CS 331: Theory of Computing

Problem Set 5

Iowa State University
Computer Science Department
February 28, 2013

Due by the midnight of March 7, 2013

Problem 1 (20 points)

Let $\Sigma = \{0, 1\}$. Consider the following language over Σ:

$$\mathscr{L} = \{ w \in \Sigma^* \mid \textit{ w contains twice as many } 0\textit{'s as } 1\textit{'s } \}.$$

- (10 points) Describe in pseudo-code (as in Example 3.7 in the textbook) of a Turing machine $\mathcal{M}$ that recognizes $\mathscr{L}$.
- (10 points) Formally define $\mathcal{M}$. You may write down a formal definition or draw a diagram like Figure 3.8 in the textbook. You may choose Γ freely.

Problem 2 (20 points)

Let $\Sigma = \{0, 1\}$ and $\Gamma = \{0, 1, \sqcup, \#, x\}$ (where $\sqcup$ denotes white space). Write a Turing machine $\mathcal{M}$ such that given any input $w \in \Sigma^*$, $\mathcal{M}$ halts with tape content $w\#w^R$.

- (10 points) Describe $\mathcal{M}$ in pseudo-code.
- (10 points) Formally define $\mathcal{M}$. You may write down a formal definition or draw a diagram like Figure 3.8.

Note: we do not care if $\mathcal{M}$ halts in accepting state or rejecting state.

Problem 3 (20 points)

Let $\mathcal{M} = \langle \Sigma, \Gamma, Q, q_0, \delta, q_{accept}, q_{reject} \rangle$ be a Turing machine where $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, \square\}$ ($\square$ denotes white space), $Q = \{q_0, q_1, q_2, q_3, q_{accept}, q_{reject}\}$, and δ is represented by the following table.

δ	0	1	$\square$
q_0	$(q_0, 0, R)$	$(q_0, 1, R)$	$(q_1, \square, L)$
q_1	$(q_2, 1, L)$	$(q_1, 0, L)$	$(q_3, 1, L)$
q_2	$(q_2, 0, L)$	$(q_2, 1, L)$	$(q_{accept}, \square, R)$
q_3	–	–	$(q_{accept}, \square, R)$

- (10 points) Show the sequence of computation of $\mathcal{M}$ when given input 01101.
- (10 points) What is the functionality of $\mathcal{M}$. Justify your answer.

Problem 4 (20 points)

Let $\mathscr{L}$ be a language over $\Sigma = \{0, 1\}$. We order finite words in $\mathscr{L}$ by length, and for words of the same length, we order them lexicographically. Prove that $\mathscr{L}$ is Turing-decidable if and only if $\mathscr{L}$ can be enumerated by an enumerator Turing machine in strictly increasing order.

Note: Your proof should consists of two parts, each of which is worth 10 points.

Problem 5 (20 points)

A 2-PDA is a 6-tuple $\langle Q, \Sigma, \Gamma, \delta, q_0, F \rangle$, where Q, Σ, Γ, and F are all finite sets, and

- Q is the set of states,
- Σ is the input alphabet,
- Γ is the stack alphabet,
- $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \times \Gamma_\epsilon \to \mathcal{P}(Q \times \Gamma_\epsilon \times \Gamma_\epsilon)$ is the transition function,
- $q_0 \in Q$ is the start state, and
- $F \subseteq Q$ is the set of accept states.

Basically, a 2-PDA is a PDA that operates on two tapes simultaneously. For example, $(q', b_1, b_2) \in \delta(q, \sigma, a_1, a_2)$ means the machine goes from q to q' when it reads letter σ, and the symbol a_1 (resp. a_2) is on the top of the stack 1 (resp. stack 2). And it replaces a_1 (resp. a_2) with b_1 (resp. b_2).

Prove that any Turing machine can be simulated by a 2-PDA.

Hint: Can a Turing machine configuration be represented by two stacks?