CS4384 : Automata Theory
Homework Assignment 4

Matthew McMillian

mgm160130@utdallas.edu

October 25, 2018

1. For each of the following languages, say whether it is context-free. If so, write a CFG or draw a PDA (non-mathematically) that generates the language. If not, prove it using the pumping lemma and/or closure properties for CFLs.

   a.) $L = \{a^{n^2} \mid n \geq 0\}$

   Assume L is a context free language and let p be its pumping length. Let S be a partitioning of a string in the language L such that $S = a^{p^2}$, and let S be defined in a way that satisfies the pumping lemma conditions $|vxy| \leq p$ and $|vy| > 0$. Choose i = 0, such that $S' = uv^i xy^i z \rightarrow uxz$. Since $|vy| > 0$, we know that either v or y contains an $a \in$ L's alphabet., thus $S' = a^{p^2-1}$, which is not necessarily in the language. Since we have pumped a string in the language and found that $S' \notin$ L, this contradicts the pumping lemma, and we conclude that L is not context free.

   b.) $L = \{a^m b^n c^{(m-n)} \mid m \geq n\}$

   We can define a PDA (seen below) that accepts the language L, this L is context free.
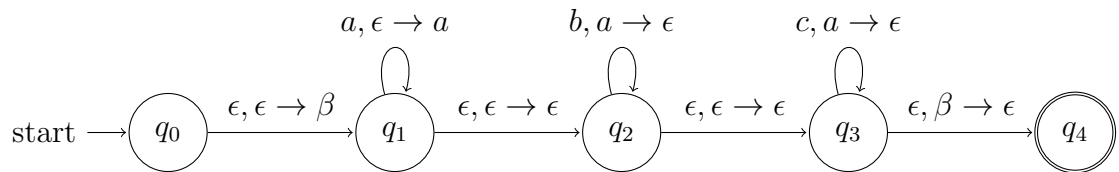
   c.) $L = \{a^m b^n c^{mn} \mid m, n \geq 0\}$

   Assume L is a context free language and let p be its pumping length. Let S be a partitioning of a string in the language L such that $S = a^p b^p c^{p^2}$, and let S be defined in a way that satisfies the pumping lemma conditions $|vxy| \leq p$ and $|vy|$

> 0. Choose i $= 0$, such that $S' = uv^ixy^iz \to uxz$. We consider 2 cases in order to analyze the string S'. Since $|vy| > 0$, we know that v and/or y must contain a character, and since $|vxy| \leq p$ we know that v,y can only be an a or b. Notice that if v,y contain any character character $\in \Sigma$. If v,y are the any character, then we obtain strings like $S' = a^{p-v_l}b^{p-y_l}c^{p^2}$, where v_l, y_l are the length of subsections v and y respectively. Notice since $|vy| > 0$, v_l, y_l must be at least one (for one of the subsections). Thus, $p - v_l * p - y_l \neq p^2$, therefore the string is not in the language. Since we have pumped a string in the language and found that $S' \notin L$, this contradicts the pumping lemma, and we conclude that L is not context free.

d.) $L = \{a^n b^n c^m \mid m > n\}$

Assume L is a context free language and let p be its pumping length. Let S be a partitioning of a string in the language L such that $S = a^p b^p c^{p+1}$, and let S be defined in a way that satisfies the pumping lemma conditions $|vxy| \leq p$ and $|vy| > 0$. Choose i $= 2$, such that $S' = uv^ixy^iz \to uv^2xy^2z$. Since $|vy| > 0$ we know that v or y must contain an element, and since $|vxy| \leq p$ we know that v,y can only be an a or b based on our partitioning. Consider the three cases below:

Case 1: v,y contain the same element(s). If v,y contain the same element(s), then trivially we can see $a^{p+v_l+y_l}b^p$ or $a^pb^{p+v_l+y_l}$ is not strictly less than $p + 1$, where v_l, y_l are the length of subsections v and y respectively. Thus, this fails the pumping lemma.

