02240 Computability and Semantics
Informatics and Mathematical Modelling
DTU

Thomas Bolander
adapted by MRH

Turing Machine Simulation 1

The purpose of this exercise is to simulate Turing Machines on the basis of an SML representation of Turing Machines. This exercise is part of mandatory assignment 3.

For simplicity we will only be considering Turing machines that adhere to the following conventions:

- The states are named $q_0, q_1, q_2, \ldots, q_n$ for some n. The state q_0 is always taken to be the start state, and the state with the highest index, q_n, is the final state.
- The set of tape symbols is $\{1, 0, X, B\}$ where B is the blank symbol and the rest are input symbols.

States, tape symbols and directions are modelled in SML by the following datatypes:

```
datatype State = q of int
datatype TapeSymbol = I | O | X | B
datatype Direction = L | R
```

Turing machines can be represented by their transition relations. In SML, we represent these transition relations as objects of type `TransRel`:

```
type TransRel = ((State * TapeSymbol) * (State * TapeSymbol * Direction)) list
```

Example. In Example 8.4 of *Introduction to Automata Theory, Languages, and Computation*, a Turing machine for computing the proper subtraction function is given by the transition table presented in Figure 1.

State	0	1	B
q_0	(q_1, B, R)	(q_5, B, R)	$-$
q_1	$(q_1, 0, R)$	$(q_2, 1, R)$	$-$
q_2	$(q_3, 1, L)$	$(q_2, 1, R)$	(q_4, B, L)
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	(q_0, B, R)
q_4	$(q_4, 0, L)$	(q_4, B, L)	$(q_6, 0, R)$
q_5	(q_5, B, R)	(q_5, B, R)	(q_6, B, R)
q_6	$-$	$-$	$-$

Figure 1: A Turing machine that computes the proper subtraction function.

In SML we would represent this table by the value of type `TransRel` given in Figure 2.

In the databar you have access to a Turing machine simulator called xTuring. To run xTuring you should type the following command in a terminal window from the directory containing `TM1.sml`:

```
~v6474/xturing
```

```
[((q 0,0),(q 1,B,R)),((q 0,I),(q 5,B,R)),
 ((q 1,0),(q 1,0,R)),((q 1,I),(q 2,I,R)),
 ((q 2,0),(q 3,I,L)),((q 2,I),(q 2,I,R)),((q 2,B),(q 4,B,L)),
 ((q 3,0),(q 3,0,L)),((q 3,I),(q 3,I,L)),((q 3,B),(q 0,B,R)),
 ((q 4,0),(q 4,0,L)),((q 4,I),(q 4,B,L)),((q 4,B),(q 6,0,R)),
 ((q 5,0),(q 5,B,R)),((q 5,I),(q 5,B,R)),((q 5,B),(q 6,B,R))]
```

Figure 2: The SML representation of the transition table in Figure 1.

The simulator can be used to simulate Turing machines represented as SML values of type `TransRel`. This is done by using the function

```
MakeXTfile: TransRel -> string -> unit
```

declared in `TM1.sml`. Given a Turing machine `M` and a filename `s`, writing

```
MakeXTfile M s
```

will produce a file `s` that can be loaded into xTuring to simulate the Turing machine `M`.

a) Test the Turing machine simulator on the subtraction machine given in Figure 2. The file `TM1.sml` contains an SML value `submachine` for this Turing Machine. You should first execute

```
MakeXTfile submachine "submachine.xt"
```

Then you should load the file `submachine.xt` into xTuring. You can then write a string of tape symbols on the tape, move the head to the desired initial position and click 'Run'. You should test the Turing machine on the following initial ID's:

$$q_0 0010$$
$$q_0 000100$$
$$q_0 100$$
$$q_0 001$$

Explain what is calculated in each case (cf. Example 8.4).

b) Give the graphical representation of a Turing machine M_{add} that converts an ID of the form

$$\alpha 1 0^m 1 0^n q_0 B$$

to

$$\alpha 1 0^{m+n} q_f B$$

where α is any string of tape symbols, n and m are non-negative integers, and q_f is the final state of the machine. Represent the machine as an SML value `ADDmachine` of type `TransRel` and test the machine using xTuring.