TOC PROBLEM SET-10

SIDDHANT CHAUDHARY
BMC201953

1. Recall the construction described in class to show that the class of CFLs is closed under inverse-homomorphism. Prove that this construction is correct.

2. Suppose L is a CFL and R is a regular language. Show that the shuffle of L and R is a CFL. Is the shuffle of two CFLs always a CFL? Why/Why not?

Solution. First, we show that the shuffle of L and R is a CFL, and our proof will be very similar to the proof in the context of regular languages. So let

$$A = (Q_A, \Sigma, \Gamma, \delta_A, s, \perp, F_A)$$

be a PDA for L, and let

$$M = (Q_M, \Sigma, \delta_M, q_0, F_M)$$

be a DFA accepting R. We make a new PDA A' which will accept the shuffle of L and R. Let

$$A' = (Q_A \times Q_M, \Sigma, \Gamma, \delta', (s, q_0), \perp, F_A \times F_M)$$

So the initial state of A' is (s, q_0) and the set of final states is $F_A \times F_M$. Next we describe the set of transitions δ'. Suppose $(q, X) \xrightarrow{c} (q', Y)$ is a transition in A (i.e $q, q' \in Q_A$, $c \in \Sigma \cup \{\epsilon\}$, $X \in \Gamma$ and $Y \in \Gamma^*$). For *every* $q'' \in Q_M$, add the transition

$$((q, q''), X) \xrightarrow{c} ((q', q''), Y)$$

in the set δ'. Next, if $q \xrightarrow{c} q'$ is a transition in M (i.e $q, q' \in Q_M$ and $c \in \Sigma$) then for *every* $q'' \in Q_A$ and $X \in \Gamma$, add the transition

$$((q'', q), X) \xrightarrow{c} ((q'', q'), X)$$

in the set δ'. In simple words, the PDA A' can randomly stimulate either A or M. Since the set of final states is $F_A \times F_M$, it is clear that A' accepts the language shuffle(L, R). This shows that shuffle(L, R) is a CFL.

Let us now show that the class of CFLs is not closed under shuffle. Consider the two languages $L_1 = \{a^n b^n \mid n \geq 1\}$ and $L_2 = \{c^n d^n \mid n \geq 1\}$, and we know that both of these languages are context free. We show that the shuffle of these two languages is *not* context free, and we do so using the pumping lemma. For the sake of contradiction, suppose shuffle(L_1, L_2) *is* context free, and let n be the pumping length for this language (as guaranteed by the pumping lemma). Consider the word

$$w = a^n c^n b^n d^n$$

which is clearly in shuffle(L_1, L_2). Moreover, the length of w is greater than n, and hence the pumping lemma can be applied. So, we can write

$$w = a^n c^n b^n d^n = uvxyz$$

where $|vxy| \leq n$, $|vy| > 0$ and $uv^i xy^i z \in$ shuffle(L_1, L_2) for each $i \geq 0$. So, we see that the subword vxy of w is non-empty. Now, there are two cases given below.

Date: October 2020.

(1) In the first case, the word vxy is composed of only a *single* letter, i.e vxy is composed of *only* one letter in the set $\{a, b, c, d\}$. Without loss of generality, suppose this letter is a. So, for sufficiently large i, the word $uv^i xy^i z$ will contain *more* a's than b's, which is a contradiction. So, this case is not possible.
(2) In the second case, the word vxy is composed of more than one letter. In this case, I claim that vxy is composed of *exactly* two letters. Because if vxy is composed of atleast three letters, it would mean that $|vxy| > n$, a contradiction. So because vxy is a contiguous and is composed of exactly two letters, the only combinations of the two letters possible are $\{a, c\}$, $\{c, b\}$ and $\{b, d\}$. Without loss of generality, suppose vxy is composed only of the letters a, c. So, for sufficiently large i, we see that $uv^i xy^i z$ contains either more a's than b's or it contains more $c's$ than d's, which is a contradiction in either case.

So, both the cases (1) and (2) give a contradiction, and hence it follows that the language shuffle(L_1, L_2) is *not* a CFL. This shows that the class of CFLs is not closed under shuffle.

3. Construct a PDA for $a^*b^*c^* \setminus \{a^n b^n c^n \mid n \geq 0\}$.

4. Suppose we generalize pushdown automata by allowing, in addition, the ability to move without examining the top of stack. Such an automaton will be able to execute even when the stack is empty. The set of transitions is now a subset of $Q \times \Sigma \cup \{\epsilon\} \times \Gamma \cup \{\epsilon\} \times Q \times \Gamma^*$. Does this model accept languages other than CFLs? Why/Why not?

