CS 381 HW 8 Solutions

Dan FitzGerald, dpf7@cornell.edu

November 1, 2006

1. Convert the following CFG to a one state PDA:

$$\begin{array}{lll} S \longrightarrow bA & S \longrightarrow aB & \\ A \longrightarrow a & A \longrightarrow bAA & A \longrightarrow aS \\ B \longrightarrow b & B \longrightarrow bS & B \longrightarrow aBB \end{array}$$

The main idea of this problem is summarized on page 238 of the text. Basically, we plan to expand the grammar from left to right on the stack. Whenever we see terminals (a and b in this case), we will compare them to the input tape and erase the symbol from the stack if they match. When we encounter a production (S,A, or B), we will guess which production to follow and expand it onto the stack. We continue this process of eating the terminals at the top of the stack and expanding productions until no productions remain, and we will continue comparing terminals to the input and erasing them if they match. In the end, we will reach an empty stack and accept if the string is in the grammar, and thus have a PDA for the grammar.
More formally, our PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$is defined as:

$$Q \quad = \quad \{1\}, \Sigma = \{a, b\}, \delta, q_0 = 1, Z_0 = Z_0, S = 1, F = K$$

where $\delta$is defined as

$$\begin{aligned} \delta(1, \epsilon, S) &\rightarrow (1, bA), \{(1, aB)\} \\ \delta(1, \epsilon, A) &\rightarrow \{(1, a), (1, bAA), (1, aS)\} \\ \delta(1, \epsilon, B) &\rightarrow \{(1, b), (1, bS), (1, aBB)\} \\ \delta(1, a, a) &\rightarrow \{(1, \epsilon)\} \\ \delta(1, b, b) &\rightarrow \{(1, \epsilon)\} \end{aligned}$$

All other transitions are empty.■

2. Problem 6.3.3: Convert the PDA $P = (\{p, q\}, \{0, 1\}, \{X, Z_0\}, \delta, q, Z_0)$to a CFG, if $\delta$is given by

$$\begin{aligned}
\delta(q, 1, Z_0) &= \{(q, XZ_0)\} \\
\delta(q, 1, X) &= \{(q, XX)\} \\
\delta(q, 0, X) &= \{(p, X)\} \\
\delta(q, \epsilon, X) &= \{(q, \epsilon)\} \\
\delta(p, 1, X) &= \{(p, \epsilon)\} \\
\delta(p, 0, Z_0) &= \{(q, Z_0)\}
\end{aligned}$$

Solution: we seek to construct $G = (V, \Sigma, R, S)$where $L(G) = N(P)$as seen on page 242 of the text. The set of variables, V, consists of (1) the start symbol S and (2) all symbols of the form $[pXq]$where $p$and $q$are states in Q and X is a stack symbol in Γ. Thus, our variables are as follows

$$S, [qXq], [qZ_0q], [qXp], [qZ_0p], [pXq], [pZ_0q], [pXp], [pZ_0p]$$

This is simply all combinations of states and stack symbols

Now let's create the productions. First we will start with the special start production. We know $q$is our start symbol and we could go to either p or stay at q from S. So, we have

$$S \to [qZ_0p] \mid [qZ_0q]$$

Now let us take care of the first line of the transition function. We know if $\delta(q, 1, Z_0)$ contains (q, XZ_0)then we have, from page 242, the productions

$$\begin{aligned}
[qXq] &\to 1[qXq][qZ_0q] \mid 1[qXp][pZ_0p] \\
[qXp] &\to 1[qXq][qZ_0p] \mid 1[qXp][pZ_0p]
\end{aligned}$$

Similarly, for the second line of δ,we have

$$\begin{aligned}
[qXq] &\to 1[qXq][qXq] \mid 1[qXp][pXq] \\
[qXp] &\to 1[qXq][qXp] \mid 1[qXp][pXp]
\end{aligned}$$

For line three:

$$\begin{aligned}
[qXq] &\to 0[pXq] \\
[qXp] &\to 0[pXp]
\end{aligned}$$

For line four:

$$[qXq] \rightarrow \epsilon$$

For line five:

$$[pXp] \rightarrow 1$$

And finally, line six:

$$\begin{aligned} [pZ_0q] &\rightarrow 0[qZ_0q] \\ [pZ_0p] &\rightarrow 0[qZ_0q] \end{aligned}$$

