CS 341 Automata Theory
Elaine Rich
Homework 7
Due: Tuesday, February 28

This assignment covers Chapter 11.

1) Let $\Sigma = \{a, b\}$. For the languages that are defined by each of the following grammars, do each of the following:

   i. List five strings that are in L.

   ii. List five strings that are not in L.

   iii. Describe L concisely. You can use regular expressions, expressions using variables (e.g., $a^n b^n$, or set theoretic expressions (e.g., $\{x : \ldots\}$)

   iv. Indicate whether or not L is regular. Prove your answer.

   a) $S \rightarrow \mathtt{a}S \mid S\mathtt{b} \mid \epsilon$

   b) $S \rightarrow \mathtt{a}S\mathtt{a} \mid \mathtt{b}S\mathtt{b} \mid a \mid b$

   c) * $S \rightarrow \mathtt{a}S \mid \mathtt{b}S \mid \epsilon$

   d) * $S \rightarrow \mathtt{a}S \mid \mathtt{a}S\mathtt{b}S \mid \epsilon$

   a) b) * 00101101

2) Consider the following grammar $G : S \rightarrow \mathtt{0}S\mathtt{1} \mid SS \mid \mathtt{10}$. Show a parse tree produced by G for each of the following strings:

   a) `010110`

   b) * `00101101`

3) Let G be the grammar of Example 11.12. Show a third parse tree that G can produce for the string (())().

4) Consider the following context free grammar G:

$$
\begin{aligned}
S &\rightarrow \mathtt{a}S\mathtt{a} \\
S &\rightarrow T \\
S &\rightarrow \epsilon \\
T &\rightarrow \mathtt{b}T \\
T &\rightarrow \mathtt{c}T \\
T &\rightarrow \epsilon
\end{aligned}
$$

   One of these rules is redundant and could be removed without altering $L(G)$. Which one?

5) Using the simple English grammar that we showed in Example 11.6, show two parse trees for each of the following sentences. In each case, indicate which parse tree almost certainly corresponds to the intended meaning of the sentence:

   a) `The bear shot Fluffy with the rifle.`

   b) `Fluffy likes the girl with the chocolate.`

6) Show a context-free grammar for each of the following languages L:

   a) $\{a^i b^j \ : \ 2i = 3j + 1\}$.
   b) * $\{w \in \{a, b\}^* \ : \ \#_a(w) = 2\#_b(w)\}$.
   c) $\{w \in \{a, b\}^* \ : \ w = w^R\}$.
   d) $\{w \in \{a, b\}^*$: every prefix of w has at least as many as as bs$\}$.
   e) $\{a^m b^n c^p d^q \ : \ m, n, p, q \geq 0$ and $m + n = p + q\}$.
   f) $\{b_i \# b_{i+1}^R \ : \ b_i$ is the binary representation of some integer i, $i \geq 0$, without leading zeros$\}$. (For example $101\#011 \in L$.)

7) * Let G be the ambiguous expression grammar of Example 11.14. Show at least three different parse trees that can be generated from G for the string `id+id*id*id`.

8) Consider the unambiguous expression grammar G of Example 11.19.

   a) Trace the derivation of the string `id+id*id*id` in G.
   b) Add exponentiation ($**$) and unary minus ($-$) to G, assigning the highest precedence to unary minus, followed by exponentiation, multiplication, and addition, in that order.

9) Let $L = \{w \in \{a, b, \cup, \epsilon, (,), *, +\}^* : w$ is a syntactically legal regular expression$\}$.

   a) Write an unambiguous context-free grammar that generates L. Your grammar should have a structure similar to the arithmetic expression grammar G that we presented in Example 11.19. It should create parse trees that:

      - Associate left given operators of equal precedence, and
      - Correspond to assigning the following precedence levels to the operators (from highest to lowest):
        1) $*$ and $+$
        2) concatenation
        3) $\cup$

   b) Show the parse tree that your grammar will produce for the string $(a \cup b)ba^*$.

10) * In Appendix I.3.1, we present a simplified grammar for URIs (Uniform Resource Identifiers), the names that we use to refer to objects on the Web.

    a) Using that grammar, show a parse tree for:
       `https://www.mystuff.wow/widgets/fradgit#sword`
    b) Write a regular expression that is equivalent to the grammar that we present.

11) Consider the grammar $G = (\{S, A, B, T, a, c\}, \{a, c\}, R, S)$, where $R = \{S \to AB,\ S \to BA,\ A \to aA,\ A \to ac,\ B \to Tc,\ T \to aT,\ T \to a\}$.

    a) Show that G is ambiguous.
    b) Find an equivalent grammar that is not ambiguous.

12) Consider the fragment of a Java grammar that is presented in Example 11.20. How could it be changed to force `each` else clause to be attached to the outermost possible `if` statement?

13) * How does the `COND` form in Lisp, as described in Section G.5, avoid the dangling else problem?