UNIVERSITY OF CAMBRIDGE

Department of Computer Science and Technology

Raspberry Pi

Section 4: Turing Machine Example Programs

The first 4 programs are found from this website and are very good demonstrations of what a 3-symbol Turing machine can do.

Binary Counter

This program reads the current binary number printed on the tape and increments it by 1 before stopping.

What does binary mean?

In decimal (base-10) numbers written using 10 different symbols, '0' to '9'. In binary (base-2), numbers are written using only 2 symbols, '0', and '1'.

For example, 902 = 900 + 0 + 2 = **9** * 10^2 + **0** * 10^1 + **2** * 10^0, and hence, is written as '**902**' in decimal.

Similarly, 11 = 8 + 0 + 2 + 1 = **1** * 2^3 + **0** * 2^2 + **1** * 2^1 + **1** * 2^0, and hence is written as '**1011**' in binary.

State Table

State	Symbol Read	Write Instruction	Move Instruction	Next State
State 0	Blank	Write 'Blank'	Move tape right	State 1
	0	Write '0'	Move tape left	State 0
	1	Write '1'	Move tape left	State 0
State 1	Blank	Write '1'	Move tape left	State 2
	0	Write '1'	Move tape right	State 2
	1	Write '0'	Move tape right	State 1
State 2	Blank	Write 'Blank'	Move tape right	Stop State
	0	Write '0'	Move tape left	State 2
	1	Write '1'	Move tape left	State 2

The program follows a very simple procedure involving the addition of '1' to the number on the tape.

Firstly, in State 0, the machine moves the tape left until it is reading the last digit of the number, which we do when we do a simple addition by hand. [video]

Next, in State 1, it adds '1' to the digit that it is reading. If it is a '0', it writes a '1' to increment it, and we are done with this step. [video]

If it is a '1', the machine replaces it with a '0', (similar to adding a 1 to a 9 in decimal), and we will have to **carry** a '1' over to the digit to the left by moving the tape right, before repeating this step of the process on that digit. [video]

Finally, in State 2, the machine merely moves the tape left again so that the right-most digit is at the head before terminating the program. (This step is not really necessary but it makes things neater.) [video]

Unary Subtraction

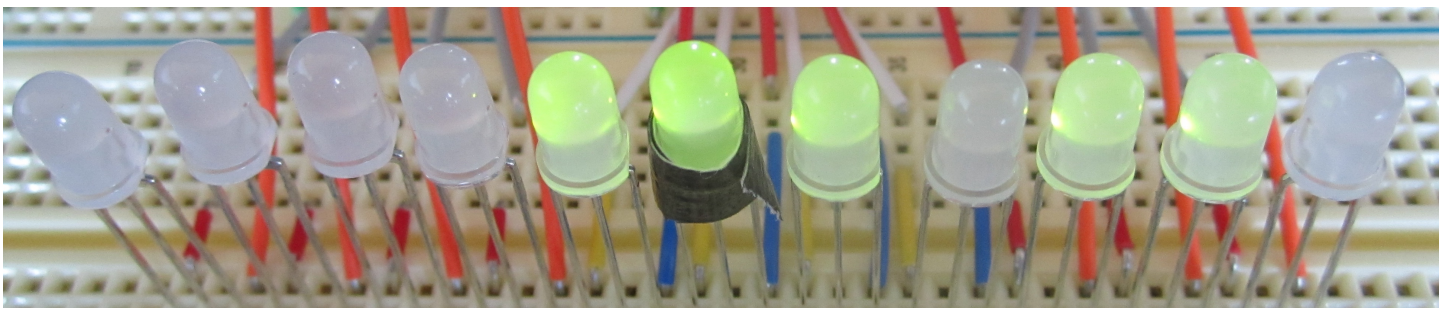
This program performs a subtraction operation on 2 unary numbers which are currently on the tape. Unary means base-1 and it only consists of the symbol '1', so '3' is '111' while 5 is '11111' and so on.

State Table

State	Symbol Read	Write Instruction	Move Instruction	Next State
State 0	Blank	Write 'Blank'	Move tape left	State 1
	0	Write '0'	Move tape left	State 0
	1	Write '1'	Move tape left	State 0
State 1	Blank	Write 'Blank'	Move tape right	State 2
	0	Write '0'	Move tape left	State 1
	1	Write '1'	Move tape left	State 1
State 2	Blank	Write 'Blank'	Move tape left	State 4
	0	Write '0'	Move tape right	State 2
	1	Write '0'	Move tape right	State 3
State 3	Blank	Write 'Blank'	Move tape right	State 7
	0	Write '0'	Move tape right	State 3
	1	Write '1'	Move tape right	State 3
State 4	Blank	Write 'Blank'	Move tape right	State 5
	0	Write '0'	Move tape left	State 4
	1	Write '1'	Move tape left	State 4
State 5	Blank	Write 'Blank'	Move tape right	State 6
	0	Write 'Blank'	Move tape right	State 5
	1	Write '1'	Move tape right	State 5
State 6	Blank	Write 'Blank'	Move tape left	Stop State
	0	Write 'Blank'	Move tape right	State 6
	1	Write '1'	Move tape right	State 6
State 7	Blank	Write 'Blank'	Move tape left	Stop State
	0	Write '0'	Move tape right	State 7
	1	Write '0'	Move tape left	State 0

This program has 8 states but it is not too complicated. Basically, it alternates between the two numbers and replaces the right-most '1' from each number with a '0'. The machine repeats this until the smaller number has no more '1's and it then erases all the '0's, leaving behind the difference between the 2 numbers.

