University of Nevada, Las Vegas
Computer Science 456/656 Fall 2005
Examination November 3, 2005

The entire examination is 200 points.

1. True or False. [5 points each]

   (a) _____ Every subset of a regular language is regular.

   (b) _____ The complement of every recursive language is recursive.

   (c) _____ The complement of every recursively enumerable language is recursively enumerable.

   (d) _____ The question of whether two context-free grammars generate the same language is undecidable.

   (e) _____ Given any context-free grammar G and any string $w \in L(G)$, there is always a unique leftmost derivation of w using G.

   (f) _____ Using multi-processors and other advanced technology, it is possible to design a machine which decides the halting problem.

   (g) _____ The question of whether two regular expressions are equivalent is decidable.

   (h) _____ Let $L = \{\langle M \rangle \mid M \text{ halts with no input}\}$. Then L is recursively enumerable.

   (i) _____ The class of languages accepted by non-deterministic Turing machines is the same as the class of languages accepted by Turing machines.

   (j) _____ Let $F(0) = 1$, and let $F(n) = 2^{F(n-1)}$ for any integer $n > 0$. Then F is a recursive function.

   (k) _____ There cannot exist any computer program that can decide whether any two C++ programs are equivalent.

2. Fill in the blanks.

   (a) [5 points] A machine which uses a given grammar to find the parse tree of any string generated by that grammar is called a _____.

   (b) [5 points] What class of machines accepts the class of context free languages? _____

   (c) [5 points] What class of machines accepts the class of recursively enumerable languages? _____

3. [15 points] What is the Church-Turing Thesis, and why is it important?

4. [15 points] What does it mean to say that a language can be enumerated in *canonical order* by a machine? What is the class of languages that can be enumerated in canonical order by a machine?

5. [15 points] Anita has developed a new programming language with lots of useful features. But her professor asks her to write a program (in this new language) that emulates a universal Turing machine. Why?

6. [25 points] Explain how, if a language is accepted by some non-deterministic Turing machine, it is also accepted by some deterministic Turing machine. (I don't need a formal proof, just a convincing outline. Assume you are writing it to convince a student who took this course but missed that lecture. Use words, examples, pictures, or whatever, to help explain it.)

7. [10 points] State the pumping lemma for context-free languages.

8. [20 points] Describe, in English, what the Turing machine in Figure 1 does. Hint: It only takes a few words. If you cannot explain it in the space alloted below, you do not have the right answer.

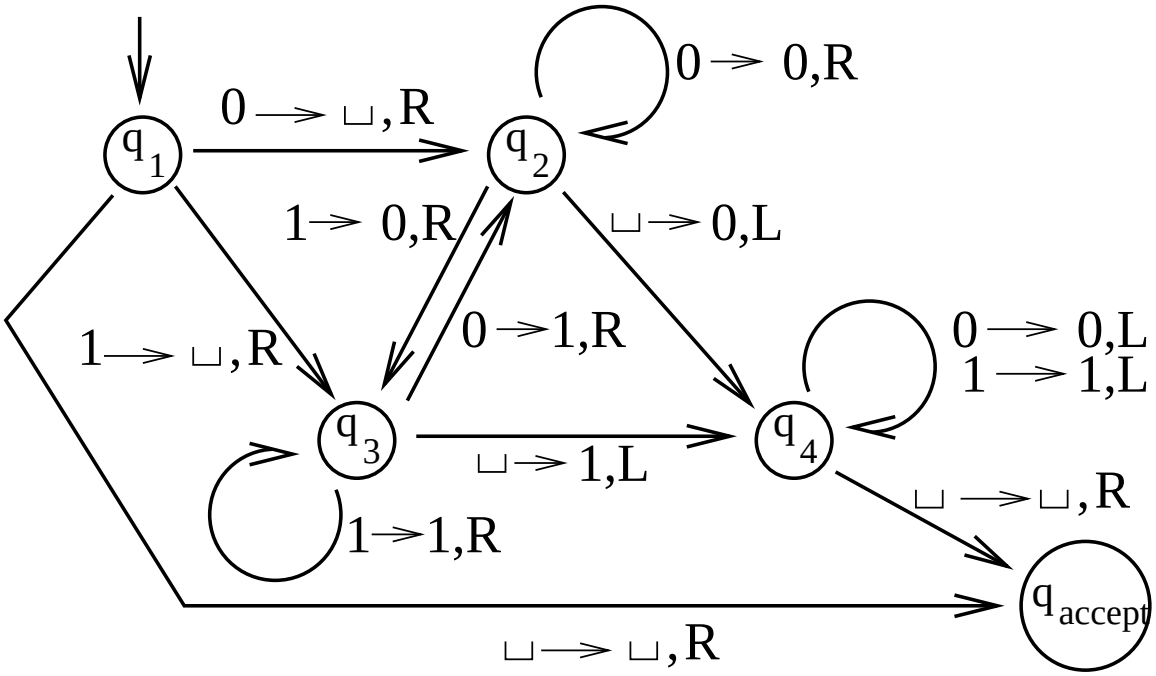


Figure 1: What does this TM do?

