DUE: Thursday, October 10

1. (7 points) Prove that every regular language is a context-free language.

2. (12 points) For each of the following CFGs, do each of the following:
   • Describe the language that it generates either with a regular expression or in set theoretic terms.
   • Tell whether the grammar is ambiguous. Justify your answer.
   • If the grammar is ambiguous, give an unambiguous grammar to generate the same language.

   a
   S \rightarrow 00S \cup 000S \cup \varepsilon

   b
   S \rightarrow 0SA \cup \varepsilon
   A \rightarrow 1AA \cup \varepsilon

   c
   S \rightarrow 0A \cup \varepsilon
   A \rightarrow 0A \cup 1B
   B \rightarrow 1B \cup 1

3. (8 points) Run the CKY algorithm on the grammar G
   S \rightarrow AB | a
   A \rightarrow a
   B \rightarrow AB \cup SA | b

   and the input string z = aaba. Show all the sets $T_{ij}$. If z belongs to $L(G)$, then show a derivation tree.

4. (5 points) Answer the following, and justify your answer.

   Conjecture: For any CFG $G_0 = (N, \Sigma, P, S)$, if $G_1 = (N, \Sigma, P \cup \{S \rightarrow SS \cup \varepsilon\}, S)$, then $L(G_1) = (L(G_0))^*$. 
1. If \( L \) is regular, then there exists a regular expression \( \alpha \) such that \( L(\alpha) = L \). We derive a CFG \( G \) from \( \alpha \) such that \( L(G) = L(\alpha) = L \), we invoke the following program.

\[
\text{CFG}(\alpha)
\]

if \( \alpha = \emptyset \) then return \( G = (\{S\}, \Sigma, \emptyset, S) \)

if \( \alpha = \epsilon \) then return \( G = (\{S\}, \Sigma, \{S \rightarrow \epsilon \}, S) \)

if \( \alpha = a \) then return \( G = (\{S\}, \Sigma, \{S \rightarrow a \}, S) \)

if \( \alpha = \beta + \gamma \) then
  
  invoke CFG(\( \beta \)) to yield \( G_\beta = (N_\beta, \Sigma, P_\beta, S_\beta) \)
  
  invoke CFG(\( \gamma \)) to yield \( G_\gamma = (N_\gamma, \Sigma, P_\gamma, S_\gamma) \)
  
return \( G = (N_\beta \cup N_\gamma \cup \{S\}, \Sigma, P_\beta \cup P_\gamma \cup \{S \rightarrow S_\beta \mid S_\gamma \}, S) \)

if \( \alpha = \beta \gamma \) then
  
  invoke CFG(\( \beta \)) to yield \( G_\beta = (N_\beta, \Sigma, P_\beta, S_\beta) \)
  
  invoke CFG(\( \gamma \)) to yield \( G_\gamma = (N_\gamma, \Sigma, P_\gamma, S_\gamma) \)
  
return \( G = (N_\beta \cup N_\gamma \cup \{S\}, \Sigma, P_\beta \cup P_\gamma \cup \{S \rightarrow S_\beta S_\gamma \}, S) \)

if \( \alpha = \beta^* \) then
  
  invoke CFG(\( \beta \)) to yield \( G_\beta = (N_\beta, \Sigma, P_\beta, S_\beta) \)
  
return \( G = (N_\beta \cup \{S\}, \Sigma, P_\beta \cup \{S \rightarrow \epsilon \mid SS_\beta \}, S) \)

2. \( \{0^n 1 (\exists k, l \in \mathbb{N}) n = 2k + 3l \} \)

The grammar is ambiguous because there exist two parse trees for 000000. They follow from the two leftmost derivations:

\( S \rightarrow 00S \rightarrow 000S \rightarrow 00000S \rightarrow 000000 \) and \( S \rightarrow 00S \rightarrow 00000S \rightarrow 000000 \)

The grammar can be made unambiguous by first generating pairs of 0's, then triples of 0's.

\[
S \rightarrow 00S1A \\
A \rightarrow 00A1\epsilon
\]

\( b \ 0^+1^+ \)

The grammar is ambiguous because there exist two parse trees for 011. They follow the two leftmost derivations which start with \( S \rightarrow 0SA \rightarrow 0A \rightarrow 01AA \). We get two distinct leftmost derivations of 011 by first applying \( A \rightarrow \epsilon \) to the first \( A \) and then applying \( A \rightarrow 1 \) to the remaining \( A \), or by first applying \( A \rightarrow 1 \) on the left \( A \) and then applying \( A \rightarrow \epsilon \) on the remaining \( A \).

An unambiguous grammar for the same language is
The grammar is unambiguous. The only one way to generate $\varepsilon$ is to use the production $S \rightarrow \varepsilon$. Otherwise the string is of the form $0^m1^n$, $m \geq 1, n \geq 2$. The only derivation consists of one application of $S \rightarrow 0A$, followed by $m-1$ applications of $A \rightarrow 0A$, followed by $A \rightarrow 1B$, followed by $n-2$ applications of $B \rightarrow 1B$, followed by one application of $B \rightarrow 1$.

3. $T_{1,1} = \{S, A\}$  $T_{1,2} = \{S, A\}$  $T_{2,2} = \{B\}$  $T_{3,3} = \{S, A\}$  $T_{4,4} = \{S, A\}$  $T_{3,4} = \emptyset$  $T_{1,3} = \{S, B\}$  $T_{2,3} = \{S, B\}$  $T_{3,4} = \{B\}$  $T_{4,4} = \{S, B\}$

A derivation tree is:

```
      S
     / \  \
    A   B
   /   \  \
  a   S   A
 /  \  |  |
A   B  a
|  |  |
A  b
```

4. The conjecture is false. Let $G_0$ be the grammar

- $S \rightarrow A$
- $A \rightarrow 0S$

$L(G_0) = \emptyset$. But adding the two new productions to get $G_1$

- $S \rightarrow A | SS | \varepsilon$
- $A \rightarrow 0S$

we get $L(G_0) = 0^*$. 