■

3. Problem 7.1.5
Eliminate all ϵproductions, unit productions, useless symbols, and then put into Chomsky normal form for this grammar:

$$\begin{aligned} S &\rightarrow aAa \mid bBb \mid \epsilon \\ A &\rightarrow C \mid a \\ B &\rightarrow C \mid b \\ C &\rightarrow CDE \mid \epsilon \\ D &\rightarrow A \mid B \mid ab \end{aligned}$$

(a) remove all ϵtransitions:
To do this, we look at all productions and mark nullable symbols. Clearly, C is nullable, as it has a direct transition to ϵ.Also, notice that $A$and B are nullable because they have a unit transition to a nullable symbol. Now, we must eliminate all ϵtransitions and add some productions to take care of the cases where we would follow the ϵtransition. In the end, we end up with:

$$\begin{aligned} S &\rightarrow aAa \mid bBb \mid aa \mid bb \\ A &\rightarrow C \mid a \\ B &\rightarrow C \mid b \\ C &\rightarrow CDE \mid DE \\ D &\rightarrow A \mid B \mid ab \end{aligned}$$

(b) Remove all unit productions
Here we have 3 unit productions. Simply replace them by the entire production:

$$
\begin{aligned}
S &\rightarrow aAa \mid bBb \mid aa \mid bb \\
A &\rightarrow CDE \mid DE \mid a \\
B &\rightarrow CDE \mid DE \mid b \\
C &\rightarrow CDE \mid DE \\
D &\rightarrow CDE \mid DE \mid a \mid b \mid ab
\end{aligned}
$$

(c) Eliminate useless symbols:
$S, A$and B all derive terminal strings. E does not exist, so we are free to eliminate all productions that have E in the body. Notice that D only occurred in productions with E, so D can be removed as well:

$$
\begin{aligned}
S &\rightarrow aAa \mid bBb \mid aa \mid bb \\
A &\rightarrow a \\
B &\rightarrow b
\end{aligned}
$$

(d) Put into CNF.
Chosky Normal Form requires every production to be of the form $A \rightarrow a$ or $A \rightarrow BC$. This is easy for the above grammar, as we can replace aa with AA and bb with BB, as well as ab with AB. The only trouble comes with S where we have two terminals and a variable. So, we create a few combination variables, say $[aA] \rightarrow AA$ and $[bB] \rightarrow BB$. This leads our grammar to be:

$$
\begin{aligned}
S &\rightarrow [aA]A \mid [bB]B \mid AA \mid BB \\
A &\rightarrow a \\
B &\rightarrow b \\
[aA] &\rightarrow AA \\
[bB] &\rightarrow BB
\end{aligned}
$$

Thus, this gives our grammar in Chomsky Normal Form. Just to be safe, notice that $\epsilon$was in our original grammar, so our CNF grammar is actually for the language $L - \epsilon$.■

4. Problem 7.2.1
Use the CFL pumping lemma to show each of these languages are not context free.
Solution: for all problems, Let n be the integer of the pumping lemma. Then, for every string $z \in L$, $|a| \leq n$, z can be written as $z = uvwxy$, $|vwx| \leq$

n, $|vx| \neq \epsilon$.
(a) $\{a^i b^j c^k \mid i < j < k\}$
Solution: Take the string $a^n b^{n+1} c^{n+2}$. Then $vwx \in a^+$, a^+b^+, b^+, b^+c^+or c^+. It is easily seen that for the first 3 cases, pumping up will give you a string not in L. The last two cases can be pumped down to get a string not in L.
(b) $\{a^n b^n c^i \mid i \leq n\}$
Solution: Take the string $a^n b^n c^n$. We now know that for whatever vwx we choose, it cannot contain a,b, and c at the same time. Thus, for every way you can break the string up, pump up and there will be at least one letter that will appear less than the others, hence making it not in the language L.
(c) $\{0^p \mid p \; is \; a \; prime\}$
Solution: Let n be the integer of the pumping lemma. Consider some prime $p \geq n + 2$. Since there are an infinite number of primes, this p is guaranteed to exist. Let our string be 0^p. Let $|vwz| = m$. Then $|uy| = p - m$. Consider

