Homework 3, Theory of Computation

1. [3, Problem 2.19] Show that, if G is a CFG in Chomsky normal form, then for any $w \in \mathcal{L}(G)$ of length $n \geq 1$, exactly $2n - 1$ steps are required for any derivation of w.

2. [3, Problem 2.20] Let G be a CFG in Chomsky normal form that contains b non-terminals. Show that, if G generates some string using a derivation with at least 2^b steps, then $\mathcal{L}(G)$ is infinite.

3. [2, Homework 6, problem 3(b)] Let $b(n)$ denote the binary representation of $n \geq 1$, leading zeros omitted. Show that the set of strings $\{b(n)\$b(n+1)^r\}$ is a CFL over the alphabet $\{0, 1, \$\}$. ($b(n)^r$ is the reverse of $b(n)$).

4. [1, Exercise 5.1.1 (d)] Show that the set of all strings over $\{0, 1\}$ with twice as many 0s as 1s is a CFL.

5. [1, Exercise 7.3.4] The *shuffle* of two strings w_1 and w_2 is the set of all strings that can be obtained by interleaving the positions of w_1 and w_2 in any way. More precisely, *shuffle*(w_1, w_2) is the set of strings w_3 such that

   (a) each position of w_3 can be assigned to w_1 or w_2, but not both,

   (b) the positions of w_3 assigned to w_1 form w_1 when read from left to right and

   (c) the positions of w_3 assigned to w_2 form w_2 when read from left to right.

   For example, if $w_1 = 01$ and $w_2 = 110$, then *shuffle*$(01, 110)$ is the set of strings $\{01110, 01101, 10110, 10101, 11010, 11001\}$. To illustrate the necessary reasoning, the fourth string, 10101, is justified by assigning the second and fifth positions to 01 and positions one, three and four to 110. We can also define shuffle of languages as *shuffle*$(L_1, L_2) = \{$*shuffle*$(w_1, w_2) \mid w_1 \in L_1, w_2 \in L_2\}$.

   (a) Show that if L_1 and L_2 are regular, so is *shuffle*(L_1, L_2).

   (b) Show that if L_1 is a CFL and L_2 is regular, then *shuffle*(L_1, L_2) is a CFL.

6. [1, Exercise 7.4.5] Modify the CYK algorithm to report the number of distinct parse trees for the given input, rather than just reporting membership in the language.

References

[1] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation (Third Edition)*. Pearson Education, 2007.

[2] Dexter C. Kozen. *Automata and Computability*. Springer-Verlag, 1997.

[3] Michael Sipser. *Introduction to the Theory of Computation*. Thomson Brooks/Cole, 1997.