Harvard University Extension School
Computer Science E-207

Problem Set 6

Due Friday November 2nd 2012 at 11:59 PM Eastern Time.
Submit your solutions in a single PDF called lastname+ps6.pdf emailed to cscie207@seas.harvard.edu.
LATE PROBLEM SETS WILL NOT BE ACCEPTED.

Problem set by **ENTER YOUR NAME HERE**

Collaboration Statement: **FILL IN YOUR COLLABORATION STATEMENT HERE (See the syllabus for information)**

See syllabus for collaboration policy.

PROBLEM 1 (8+4+6 points)

A function $f : \Sigma^* \to \Sigma^*$ is *computable* if there is a Turing machine M that, on every input $w \in \Sigma^*$, halts (in state q_{accept}) with the tape containing just the string $f(w)$ (followed by blank symbols). In this case, we say that M *computes* the function f.

(A) Draw a state diagram for a (one-tape, deterministic) Turing machine M that computes $f(a^n) = a^{\lceil \log_2(n) \rceil}$. Here, $\lceil x \rceil$ denotes the *ceiling* of x, i.e., the smallest integer y such that $x \le y$. Also include a prose description of how your Turing Machine works. Your state diagram should look similar to those on pages 144 in Sipser 2nd Edition. (Hint: $\lceil \log_2(n) \rceil$ is the length of the binary representation of n.)

(B) Using your Turing machine in (A), draw a state diagram for a (one-tape, deterministic) Turing machine that decides the language $\{a^n : \lceil \log_2(n) \rceil \text{ is even}\}$. You needn't redraw the entire state diagram, just the modifications to your machine in (A).

Note: For the remaining problems on the assignment, describe any TMs you construct at an *implementation level*, as in the proof of Theorem 3.13 (Multitape Turing Machines on page 149 in Sipser 2nd edition).

(C) Show that a function f over alphabet Σ is computable if and only if the language

$$L = \{w\$^i\sigma : w \in \Sigma^*, \sigma \in \Sigma, \text{ the } i\text{'th symbol of } f(w) \text{ is } \sigma\}$$

where \$ is a new symbol not in Σ, is decidable. (Thus, we can study computability of functions by studying decidability of languages.)

PROBLEM 2 (8 points)

Define a *RAM-TM* (Random Access Memory Turing Machine) as a Turing Machine with instant random access to some tape representing its memory. More specifically a RAM-TM has three tapes with the following properties:

1. A work tape that it can use like an ordinary TM. (You can think of this as containing "registers" or short term local memory for the TM.)

2. A "RAM" tape that represents the memory itself. This starts out with the input written (starting at location 0) and is blank elsewhere. The TM cannot use this tape directly, and instead reads and writes to this tape only through the RAM access tape described below

3. A RAM-access tape where the TM inputs its read and write "commands". To write to memory the TM writes a string of the form $nnnnW\sigma\$$ on this tape, where $nnnn$ is an integer in binary and σ is an alphabet symbol. Once this command is written the symbol σ is instantly written to location $nnnn$ on the RAM tape (tape 2). Similarly to read from memory the TM writes a string of the form $nnnnR\$$, and which point the $nnnnR\$$ command instantly disappears replaced by whatever is currently at location $nnnn$ on the RAM tape. (Here R, W, and \$ are special symbols used only on the RAM-access tape.)

Show that a RAM-TM is no more powerful than a normal TM, specifically that they recognize and decide the same classes of languages.

PROBLEM 3 (2 + 4 + 4 + 2 points)

Let $G = (V, \Sigma, R, S)$ where $V = \{S, V\}$, $\Sigma = \{a, b\}$, and R is the set of rules:

$$S \to bSS \mid aS \mid aV$$

$$V \to aVb \mid bVa \mid VV \mid \varepsilon$$

(A) In one sentence, what language does G generate?

(B) Transform G into an equivalent grammar G' in Chomsky normal form. Show your work.

(C) Verify that the string $abaab$ is generated by G', using the recognition algorithm for grammars in Chomsky normal form given in class. Show the complete filled-in matrix.

(D) Draw a parse tree for the derivation of $abaab$ from the transformed grammar G'.