CPSC 201: Introduction to Computer Science Carsten Schürmann Date: March 5, 2002

Homework 5

Due: 11:30am Friday March 8, 2002.

Problem 1: Turing Machines (50 points)

The Turing machine we have discussed in class is a deterministic Turing machine. This means that the first applicable instructions found in a program is deterministically executed. A non-deterministic Turing machine on the other hand is a Turing machine that selects one of all applicable instructions at random. Please edit the code discussed in class, and make it non-deterministic. When you program, you might want to use of the following functions. (You don't need to know what a random generator is, simiply use it.)

Random.rand : int * int -> Random.rand Random.rand (n, m) = R Generates a random generator (n, m are arbitrary). Random.randRange : int * int -> Random.rand -> int Random.randRange (n, m) R = i Generates an integer between n, and m using the random generator R. List.length : 'a list -> int List.length 1 = n n is the length of the list 1. List.nth : 'a list * int -> 'a

Hint: Edit the lookup function.

List.nth (1, i) = xx is the ith element in list 1.

Problem 2: Busy Beaver (50 points)

Write a Turing machine program using at most 5 states. The goal of this problem is it to generate as many 1's on the Turing tape as possible. This means that your Turing program is evaluated according to how many ones your Turing generated on the tape after termination. You should assume the tape to be blank before your machine is executed.

Your program will be tested on the deterministic Turing machine from class. The Turing program, that generates the most number of ones with n states, is so special, that it has been named *busy beaver*. Here is an example of the busy beaver program with 3 states.

```
val busybeaver = Program
[Command ((State 1, One), SOME ( One, State 3, Left)),
Command ((State 1, Blank), SOME ( One, State 2, Right)),
Command ((State 2, One), SOME ( One, State 2, Right)),
Command ((State 2, Blank), SOME ( One, State 1, Left)),
Command ((State 3, One), NONE),
Command ((State 3, Blank), SOME ( One, State 2, Left))]
```

Your assignment is *not* to find the busy beaver of size 5, but to write a program that generates as many 1's on the tape as possible.

- 1. Explain the idea behind your algorithm.
- 2. Design and describe the meaning of states.
- 3. Program your machine.

During spring break, we will be running a competition of your code against each other. For up to date results, check the course homepage. The award ceremony will take place in class after spring break. Good luck, and get ready to find the busy beaver.