Ryan Darras, CS 5070 - HW05 2.(2, 4(b,c,e,f), 9, 14, 15, 27)

Problem 2.2

a) Use the languages A = $\{a^m b^n c^n \mid m,n \geq 0\}$ and B = $\{a^n b^n c^m \mid m,n \geq 0\}$ together with Example 2.36 to show that the class of context-free languages is not closed under intersection.
b) Use part (a) and DeMorgan's law (Theorem 0.20) to show that the class of context-free languages is not closed under complementation.

Problem 2.2a Answer

Languages A and B have CFG's G_a and G_b such that

G_a = S → XY, X → aX | ε, Y → bYc | ε
G_b = S → XY, X → aXb | ε, Y → cY | ε

The intersection A ∩ B = $\{ a^n b^n c^n \mid n \geq 0 \}$ is not context free as shown in example 2.36 in the book. Hence, the class of context-free languages is not closed under intersection.

Problem 2.2b Answer

Proof by Contradiction:
Assume that the class of context-free languages is closed under complementation. Take A and B from part a, because we are assuming that the class is closed under complementation we can say that $\overline{A}$ and $\overline{B}$ are context-free. Therefore $\overline{A} \cup \overline{B}$ is context-free. Again, assuming that a class is closed under complementation we can say that $\overline{\overline{A} \cup \overline{B}}$ is context-free. DeMorgan's law says that $\overline{A \cup B} = \overline{A} \cap \overline{B}$, from which we can derive A ∩ B = $\overline{\overline{A} \cup \overline{B}}$. From part a we discovered that A and B are not closed under intersection, hence we have a contradiction which means our initial assumption must be untrue. Therefore, the class of context-free languages is not closed under complementation.

Problem 2.4(b,c,e,f)

Give context-free grammars that generate the following languages. In all parts, the alphabet Σ is {0,1}.

b) {w| w starts and ends with the same symbol}

c) {w| the length of w is odd}

e) {w| w = wR, that is,w is a palindrome}

f) The empty set

Problem 2.4b Answer

$G_b = S \rightarrow X$ $X \rightarrow 0Y0 \mid 1Y1$ $Y \rightarrow 0Y \mid 1Y \mid \varepsilon$

Problem 2.4c Answer

$G_c = S \rightarrow XY$ $X \rightarrow 00X \mid 01X \mid 10X \mid 11X \mid \varepsilon$ $Y \rightarrow 0 \mid 1$

Problem 2.4e Answer

$G_e = S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon$ (shown in class)

Problem 2.4f Answer

$G_f = S \rightarrow \varepsilon$

Problem 2.9

Give a context-free grammar that generates the language

$A = \{a^i b^j c^k \mid i = j \text{ or } j = k \text{ where } i,j,k \geq 0\}$.

Is your grammar ambiguous? Why or why not?

Problem 2.9a Answer

$G_a = S \rightarrow XY \mid AB$ $X \rightarrow aXb \mid \varepsilon$ $Y \rightarrow cY \mid \varepsilon$ $A \rightarrow aA \mid \varepsilon$ $B \rightarrow bBc \mid \varepsilon$

This grammar is unambiguous because for each output there is only one parse tree.

Problem 2.14

Convert the following CFG into an equivalent CFG in Chomsky normal form, using the procedure given in Theorem 2.9.

$A \rightarrow BAB \mid B \mid \varepsilon$
$B \rightarrow 00 \mid \varepsilon$

Problem 2.14 Answer

$S_0 \rightarrow A \mid \varepsilon$
$A \rightarrow A_1B \mid U_0U_0 \mid AB \mid BA$
$A_1 \rightarrow BA$
$B \rightarrow U_0U_0$
$U_0 \rightarrow 0$

Problem 2.15

Give a counterexample to show that the following construction fails to prove that the class of context-free languages is closed under star. Let A be a CFL that is generated by the CFG $G = (V,\Sigma,R,S)$. Add the new rule $S \rightarrow SS$ and call the resulting grammar G′. This grammar is supposed to generate A*.