9. [30 points] Consider the context-free grammar G, with start symbol S and productions as follows:

   (a) $S \to SA$

   (b) $S \to SB$

   (c) $S \to -S$

   (d) $A \to -S$

   (e) $B \to *S$

   (f) $S \to a$

   (g) $S \to b$

   (h) $S \to c$

   Use the CYK algorithm to determine whether $w = -a - b * a - c$ is generated by G, using Figure 2.

   Each of the 36 cells of Figure 2 represents one non-empty substring of w. In each cell, write each grammar symbol that derives that particular substring in zero or more steps. If there are none, simply leave the cell blank.

   Do not insert lines between cells, since I did not ask you to construct a parse tree.

   If a symbol is generated in a cell in more than one way, only write it once. Each cell should contain zero or more grammar symbols, but no grammar symbol twice.

   I worked it, and got a total of 36 entries in all the cells. (Of course, I could have missed one.)

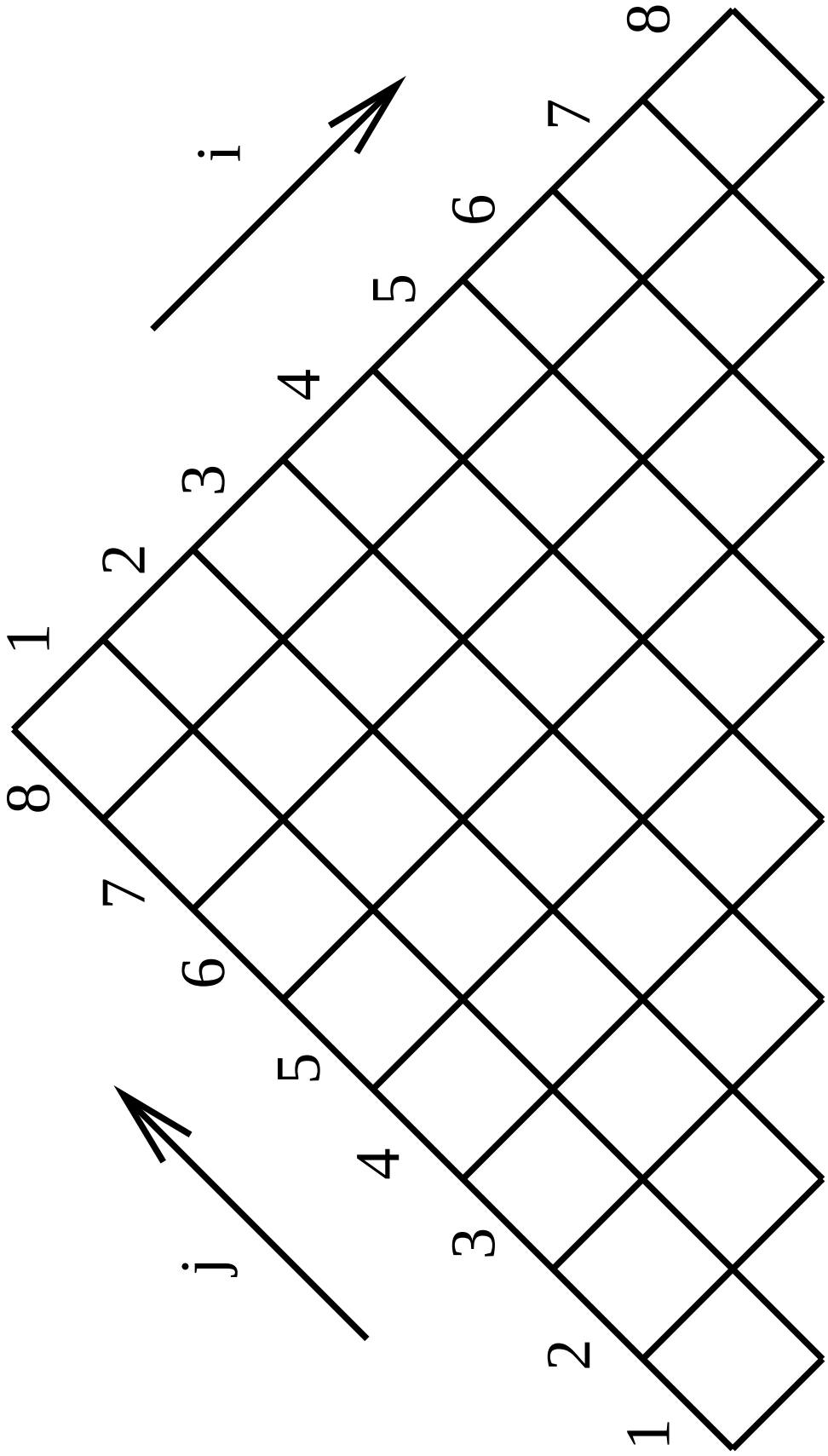


Figure 2: Fill in the cells to work problem 9.