10039

(3)

(3)(a) We can say that every regular language is accepted by some PDA that accepts by final state (since stack is never popped).

Here, we just ignore the stacks and the remainder of the PDA acts like a NFA with a final state.

Since an NFA accepts a regular language, thus language of PDA is also regular.