TOC PROBLEM SET-11

SIDDHANT CHAUDHARY
BMC201953

Proposition 0.1. *Every DCFL is a CFL.*

Proof. It is enough to show that every DPDA is equivalent to some PDA. So, suppose $A = (Q, \Sigma, \Gamma, \delta, \perp, \$, s, F)$ be a DPDA, where $\$$ is the right-end marker. We will make a PDA A' which is equivalent to A. Let Q' be a copy of the set Q that is disjoint from Q, i.e $Q \cap Q' = \phi$. For any $q \in Q$, the copy of q in Q' will be denoted by q'. Now, let

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

so that F' is the new set of final states, where F' is the copy of F inside Q. Now we describe the set δ' of transitions. Suppose

$$(q, X) \xrightarrow{c} (q_1, Y)$$

is a transition in A, where $q, q_1 \in Q$, $c \in \Sigma \cup \{\epsilon\}$, $X \in \Gamma$ and $Y \in \Gamma^*$. Then, add the *same* transition to the set δ' as well. Next, suppose there is a transition

$$(q, X) \xrightarrow{\$} (q_f, Y)$$

in A, where $q, q_f \in Q$, $q_f \in F$, $X \in \Gamma$ and $Y \in \Gamma^*$. Then, add the transition

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

to the set δ'. So essentially, we have removed all transitions involving $\$$ and made them appropriate ϵ-transitions in the PDA A'. It is easy to see that $L(A) = L(A')$, and this completes the proof. ■

1. Check whether the language $L = \{w \in \{a, b\}^* \mid w \neq xx \text{ for any } x \in \{a, b^*\}\}$ is a deterministic CFL (DCFL).

Solution. In class (in Lecture 16) we showed that the language

$$L^c = \{ww \mid w \in \Sigma^*\}$$

is not a CFL. In particular, L^c is not a DCFL, since every DCFL is also a CFL by **Proposition** 0.1. Now if L was a DCFL, it would imply that L^c is also a DCFL (since DCFLs are closed under complementation), and that is a contradiction. So, it follows that L is not a DCFL. This is an important example as it shows that deterministic PDAs are not as powerful as non-deterministic ones. ■

2. Every DPDA M can be converted to a DPDA M' such that M' processes the entire input. Prove that we can convert M' to a DPDA M'' such that every transition is either a push or pop operation on the stack but not both.

Solution. To be completed. ■

Date: November 2020.

3. Prove that if L is a DCFL then there exists a Myhill-Nerode equivalence class of infinite cardinality (Use problem **2.** for a reference of the structure of a DPDA recognizing L).

Solution. Let L be a DCFL that is accepted by the DPDA M, and without loss of generality suppose M never empties its stack (it can be easily shown that every DPDA is equivalent to one that never empties its stack, and the proof is very similar to the one in the case of PDAs). Consider the set

$$S_1 := \{(q,X) \in Q \times \Gamma \mid \nexists w \in \Sigma^* \text{ s.t } (q,X) \xrightarrow[*]{w} (q', \epsilon)\ ,\ q' \in Q\}$$

Clearly, S_1 is a finite set (because it is a subset of the finite set $Q \times \Gamma$), and also S_1 is non-empty because $(s, \perp) \in S_1$ (because M never empties its stack). Now, consider the set

$$W := \{w \in \Sigma^* \mid \text{If } (s,\perp) \xrightarrow[*]{w} (q, X\gamma_w) \text{ for some } \gamma_w \in \Gamma^*\text{, then } (q,X) \in S_1\}$$

We will show that W is an *infinite* set. For the sake of contradiction, suppose W is a finite set, and hence the lengths of words in W is bounded above by some constant, say C. Now, let $w \in \Sigma^*$ by any word with $|w| > C$. So, there is a unique run

$$(s,\perp) \xrightarrow[*]{w} (q, X\gamma_w)$$

for some $q \in Q$ and $X \in \Gamma$ (recall that M never emptied its stack). Because $w \notin W$, we see that $(q,X) \notin S_1$, and hence there is some word $w_1 \in \Sigma^*$ such that

$$(q,X) \xrightarrow{w_1} (q_1, \epsilon)$$

for some $q_1 \in Q$, and hence ww_1 has the following run.