Solution. No, this new model *does not* accept languages other than CFLs, and we will now prove this. So suppose

$$A = (Q, \Sigma, \Gamma, \delta, s, \perp, F)$$

where $\delta \subseteq Q \times \Sigma \cup \{\epsilon\} \times \Gamma \cup \{\epsilon\} \times Q \times \Gamma^*$, i.e suppose A is a PDA which has the ability to move without examiniing the top of stack. We will make a PDA A' (without the ability to move without examining the top of stack) such that $L(A') = L(A)$. Let $\perp_1 \neq \perp$ be a new bottom symbol, and let $s_0 \notin Q$ be a new symbol. Define a PDA A' as follows.

$$A' = (Q \cup \{s_0\}, \Sigma, \Gamma \cup \{\perp_1\}, \delta', s_0, \perp_1, F)$$

i.e s_0 is the new start state of A', and $\perp_1$ is the new stack bottom symbol for A'. Next, we describe the set of transitions δ'. First, add the transition

$$(s_0, \perp_1) \xrightarrow{\epsilon} (s, \perp\perp_1)$$

to δ'. By doing this, the PDA A' will stimulate the PDA A such that the stack symbol $\perp_1$ always remains as the bottom-most symbol in the stack. Next, if

$$(q, X) \xrightarrow{c} (q', Y)$$

is a transition in δ (where $q, q' \in Q$, $c \in \Sigma \cup \{\epsilon\}$, $X \in \Gamma$ and $Y \in \Gamma^*$), then add the transition

$$(q, X) \xrightarrow{c} (q', Y)$$

to the set δ' (in simpler words, all transitions which involve a non-empty stack in A are also included in δ'). Finally, if

$$(q, \epsilon) \xrightarrow{c} (q', Y)$$

is a transition in δ (where $q, q' \in Q$, $c \in \Sigma \cup \{\epsilon\}$ and $Y \in \Gamma^*$), then add the transition

$$(q, \perp_1) \xrightarrow{c} (q', Y \perp_1)$$

to the set δ' (so in simpler words, all transitions involving an empty stack in A have been made a transition *without* an empty stack in A'). It is then not hard to see that for *all* $q_f \in F$,

$$(s, \perp) \xrightarrow[*]{w} (q_f, \gamma) \text{ in } A \iff (s_0, \perp_1) \xrightarrow[*]{w} (q_f, \gamma \perp_1) \text{ in } A'$$

and from this, it is clear that $L(A) = L(A')$. So, this means that languages accepted by this new class of PDAs are CFLs as well, and hence this completes the proof.
Update: This solution is *almost* complete, but there is a key thing missing. Observe that the PDA can move without examining the top of the stack, so more productions need to be added, because it doesn't necessarily mean that the top of the stack is empty.

5. Suppose we generalize pushdown automata as follows: An extended PDA is a tuple $(Q, \Sigma, \Gamma, \delta, s, \perp, R)$ where the components other than R are as before. Further R is a regular language over Γ. A word w is accepted by an extended automaton if there is a run $(s, \perp) \xrightarrow{w} (q, \gamma)$ for some $q \in Q$ and $\gamma \in R$. Show that for every such automaton there is an equivalent pushdown automaton.

Solution. Let

$$A = (Q_A, \Sigma, \Gamma, \delta_A, s, \perp, R)$$

be an *extended* PDA, where R is a regular language. Since R is regular, there is a DFA M given by

$$M = (Q_M, \Gamma, \delta_M, q_0, F)$$

such that $L(M) = R$, where q_0 is the initial state of M, and $F \subseteq Q_M$ is the set of final states of M. Our strategy will be this: suppose in the PDA A, the current configuration of the PDA is (q, γ). Then, we would like to non-deterministically check whether $\gamma \in R$. If it does, then we accept the word read so far, otherwise we discard the word. Now we make all this formal.

Without loss of generality, suppose $Q_A \cap Q_M = \phi$ (i.e they are disjoint). We make a new PDA A' which will accept words via final states *and* empty stack (and in class, we have already proven that this is equivalent to the usual acceptance by final states). So define

$$A' = (Q_A \cup Q_M, \Sigma, \Gamma, \delta', s, \perp, F)$$

and so observe that the initial state of A' is the same as that of A. Also, observe that the set of final states is F, and so a word w is accepted if and only if there is a run $(s, \perp) \xrightarrow[*]{w} (q_f, \epsilon)$ in the PDA A', where $q_f \in F$. Now we describe the set of transitions δ'. Put

$$\delta' = \delta_A \cup H$$

where H is a set of transitions that we will describe in a moment. So, all transitions of A are also in A'. Next, for *every* $q \in Q_A$ and $X \in \Gamma \cup \{\epsilon\}$ (note that X is allowed to be empty since $\epsilon \in R$ is possible), add the transition

$$(q, X) \xrightarrow{\epsilon} (q_0, X)$$

in H (this is where Q_A and Q_M being disjoint is useful to us). Then, for *every* transition $q \xrightarrow{c} q'$ in M (i.e $q, q' \in Q_M$ and $c \in \Gamma$), add the transition

$$(q, c) \xrightarrow{\epsilon} (q', \epsilon)$$

in H (so in easy words, if (q, γ) is the current configuration of the PDA A, then the transitions in H allow us to non-deterministically check whether $\gamma \in R$ or not). So, by the nature of the transitions, it is not hard to see that

$$(s, \perp) \xrightarrow[*]{w} (q_f, \epsilon) \text{ in } A', q_f \in F \iff (s, \perp) \xrightarrow[*]{w} (q, \gamma) \text{ in } A \text{ for some } \gamma \in R, q \in Q_A$$

So we see that the language accepted by the extended PDA A is the same as the language accepted by the PDA A' via final states and empty stack, and hence there is an equivalent PDA for A, completing the proof.

6. Suppose you are given an extended automaton (as defined above) without ϵ-transitions. Show that you can construct an equivalent PDA without ϵ-transitions directly i.e. without appealing to any theorem proved in class. [Hint: It helps to consider a FA for reverse of R. Can you extend the stack alphabet so that you can "maintain" the state of this FA as part of the stack?]