CS3133
HW#7

DUE: Thursday, October 9

1. (20 points) a For the grammar $S \rightarrow 0S1|01$, derive an equivalent grammar $G$ in Chomsky Normal Form.
   b Describe $L(G)$ is English or set theoretic terms, and justify your answer.
   c Run the CKY algorithm to test if 000111 $\in L(G)$. Show all the sets $T_{ij}$. 1 $\leq i \leq j \leq 6$. If 000111 $\in L(G)$, show all parse trees in $G$.
   d Is $G$ ambiguous? Justify your answer. If $G$ is ambiguous, construct an unambiguous grammar equivalent to $G$.

2. (8 points) Is $\{a^ib^jc^kd^l \mid (0 \leq i,j,k,l) \land ((j = l) \lor (i \leq k) \lor (i + j = k + l))\}$ a CFL? Justify your answer.
1. \(a\) In order to apply the CKY algorithm, we first need to convert \(S \rightarrow 0S1|01\) to CNF. We first derive the equivalent rules:

\[
S \rightarrow A_0 A_1 \mid A_0 A_i \\
A_0 \rightarrow 0 \\
A_i \rightarrow 1
\]

We then process all productions (there is only one in our case) with a right-hand side of length greater than two:

\[
S \rightarrow A_0 C_1 \mid A_0 A_i \\
C_1 \rightarrow SA_i \\
A_0 \rightarrow 0 \\
A_i \rightarrow 1
\]

which is now in CNF.

\(b\) \(L(G) = \{0^n1^n | n \geq 1\}\). We prove this for the original grammar, \(S \rightarrow 0S1|01\).

Assume that \(z = 0^n1^n\) for some \(n \geq 1\). If \(n=1\), then by applying the production \(S \rightarrow 01\) we obtain the derivation \(S \Rightarrow 01\). If \(n>1\), then with \(n-1\) applications of the production \(S \rightarrow 0S1\) we obtain the derivation \(S \Rightarrow 0^{n-1}S1^{n-1}\), and then with one application of the production \(S \rightarrow 01\) we finish the derivation \(S \Rightarrow 0^n1^n\). So \(0^n1^n \in L(G)\).

Only one derivation in \(G\) starts with an application of the production \(S \rightarrow 01\), and that is the derivation \(S \Rightarrow 01\), and \(01 \in \{0^n1^n | n \geq 1\}\). All other derivations in \(G\) start with some number, say \(m \geq 1\), of applications of the production \(S \rightarrow 0S1\). Thus they start with \(S \Rightarrow 0^mS1^m\). In order to derive a string of the language \(L(G)\), they must then ultimately invoke one application of the production \(S \rightarrow 01\), yielding \(S \Rightarrow 0^nS1^n \Rightarrow 0^{n+1}1^{m+1}\), and \(0^{m+1}1^{m+1} \in \{0^n1^n | n \geq 1\}\).

\(c\) Now we apply the CKY algorithm to test if \(0001111 \in L(G)\):

\[
\begin{align*}
T_{1,1} &= T_{2,2} = T_{3,3} = \{A_0\} \\
T_{4,4} &= T_{5,5} = T_{6,6} = \{A_1\} \\
T_{1,2} &= T_{2,3} = \emptyset, \\
T_{3,4} &= \{S\}, \\
T_{4,5} &= T_{5,6} = \emptyset \\
T_{1,3} &= T_{2,4} = \emptyset, \\
T_{3,5} &= \{C_1\}, \\
T_{4,6} &= \emptyset \\
T_{1,4} &= \emptyset, \\
T_{2,5} &= \{S\}, \\
T_{3,6} &= \emptyset \\
T_{1,5} &= \emptyset, \\
T_{2,6} &= \{C_1\} \\
T_{1,6} &= \{S\}
\end{align*}
\]
A parse tree of 000111 is:  

```
S / \  
A_0 C_1  
| / \  
0 S A_1  
/ \ 1  
A_0 C_1 1  
| / \  
0 S A_1  
/ \ 1  
A_0 A_1 1  
1 1  
0 1
```

So 000111 does indeed belong to \( L(G) \).

\( d \) \( G \) is unambiguous. We prove this by induction on the number of applications of the production \( S \rightarrow A_0 C_1 \) to derive a string in \( L(G) \).

If there are no applications of the production \( S \rightarrow A_0 C_1 \), then the only derivation starts with an application of the production \( S \rightarrow A_0 A_1 \) yields the unique parse tree

```
S / \  
A_0 A_1  
| 1  
A_0 A_1 1  
1 1  
0 1
```

Assume that for some fixed \( k \geq 0 \), all derivations of strings in \( L(G) \) with \( k \) applications of the production \( S \rightarrow A_0 A_1 \) are unambiguous. \textbf{Any} derivation with more than \( k \) applications of the production yields the "head" of any parse tree
to derive some string $0w1 \in \{0^n1^m|n \geq 1\}$, $w \in \{0,1\}^*$. But the derivation $S \Rightarrow w$ must use fewer than $k$ applications of the production $S \Rightarrow A_0C_1$. By the induction hypothesis, this derivation can admit only one parse tree under the lower $S$ above. So $G$ is unambiguous.

2. We show that $L = \{a^ib^jc^kd^l|0 \leq i,j,k,l \land (j = l) \lor (i \leq k) \lor (i + j = k + l)\}$ is a CFL by first showing that $L_0 = \{a^ib^jc^kd^l|0 \leq i,j,k,l \land (j = l)\}$, $L_1 = \{a^ib^jc^kd^l|0 \leq i,j,k,l \land (i \leq k)\}$, and $L_2 = \{a^ib^jc^kd^l|0 \leq i,j,k,l \land (i + j = k + l)\}$ are all CFLs:

$L_0$ is generated by

$$
S \rightarrow AB
$$

$$
A \rightarrow aA | \epsilon
$$

$$
B \rightarrow bBd | C
$$

$$
C \rightarrow cC | \epsilon
$$

$L_1$ is generated by

$$
S \rightarrow AD
$$

$$
A \rightarrow aAc | BC
$$

$$
B \rightarrow bBe | \epsilon
$$

$$
C \rightarrow cC | \epsilon
$$

$$
D \rightarrow dD | \epsilon
$$

$L_2$ is generated by

$$
S \rightarrow aSd | A | C
$$

$$
A \rightarrow bAd | B
$$

$$
B \rightarrow bBe | \epsilon
$$

$$
C \rightarrow aCc | B
$$

Since $L = L_0 \cup L_1 \cup L_2$, we can just appeal to the fact that CFLs are closed under union to prove that $L$ is context-free.