COMS 331: Theory of Computation
Summer 2023

Homework Assignment 3

Due: 11:59PM, June 9 (Friday).

Problem 1 (90 points)

1. Give an English description of the language generated by the grammar $G = (\{S, A\}, \{a, b\}, S, P)$, where the productions P are:
$$S \to aA, A \to abS, S \to \epsilon.$$
2. Give an English description of the language generated by the grammar $G = (\{S, A, B\}, \{a, b\}, S, P)$, where the rules P are:
$$S \to b, S \to Aa, A \to B, B \to Aa.$$
3. Give an English description of the language generated by the grammar $G = (\{S, A, B\}, \{a\}, S, P)$, where the rules P are:
$$S \to Aa, A \to B, B \to Aa.$$
4. Give the derivation of string $aabab$ using the grammar $G = (\{S, Y\}, \{a, b\}, S, P)$, where P:
$$P : S \to aSb|aY|bY, Y \to aY|bY|a|b$$
5. Give the parse tree of string $abbab$ using the grammar $G = (\{S, Y\}, \{a, b\}, S, P)$, where P:
$$P : S \to aSb|aY|bY, Y \to aY|bY|a|b$$

Problem 2 (20 points)

Let the context-free grammar $G = (\{S, Y\}, \{a, b\}, S, P)$, where P:
$$S \to YS \mid \epsilon, Y \to ab \mid aYb \mid Yb$$
1) Prove G is ambiguous.
2) What is the language generated by G?

Problem 3 (20 points)

Define a context-free grammar for the language

$$L = \{a^n b^m c^{n-m} \mid n \geq m\}$$

.

Problem 4 (20 points)

Consider the alphabet $\Sigma = \{a, b, (,), +, *, \emptyset\}$. Construct a CFG G that generates all strings in Σ^* that are (correctly parenthesized) regular expressions over $\{a, b\}$.

Problem 5 (25 points)

Define a context-free grammar for the language

$$L = \{xyzy^R x^R \mid x \in \{a, b\}^*, y \in \{c, d\}^*, z \in \{a\}^*\}$$

Problem 6 (25 points)

The truth value of a logical expression is defined recursively as:

- The truth value of t is t.
- The truth value of f is f.
- The truth value of $(x_1 \wedge x_2)$ is t if both x_1 and x_2 have truth value t, it is f otherwise.
- The truth value of $(x1 \vee x2)$ is f if both x_1 and x_2 have truth value f, it is t otherwise.
- The truth value of $\neg(x)$ is f if x has truth value t, it is t otherwise.

Define a CFG that generates the following language over $\{t, f, \wedge, \vee, \neg, (,), =\}$:

$$L = \{w = x \mid w \text{ is a logical expression over } \{t, f\},\ x \in \{t, f\}, \text{ and } x \text{ is the truth value of } w\}$$

Thus, "$t = t$", "$((t \wedge f) \vee f) = f$", and "$\neg(((t \wedge f) \vee f)) = t$" are in L, but "$((t \wedge f) \vee f) = t$" and "$(t \wedge f) \vee f = f$" are not: the former because $((t \wedge f) \vee f)$ is false and not true, the latter because the expression lacks the outermost set of parentheses.