CS320: Problems for Day 12, Winter 2023

Problem 1 Let L be the language defined by the regular expression:

$$(\boldsymbol{bb} \cup \boldsymbol{cc})^*((\boldsymbol{a} \cup \boldsymbol{ba})\boldsymbol{cd})^*$$

(a) Write a complete formal definition of a context-free grammar that generates L. If such grammar does not exist, prove it.

(b) Is it possible to write a computer program (algorithm) that operates as follows:

INPUT: An arbitrary Turing machine ζ accepting strings over $\{a, b, c, d\}$.

OUTPUT: **yes** if ζ accepts L and **no** if ζ does not accept L.

Explain your answer briefly.

Problem 2 Does there exist an algorithm that solves the following problem:

INPUT: An arbitrary context-free grammar G and an arbitrary compiler $\mathcal{P}$.

QUESTION: Is the language specified by G equal to the set of programs that compile under $\mathcal{P}$?

Explain your answer briefly.

Problem 3 Consider the Turing machine:

$$M = (Q, \Sigma, \Gamma, \delta, p)$$

such that: $Q = \{p, q, s, t, e\}$; $\Sigma = \{a, b, c\}$;
$\Gamma = \{B, X, Y, Z, a, b, c\}$.
and δ is defined by the following transition set:

$$\begin{array}{l}
[p, a, q, X, R] \\
[p, b, q, Y, R] \\
[p, c, q, Z, R] \\
\\
[q, a, q, a, R] \\
[q, b, q, b, R] \\
[q, c, q, c, R] \\
[q, B, s, B, L] \\
\\
[s, X, t, B, R] \\
[s, Y, t, B, R] \\
[s, Z, t, B, R] \\
[s, a, e, a, R] \\
[s, b, e, b, R] \\
[s, c, e, c, R] \\
\\
[e, B, e, B, R]
\end{array}$$

(where B is the designated blank symbol.)

Let L be the set of strings accepted by M (by halting.)

(a) Draw a state-transition graph of a finite automaton M' that accepts L. If such an automaton does not exist, prove it.

(b) Write a complete formal definition of a Turing machine M_1 that accepts the language L and halts on every input. In short:

$$(\tau \in L) \implies (M_1(\tau) \searrow \text{ and accept })$$

and also:

$$(\tau \notin L) \implies (M_1(\tau) \searrow \text{ and reject })$$

Your construction should be readable as well as accurate; you may comment it. If such a Turing machine does not exist, prove it.

(c) Write a complete formal definition of a Turing machine M_2 that halts on every input and recognizes Turing machines that accept L. In short:

$$(L(\eta) = L) \implies (M_2(\eta) \searrow \text{ and accept })$$

and also:

$$(L(\eta) \neq L) \implies (M_2(\eta) \searrow \text{ and reject })$$

Your construction should be readable as well as accurate; you may comment it. If such a Turing machine does not exist, prove it.