I shall demonstrate this using an example of '3' - '2' or '111' – '11' and we need to initialise the LEDs as shown below. The larger number has to be placed to the left of the smaller number with a blank symbol in between, and the head has to be positioned at any digit of the larger number.



In States 0 and 1, the machine moves the tape left until it reads the last digit of the smaller number before transiting to State 2, where it searches the smaller number for a '1' and replaces it with a '0'. [video]

The machine then moves to the last digit of the larger number (State 3), before searching the larger number for a '1' and replacing it with a '0' in State 7. [video]

The whole process repeats from State 0 again until the smaller number runs out of '1's. [video]

Now the machine is back in State 2 where it is searching the smaller number for a '1' but it fails to do so and encounters a 'blank' symbol instead, so it moves back to the last digit of the smaller number in State 4. [video]

Finally, the machine removes the '0's in the smaller number (State 5), and then those in the larger number (State 6) and terminates the program when it is done. [video]

Try out other combinations yourself!

3-State Busy Beaver

The busy beaver problem is to find the most number of '1's that a 2-symbol, n-state Turing machine can print on an initially blank tape before halting. The aim of this is to find the smallest program which outputs as much data as possible, and can also stop eventually. Visit this website for more information on the problem.

For a 3-State machine, the maximum number of '1's that it can print is proven to be 6, and it takes 14 steps for the Turing machine to do so. The state table for the program is shown below. Since only 2 symbols are required, the instructions for the '0' symbol are left as the default settings.

State Table

State	Symbol Read	Write Instruction	Move Instruction	Next State
State 0	Blank	Write '1'	Move tape left	State 1
	0	Write 'Blank'	None	State 0
	1	Write '1'	None	Stop State
State 1	Blank	Write 'Blank'	Move tape left	State 2
	0	Write 'Blank'	None	State 0
	1	Write '1'	Move tape left	State 1
State 2	Blank	Write '1'	Move tape right	State 2
	0	Write 'Blank'	None	State 0
	1	Write '1'	Move tape right	State 0

[video]

4-State Busy Beaver

For a 4-state busybeaver, the maximum number of '1's that can be printed is 13, and it takes 107 steps.

State Table

State	Symbol Read	Write Instruction	Move Instruction	Next State
State 0	Blank	Write '1'	Move tape right	State 1
	0	Write 'Blank'	None	State 0
	1	Write '1'	Move tape left	State 1
State 1	Blank	Write '1'	Move tape left	State 0
	0	Write 'Blank'	None	State 0
	1	Write 'Blank'	Move tape left	State 2
State 2	Blank	Write '1'	Move tape right	Stop State
	0	Write 'Blank'	None	State 0
	1	Write '1'	Move tape left	State 3
State 3	Blank	Write '1'	Move tape right	State 3
	0	Write 'Blank'	None	State 0

	1	Write ‘Blank’	Move tape right	State 0

[video]

As there are only 11 LEDs, the machine is unable to display all 14 lit LEDs but it can be checked by moving the tape and counting the number of green LEDs. [video]

Palindrome Detector

This program checks a string of symbols on the tape and returns a ‘1’ (green) if it is a palindrome and a ‘0’ (red) if it is not.

State Table

State	Symbol Read	Write Instruction	Move Instruction	Next State
State 0	Blank	Write ‘Blank’	Move tape left	State 1
	0	Write ‘0’	Move tape right	State 0
	1	Write ‘1’	Move tape right	State 0
State 1	Blank	Write ‘1’	None	Stop State
	0	Write ‘Blank’	Move tape left	State 2
	1	Write ‘Blank’	Move tape left	State 4
State 2	Blank	Write ‘Blank’	Move tape right	State 3
	0	Write ‘0’	Move tape left	State 2
	1	Write ‘1’	Move tape left	State 2
State 3	Blank	Write ‘Blank’	None	State 0
	0	Write ‘Blank’	Move tape right	State 0
	1	Write ‘1’	None	State 6
State 4	Blank	Write ‘Blank’	Move tape right	State 5
	0	Write ‘0’	Move tape left	State 4
	1	Write ‘1’	Move tape left	State 4
State 5	Blank	Write ‘Blank’	None	State 0
	0	Write ‘0’	None	State 6
	1	Write ‘Blank’	Move tape right	State 0
State 6	Blank	Write ‘0’	None	Stop State
	0	Write ‘Blank’	Move tape right	State 6
	1	Write ‘Blank’	Move tape right	State 6

The program first moves the tape until it is reading the first symbol of the string in State 0. It then checks the symbol in State 1 and changes it to a ‘blank’ symbol. [video]

If the symbol read is ‘0’, it transits to State 2 where it moves the tape until it is reading the last symbol and checks if it is a ‘0’ too in State 3. If the last symbol is a ‘0’, the string could possibly be a palindrome, so the machine changes it to a ‘blank’ symbol before returning to State 0 where it checks the next symbol. [video]

Otherwise, the machine changes all the symbols to ‘blank’ symbols and prints a ‘0’ before stopping in State 6. [video]

States 4 and 5 are similar to States 2 and 3 but are for processing the tape when the symbol read is ‘1’ instead of ‘0’. Finally, when the last symbol has been checked and a palindrome is confirmed, the machine prints a ‘1’ and stops. [video]

© 2012 Robert Mullins
Information provided by Robert Mullins