

## Homework 6

Due: Wednesday, April 2, 2003.

### Guidelines

**Exercise 1** This exercise asks you to write a Turing machine program that decides if a tape contains three times as many 1's as there are 0's. We follow the standard assumption that our alphabet contains three different characters 1, 0, and \_ (empty). Furthermore you can assume that the blocks of characters on the tape is delimited on both sides by an empty character, and does not contain any further empty characters. Initially, the read write head of the Turing machine is positioned over the left most of the block of characters.

Upon termination, your tape should show a 1 to the left of the read/write head, if there were three times as many 1's as there are 0's, and a 0 otherwise. During operation, the original string may be consumed.

**Example:** If the the tape contains

...	1	0	1	0	1	1	1	1	...
-----	---	---	---	---	---	---	---	---	-----

your program should end in the following configuration.

...	1	?	...
-----	---	---	-----

In this setting, the “?” stands for any character and the position of the read write head is denoted by a box. *Hint: If you find it helpful you may extend the alphabet by a fourth character (let's call it \*), which can be used to mark those positions on the tape already visited and counted.*

1. Think about the states you need, and about which invariant each state satisfies. Simply give a list and an explanation in English.
2. Give the program as a state transition graph with states as vertices and actions as edges, marked by “x/ym” (x is the character found under the read and write head, y, the character to be written at the current position of the read write head, and m says in which direction the head is moving next.)
3. How many steps does your algorithm do in the worst case?
4. Categorize the runtime of your program in  $O(g(n))$  notation. Find the smallest class (without proof).
5. Implement your program on the Turing machine from class. The code can be found in the class directory under `/c/cs201/lib/code/ass6/turing.sml`. Run it on “01010000111” and “111011011101”.