

Please fill out the class feedback survey. (Link on piazza and course homepage.)

Harvard University  
Computer Science 121

Problem Set 6

Due Tuesday, October 30th 2012 at 11:59 PM.

Submit your solutions electronically on the course website, located at <http://people.seas.harvard.edu/~salil/cs121/fall12/>. On the site, click the “Problem Set Submission” button and provide your login info. Once logged in, place the solutions to Parts A, B, and C in separate files named lastname-ps6a.pdf, lastname-ps6b.pdf, and lastname-ps6c.pdf respectively, in the appropriate dropboxes.

Late problem sets may be turned in until Friday November 2nd 2012 at 11:59 PM with a 20% penalty.

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.

**PART A (Graded by Nick)**

PROBLEM 1 (8+4+6 points)

A function  $f : \Sigma^* \rightarrow \Sigma^*$  is *computable* if there is a Turing machine  $M$  that, on every input  $w \in \Sigma^*$ , halts (in state  $q_{\text{accept}}$ ) 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  $x \leq y$ . 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 following problem 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  $\Sigma$  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  $\Sigma$ , is decidable. (Thus, we can study computability of functions by studying decidability of languages.)

**Harvard University  
Computer Science 121**

**Problem Set 6**

Due Tuesday, October 30, 2012 at 11:59 PM.

Submit your solutions electronically on the course website, located at <http://people.seas.harvard.edu/~salil/cs121/fall12/>. On the site, click the “Problem Set Submission” button and provide your login info. Once logged in, place the solutions to Parts A, B, and C in separate files named lastname-ps6a.pdf, lastname-ps6b.pdf, and lastname-ps6c.pdf respectively, in the appropriate dropboxes.

Late problem sets may be turned in until Friday, November 2, 2012 at 11:59 PM with a 20% penalty.

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.

**PART B (Graded by Spencer)**

Note: For the following problems 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).

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  $\sigma$  is an alphabet symbol. Once this command is written the symbol  $\sigma$  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.

**Harvard University  
Computer Science 121**

**Problem Set 6**

Due Tuesday, October 30, 2012 at 11:59 PM.

Submit your solutions electronically on the course website, located at <http://people.seas.harvard.edu/~sali1/cs121/fall12/>. On the site, click the "Problem Set Submission" button and provide your login info. Once logged in, place the solutions to Parts A, B, and C in separate files named lastname-ps6a.pdf, lastname-ps6b.pdf, and lastname-ps6c.pdf respectively, in the appropriate dropboxes.

Late problem sets may be turned in until Friday, November 2, 2012 at 11:59 PM with a 20% penalty.

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.

**PART C (Graded by Thomas)**

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:

$$S \rightarrow bSS \mid aS \mid aV$$

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

- (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  $G'$ .