Middle East Technical University | Department of Computer Engineering

CENG 280

Formal Languages and Abstract Machines

Spring 2017-2018

Take Home Exam 3

Due date: May 26, 2018

Objectives

To familiarize with computation using Turing Machines and inspect aspects of language hierarchy covered throughout the course.

Specifications

You must adhere to the notation conventions adopted in the textbook. Use the standard deterministic Turing Machine as defined in the book unless stated otherwise. You should utilize basic machines notation for the fifth question.

Your solution should be delivered as a .tex file based on your modification of the provided template file.

Questions and submission regulations are included in subsequent sections. Designing your solutions to the tasks, explicitly state any assumptions you make and pay particular attention to the notation you use. Your proofs must be sound and complete. Grading will be heavily affected by the formalization of your solutions.

Question 1 (10 pts)

Given the context-free grammmar $G = (V, \Sigma, R, S)$ where $V = \{S, X, a, b, c\}$ and $\Sigma = \{a, b, c\}$ and $R = \{S \to aSXaX,\ S \to bSXbX,\ S \to c,\ X \to aX,\ X \to bX,\ X \to e\}$

a. Construct a bottom-up parser for G.

b. Trace $w = abbcbabbaa$ using yields-in-one-step relation on successive configurations of the parser you defined in **(a)**.

Question 2 (15 pts)

In each part, give a formal description for the Turing machines that computes the given function.

a. $f(x) = \begin{cases} x+1 & \text{if x is odd} \\ \frac{x}{2} & \text{if x is even} \end{cases}$, where $x \geq 1$.

b. $g(x, y) = |x - y|$, where $x, y \geq 1$. (ungraded)

Numbers will be given to TM's in unary. Initially, in part (a), tape is of the form ($\triangleright\sqcup$11...11$\sqcup\sqcup$...) where number of 1's equal to x, and in part (b), ($\triangleright\sqcup$11...11$\square$11...11$\sqcup\sqcup$...), where number of 1's before the square symbol represent x and number of 1's after the square symbol represent y. For both parts, initial position of the head is the blank symbol which is between left end symbol and the input. When the machines halt, tape will be of the form ($\triangleright\sqcup$11...11$\sqcup\sqcup$...) where number of 1's equal to the output of the function. Final position of the head does not matter.

For example, the TM in part (a) will be started with tape ($\triangleright\sqcup$111$\sqcup\sqcup$...) to calculate $f(3)$, and when the machine halts, the tape will have ($\triangleright\sqcup$1111$\sqcup\sqcup$...), which is the unary representation of 4. Similarly, for $f(4)$, it will be started with tape ($\triangleright\sqcup$1111$\sqcup\sqcup$...), and, when the machine halts, the tape will have ($\triangleright\sqcup$11$\sqcup\sqcup$...), which is the unary representation of 2. Note that the final position of the head is not specified here since it may differ according to your description.

In your description, represent the TM's as quintuples (K, Σ, δ, s, H) where K is the finite set of states; Σ is the alphabet containing blank symbol $\sqcup$, input separator symbol $\square$ (for part (b)), left end symbol $\triangleright$, and any other symbol you need to compute the functions, but not containing symbols $\to$ and $\leftarrow$; δ is the transition function from (K-H)$\times\Sigma$ to K$\times\Sigma\times\{\to, \leftarrow\}$; s is the start state and H is the set of halting states.

Question 3 (10 pts)

Move-restricted Turing Machines are like standard Turing Machines with semi-infinite tapes, but do not have the ability to move their head left, i.e. they may just read, write, and move towards the right. What is the set of languages decided by Move-restricted Turing Machines? Justify your answer.

Question 4 (25 pts)

Queue-based deterministic Turing Machines utilize a semi-infinite tape allowing only the following operations:

- *front:* access the first element of the input string on tape
- *rear:* access the last element of the input string on tape assuming that the rest of the tape is filled with blank symbols
- *enqueue:* append a new element to the end of the queue which becomes the rear element
- *dequeue:* remove the first element of the queue and update the front element accordingly.

Let queue-based TM have effectively two heads pointing to front and rear positions, and other than enqueue / dequeue operations it is not allowed to move heads left or right.

a. Formally define the queue-based TM using tuple notation. State the type of each constituent of the tuple.

b. Formally define the notion of configuration for the queue-based TM.

c. Assume that the front and rear heads initially point to the beginning and end of input string of the queue-based TM. Formally define the yields-in-one-step relation of the machine. Write in set notation the language recognized by the reflexive transitive closure of yields-in-one-step relation.

d. Prove that queue-based deterministic TM is equivalent to the standard TM.

e. Formally define a queue-based TM to decide the language $L = \{wcw : w \in \{a, b\}^*\}$.

Question 5 (20 pts)

Given the language $L = \{a^n b^{2^n} c^{3n} : n \in \mathbb{N}\}$ (Note that strings in L can be enumerated as $\{b, abbccc, aabbbbccccccc, aaabbbbbbbbccccccccc, ...\}$),

a. Use basic machines notation to provide a deterministic TM that decides L.

b. Formally define a grammar that generates only all the strings in L.

Question 6 (20 pts)

You are given the following information about formal languages L_1, L_2, L_3, L_4, and L_5, all defined over alphabet $\Sigma = \{a, b\}$:

- A regular grammar can generate all strings in L_1 and nothing else.
- A deterministic top-down parser can be built for context-free grammar whose language is L_2.
- Only an inherently ambiguous context-free grammar can be provided for L_3.
- There is a TM M that decides $\overline{L_4} \cap \overline{L(a^*b^*)}$.
- An unrestricted grammar can generate all strings in L_5.

Let $L = (\overline{L_2}L_1 \cap L_5)^* \cup (L_3^*\overline{L_4})$.

a. Do TM's M_1, M_2, M_2, M_3, M_4 and M_5 that accept L_1, L_2, L_3, L_4, and L_5 respectively exist?

b. Are there always algorithms for membership problems associated with each of L_1, L_2, L_3, L_4, and L_5?

c. Show precisely how to accept a string in L by using abstract machines constructed for L_1, L_2, L_3, L_4, and L_5.

d. Can we always come up with a TM that semidecides $\overline{L}$? Justify your answer.

Question 7 (ungraded)

Prove that L is not recursively enumerable where

$L = \{"M""w" : "M" \text{ is encoding of a Turing machine and } w \notin L(M)\}$.

Submission

- **Late Submission:** You have 2 days in total for late submission of all homeworks. All homeworks will be graded as normal during this period. No further late submissions are accepted.
- You should submit your THE1 as a .tex file. Please use the template provided on COW with appropriate modifications.
- Do not submit solutions for not-graded questions. Yet solving them is advisable in studying for the midterm.
- Soft-copies should be uploaded strictly by the deadline.

Regulations

1. **Cheating: We have zero tolerance policy for cheating**. People involved in cheating will be punished according to the university regulations.
2. **Newsgroup:** You must follow the newsgroup (news.ceng.metu.edu.tr) for discussions and possible updates on a daily basis.