Problem 2.15 Answer

Let the grammar G for CFL A be $S \rightarrow a \mid b$. If we add the new rule $S \rightarrow SS$ we get G' = $S \rightarrow a \mid b \mid SS$. This is missing a key component of what makes the star rule. That component is the fact that A* must contain ε, which in this counterexample it does not. Therefore by adding the rule $S \rightarrow SS$ to the grammar of the CFL, you do not transform the CFL into itself*.

Problem 2.27

Let $G = (V, \Sigma, R, \langle \text{STMT} \rangle)$ be the following grammar.

$$
\begin{aligned}
\langle \text{STMT} \rangle &\to \langle \text{ASSIGN} \rangle \mid \langle \text{IF-THEN} \rangle \mid \langle \text{IF-THEN-ELSE} \rangle \\
\langle \text{IF-THEN} \rangle &\to \texttt{if condition then } \langle \text{STMT} \rangle \\
\langle \text{IF-THEN-ELSE} \rangle &\to \texttt{if condition then } \langle \text{STMT} \rangle \texttt{ else } \langle \text{STMT} \rangle \\
\langle \text{ASSIGN} \rangle &\to \texttt{a:=1}
\end{aligned}
$$

$$
\begin{aligned}
\Sigma &= \{\texttt{if}, \texttt{condition}, \texttt{then}, \texttt{else}, \texttt{a:=1}\} \\
V &= \{\langle \text{STMT} \rangle, \langle \text{IF-THEN} \rangle, \langle \text{IF-THEN-ELSE} \rangle, \langle \text{ASSIGN} \rangle\}
\end{aligned}
$$

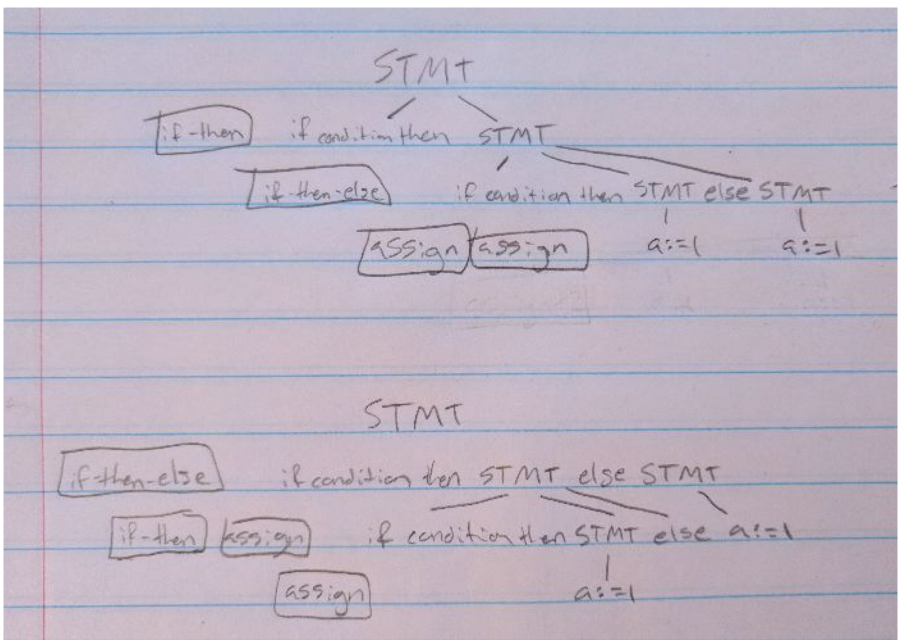
G is a natural-looking grammar for a fragment of a programming language, but G is ambiguous.

a. Show that G is ambiguous.

b. Give a new unambiguous grammar for the same language.

Problem 2.27a Answer

To show that G is ambiguous, all we need to do is show that 2 different parse trees can produce the same result.



Problem 2.27b Answer

<STMT> → <STMT><if-then> | <if-then>

<if-then> → if condition then <assign> | if condition then <assign> else <assign>

<assign → a:=1