Case 2: v,y contain different element(s). Trivially we can see that v,y contain different elements of different lengths then the partitioning and subsequently the pumping lemma fails. If they are the same length, we can see that $a^{p+v_l}b^{p+y_l}$ is not strictly less than $p + 1$, thus this case fails the pumping lemma.

Since we have pumped a string in the language and found that $S' \notin L$ based on multiple cases, this contradicts the pumping lemma, and we conclude that L is not context free.

2. Define $R(L) = \{s^r \mid s \in L\}$ That is, R(L) is the language of all strings that are reverses of strings in L. Prove that R is the closure property of CFLs. *(Hint: Assume there exists a CFG in CNF that generates L, and use it to mathematically define a new CFG that generates R(L)).*

Consider a CFG in CNF defined by G $= (V,\Sigma,R,S)$. Since we know G is in CNF, we know that G does not contain a mix of terminals and non-terminals. Consider G' $= (V,\Sigma,R_{reversed},S)$ such that $R_{reversed}$ is equal to the set of rules consisting of $<$left$> \to <$right$>$, but reversed to $<$left$> \to <$right$_{reversed} >$. Since G is in CNF, we know that no non-terminals and terminals are mixed, thus if we reverse the right side, we will get anything G can create but reversed.

3. Define $C(L1, L2) = \{s1s2 \mid s1 \in L1, s2 \in L2, |s1| = |s2|\}$ That is, C(L1, L2) is a

special kind of language concatenation in which only the equal-length strings in L1 and L2 are paired together. Prove that if L1 and L2 are both regular languages, then C(L1, L2) is a CFL. (*Hint: Assume there exist DFAs A1 and A2 accepting L1 and L2, respectively, and use them to mathematically define a PDA that accepts C(L1, L2)).*

We can define a new PDA below that accepts C(L1, L2).
Q : The set of states $= \{q_0, q_1\}$
Σ^*: The alphabet defined by $\Sigma_{L1} \bigcup \Sigma_{L2}$
Γ: The alphabet of the stack $= \Sigma^*$
δ: The set of transitions given by $\{((q, q+1), \Sigma_{L1}) \mid q \in Q_{L1}, q + 1 \in Q_{L2}\} \bigcup$
$\qquad\qquad\qquad\qquad\qquad\quad \{((q, q+1), \Sigma_{L2}) \mid q \in Q_{L2}, q + 1 \in Q_{L1}\}$
$q_0 \in$ Q: The starting state
F $\subset$ Q: The accepting states s.t. the state number mod $2 = 0$.
Thus, we have proven that C(L1, L2) is a CFL.

4. PDAs with two stacks are strictly more powerful than PDAs with one stack. Prove that 2-stack PDAs are not a valid model for CFLs by giving an example of a language that is not context free and yet accepted by a 2-stack PDA. Describe (no math necessary) how a 2-stack PDA would accept that language.

Consider the language $L = \{a^n b^n c^n \mid n \geq 0\}$, which we know to be non-context-free since we proved it in class. Now consider this 2-stack PDA given below.
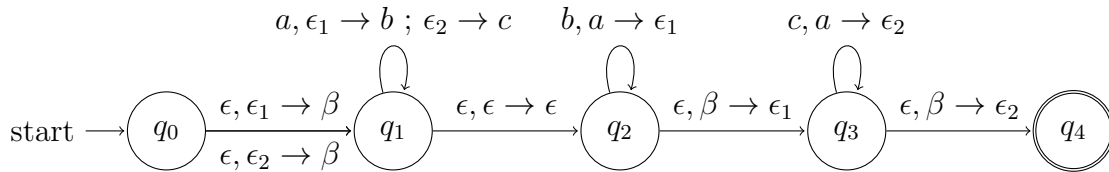
Here we have defined a 2-stack PDA that accepts our non-context free language. On each operation, the stack may either push or pop 2 elements from either stack. In this case, we show how this wouldn't work by pushing two elements onto two stacks at the same time, which allows us to pop twice separately from each stack until we reach our accepting state. This is problematic since our PDA acceptes a language that is not context free.