$$(s,\perp) \xrightarrow[*]{w} (q, X\gamma_w) \xrightarrow[*]{w_1} (q_1, \gamma_w)$$

Now if $\gamma_w = \epsilon$, then this contradicts the fact that M never emptied its stack. So $\gamma_w \neq \epsilon$, and we can write $\gamma_w = X_1\gamma_1$ for some $X_1 \in \Gamma$ and $\gamma_1 \in \Gamma^*$, so that $|\gamma_w| > |\gamma_1|$, and hence the run is

$$(s,\perp) \xrightarrow[*]{w} (q, X\gamma_w) \xrightarrow[*]{w_1} (q_1, X_1\gamma_1)$$

Now because $|ww_1| > C$, $ww_1 \notin W$, and hence $(q_1, X_1) \notin S_1$. So, there is some $w_2 \in \Sigma^*$ such that

$$(q_1, X_1) \xrightarrow[*]{w_2} (q_2, \epsilon)$$

for some $q_2 \in Q$, and hence ww_1w_2 has the run

$$(s,\perp) \xrightarrow[*]{w} (q, X\gamma_w) \xrightarrow[*]{w_1} (q_1, X_1\gamma_1) \xrightarrow[*]{w_2} (q_2, \gamma_1)$$

Again, if $\gamma_1 = \epsilon$, then it contradicts the fact that M never empties its stack. So, $\gamma_1 = X_2\gamma_2$ for some $X_2 \in \Gamma$, $\gamma_2 \in \Gamma^*$, so that $|\gamma_1| > |\gamma_2|$, and hence the run is

$$(s,\perp) \xrightarrow[*]{w} (q, X\gamma_w) \xrightarrow[*]{w_1} (q_1, X_1\gamma_1) \xrightarrow[*]{w_2} (q_2, X_2\gamma_2)$$

Now repeating the same argument finitely many times (the argument can't be repeated infinitely many times), we will get words $w_1, w_2, ..., w_n, w_{n+1} \in \Sigma^*$ with a run

$$(s,\perp) \xrightarrow[*]{w} (q, X\gamma_w) \xrightarrow[*]{w_1} (q_1, X_1\gamma_1) \xrightarrow{w_2} ... \xrightarrow[*]{w_n} (q_n, X_n\gamma_n) \xrightarrow[*]{w_{n+1}} (q_{n+1}, \gamma_n)$$

where $(q, X), (q_i, X_i) \notin S_1$ for $1 \le i \le n$ and

$$|\gamma_w| > |\gamma_1| > \dots > |\gamma_n| = 0$$

but again this is a contradiction to our assumption that M never empties its stack. So, it follows that W is an *infinite* set.
Now, because W is an infinite set and S_1 is a finite non-empty set, it follows (by the pigeonhole principle) that there is some $(q, X) \in S_1$ such that there are infinitely many words $w \in W$ such that

$$(s, \perp) \xrightarrow[*]{w} (q, X\gamma_w)$$

for some $\gamma_w \in \Gamma^*$. So let

$$W' := \{w \in W \mid (s, \perp) \xrightarrow[*]{w} (q, X\gamma_w) \text{ for some } \gamma_w \in \Gamma^*\}$$

so that W' is an infinite set.

Using this pair (q, X), we are all set to show that there is a Myhill-Nerode class of infinite cardinality for the language $L = L(M)$. Suppose w' is a fixed word, with $w' \in \Sigma^*$. Then starting from the configuration (q, X), we see that w' has a unique run

$$(q, X) \xrightarrow[*]{w'} (q', \alpha X)$$

for some $\alpha \in \Gamma^*$ and $q' \in Q$, and this is because $(q, X) \in S_1$, so that in this run, X will *never* be popped from the stack (because if it was popped, it would contradict the fact that $(q, X) \in S_1$). So, for *any* $\gamma \in \Gamma^*$, the run of w' starting from the configuration $(q, X\gamma)$ is of the form

$$(q, X\gamma) \xrightarrow[*]{w'} (q', \alpha X\gamma)$$

Again, this is because of the fact that $(q, X) \in S_1$, and the word γ will *never* be exposed in the stack during this run (because X will never be popped), and hence the run is *independent* of γ. In particular, we have shown that if $w \in W'$ is *any* word, then the run for the word ww' always ends up at the state q', i.e the run is

$$(s, \perp) \xrightarrow[*]{w} (q, X\gamma_w) \xrightarrow[*]{w'} (q', \alpha X\gamma_w)$$