$$u(vwx)^{p-m}y \quad = \quad |uy| + (p-m)(vwx) = p - m + (p-m)m = (m+1)(p-m)$$

So, it appears that the presence of the two factors $(m+1)$ and $(p-m)$ shows that we do not have a prime. But, we must ensure one of these factors is not 1. But $m + 1 > 1$ because $|vwx| \neq \epsilon$. Likewise, $p - m > 1$because $p \geq n + 2$ was chosen and $m \leq n$ since $m = |vwx| \leq |uy| \leq n$. So, $p - m > 2$. Thus this string is not in the language.
(d) $\{0^i 1^j \mid j = i^2\}$
Solution: Take our string to be $0^n 1^{n^2}$. No matter which way we break up the string, this equality will break. If we have $|vwx| = 0^+$then we can pump up once and we will have $0^{n+1}1^{n^2}$. We know that $(n+1)^2 \neq n^2$. The other case would be when $|vwx| = 0^+1^+$. But when we pump, we will have $0^{n+1}1^{n^2+1}$. We also know that $(n+1)^2 \neq n^2 + 1$. The last case is analagous to the first case.
(e) $\{a^n b^n c^i \mid n \leq i \leq 2n\}$
Solution: Take the string $a^n b^n c^{2n}$. Once again, we know we can't have a,b, and c all present in $|vwx|$ at the same time. Thus, we can pump up the parts featuring a^+, b^+, or a^+b^+at least n times (if that is how the string got broken up). Thus, i can be $2n$ for the worst case, and by pumping up more than n times we will have $i < n - 1$, which is not in the language. Similar pumping can be done for the other cases.
(f) $\{ww^R w \mid w \in (0+1)^*\}$
Solution: Take the string $0^n 110^n 0^n 1$. No matter how you break up the string, all three blocks of 0's will not be present in $|vwx|$. So, pumping up will give you a string with at least one block of 0's with more than the others. Hence, this is not in the language.
■

5. Problem 7.3.3. (b) and (c).
   Show that CFL's aren't closed under these operations
   (b) max
   Solution: take $L = \{a^i b^j c^k \mid i \leq j \text{ or } i \leq k\}$. We have shown this to be a CFL in a previous HW. Then $max(L) = \{a^i b^j c^k \mid k = max(i,j)\}$.
   Use the pumping lemma with the string $a^n b^n c^n$(which has been previously done) to show this is not a CFL. Hence, CFL's arent closed under max.
   (d) $half$
   Solution: Take $L = \{a^n b^n c^m d^{3m} \mid m, n > 0\}$. Clearly this is CFL. Now take $half(L) \cap a^* b^* c^+$and this will give you only the cases where $n < m$, (we are throwing out all the cases where $half$ returned a string with d^+on the end of it), which will only happen when $n > m$). $L' = \{a^n b^n c^j \mid j < n\}$. Clearly this is not CFL, and hence $half$ is not a closure property.
   ■

6. Problem 7.3.4 d and e
   (d) Show that if L is a CFL and R is a regular language then $shuffle(L, R)$ is a CFL.
   Proof by machine construction: Since R is a regular language, it has a DFA. We will construct a CFL to accept $shuffle(L, R)$ as follows:

$$\begin{aligned} Q &= Q_L \times Q_R \\ \Sigma &= \Sigma \\ \Gamma &= \Gamma \\ q_0 &= q_{0L} \times q_{0R} \\ Z_0 &= Z_{0L} \\ F &= F_L \times F_R \end{aligned}$$

   The transition function is easier to decribe in words: on each read of the input tape, the newly constructed machine guesses whether the input came from L or R, and transitions accordingly. ($\delta([p\ q],\ a,\ \beta)$ contains $\{\{[s\ q],\ \Theta\}, \{[p\ t],\ \beta\}\}$if $\delta_L(p, a, \beta)$ contains $\{(s, \Theta)\}$and $\delta_R(q, a) = t$. We will accept in this CFL if one of our branches ends in a final state for L and a final state for R. Since we can construct a PDA for $shuffle$, then it holds that it is a CFL.
   (e) show that if $L_1$and $L_2$are CFLs then $shuffle(L_1, L_2)$ need not be a CFL.
   Solution:
   Take $L_1 = \{a^n b^n \mid n > 0\}$ and $L_2 = \{c^m d^m \mid m > 0\}$. Now, $shuffle(L_1, L_2) \cap a^* c^* b^* d^*$ will give us all the strings of the form $\{a^n c^m b^n d^m \mid m, n > 0\}$,which is clearly not a CFL. Since we know that CFLs are closed under untion with a regular set and our set is clearly regular, it holds that CFLs

are not closed under $shuffle$.
■