So this means that for any $w_1, w_2 \in W'$, either $w_1w', w_2w' \in L$ or $w_1w', w_2w' \notin L$. Now because the word w' was arbitrary, this shows that all words in W' belong to the same Myhill-Nerode equivalence class, and since W' is infinite, it follows that there is a Myhill-Nerode class of infinite cardinality, completing the proof. ■

4. Can you construct a DPDA recognizing the language $\{w \in \{a, b\}^* \mid w = w^r\}$ of palindromes?

Solution. To be completed. However, the answer is a no, because every Myhill-Nerode equivalence class for this language is finite, which contradicts problem 3. ■

5. Construct a DPDA recognizing the language

$$\{a^m b^n c^n \mid m, n \geq 1\} \cup \{a^m b^n d^m \mid m, n \geq 1\}$$

[**Bonus**: Can you remove the ϵ-transitions without compromising the determinicity?]

Solution. Suppose $\Sigma = \{a, b, c, d\}$ and $\Gamma = \{\perp, A, B\}$, and let A be a DPDA defined as follows

$$A = (\{s, q_1, q_2, q_f, q_{\text{dead}}\}, \Sigma, \Gamma, \delta, \perp, \$, s, \{q_f\})$$

where $\$$ is the usual *right-endmarker*. So, there are five states s, q_1, q_2, q_f and q_{dead} where s is the starting state and the *only* final state is q_f. The state q_{dead} will act as a rejecting state. Now, we describe the set of transitions δ. First, the following transitions will be present in δ.

$(s, \perp) \xrightarrow{a} (s, A \perp)$, $(s, \perp) \xrightarrow{b,c,d,\$} (q_{\text{dead}}, \perp)$

$(s, A) \xrightarrow{a} (s, AA)$, $(s, A) \xrightarrow{b} (s, BA)$, $(s, A) \xrightarrow{c,d,\$} (q_{\text{dead}}, A)$

$(s, B) \xrightarrow{b} (s, BB)$, $(s, B) \xrightarrow{c} (q_1, \epsilon)$, $(s, B) \xrightarrow{d} (q_2, B)$, $(s, B) \xrightarrow{a,\$} (q_{\text{dead}}, B)$

$(q_1, B) \xrightarrow{c} (q_1, \epsilon)$, $(q_1, B) \xrightarrow{a,b,d,\$} (q_{\text{dead}}, B)$

$(q_1, A) \xrightarrow{\$} (q_f, A)$, $(q_1, A) \xrightarrow{a,b,c,d} (q_{\text{dead}}, A)$

$(q_2, B) \xrightarrow{\epsilon} (q_2, \epsilon)$

$(q_2, A) \xrightarrow{d} (q_2, \epsilon)$, $(q_2, A) \xrightarrow{\$} (q_2, \epsilon)$, $(q_2, A) \xrightarrow{a,b,c} (q_{\text{dead}}, A)$

$(q_2, \perp) \xrightarrow{\epsilon} (q_f, \perp)$

Then, for any $X \in \Gamma$ add the transitions

$$(q_f, X) \xrightarrow{\epsilon} (q_f, X)$$
$$(q_{\text{dead}}, X) \xrightarrow{\epsilon} (q_{\text{dead}}, X)$$

Finally, if there is some $q \in Q$ and $X \in \Gamma$ such that the above list *doesn't* contain any transition for (q, X), then just add the transition

$$(q, X) \xrightarrow{\epsilon} (q_{\text{dead}}, X)$$

It is then clear that A is indeed a DPDA by looking at the nature of the transitions in δ. Let me now roughly explain the idea. The state q_1 helps in accepting words of the form $a^m b^n c^n$, and the state q_2 helps in accepting words of the form $a^m b^n d^m$. The state s just helps in reading a's and b's and pushes A or B on the top of the stack. If we read a c, we start popping a B for every letter c read thus far, and if we read a d, then we first pop all the B's from the stack, and then we pop an A for every d read thus far. However, notice that the first d read will help in entering the state q_2 and to delete the B's, but it will be *lost*, so in the end we have to check whether the contents of the stack are $A \perp$, and if they are, we pop the A by reading $\$$ and hence the word is accepted. ■