

Turing Machine as function computer

Prof. (Dr.) K.R. Chowdhary
Email: kr.chowdhary@iitj.ac.in
Webpage: www.krchowdhary.com

Formerly at department of Computer Science and Engineering
MBM Engineering College, Jodhpur

Monday 10th April, 2017

- Turing machine can compute functions, like $y = f(x)$. Initially a value corresponding to x is put on the tape, and Turing machine is run. after having performed the computations, the result $f(x)$ will replace the original value, and head returns to begin.
- The input is usually represented in unary format. For 0 is represented 1, the 1 is represented as 11 and n is represented as 1^{n+1} .
- To add two numbers m and n represented as 1^{m+1} and 1^{n+1} . Their sum $m + n$ is 1^{m+n-1} .
- The set of arguments for the function $f(x_1, x_2, \dots, x_n)$ is put on the tape as a string of 1s and 0s, having value $B1^{x_1}B1^{x_2}B\dots i^{x_n}$.

Function Computing with TM

To find out the sum of $m + n$, the arguments are stored on tape in the begin as $B1^{m+1}B1^{n+1}B$, with ID as $q_0B1^{m+1}B1^{n+1}B$. The head will scan the tape, replacing the center B separator with 1, finally two 1s at the end are replaced by 0s, to obtain ID as q_jB1^{m+n-1} .

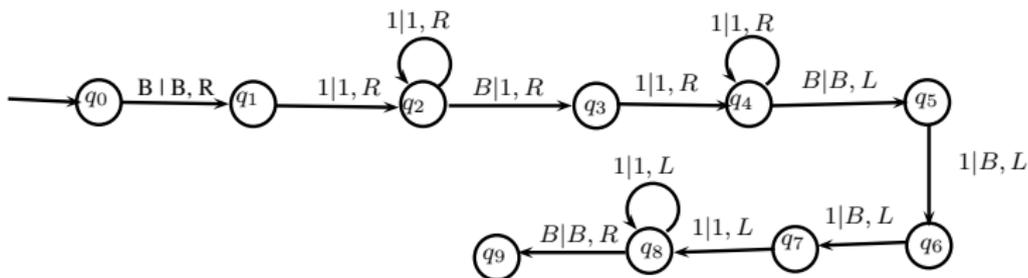


Figure 1: Turing machine adding m and n .

Primitive recursion functions: Successor function

- All computations in TM can be done using primitive recursive (PR) functions. The zero (Z) and successor (S) functions are PR and composition of PR with PR are PR, and nothing else are PR.
- For example, to find $3 + 2$, we can do this by $\text{succ}(\text{succ}(3))$. This is nothing but $S \circ S(3)$, which is composition of the function S with itself. If the n is an integer, represented in unary as 1^{n+1} , then its successor is 1^{n+1} , and can be computed using TM as follows:

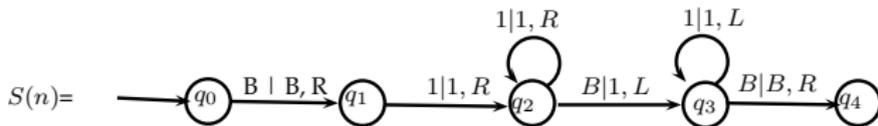


Figure 2: Turing machine computing the successor function.

Successor function for binary number

- Sample Rules:
If read 1, write 0, Go Right,
repeat.
If read 0, write 1, Rewind and
halt
- If read B, write 1, rewind and
halt
- Let's see how they are carried
out on a piece of paper that
contains the binary
representation of 47 in reverse
order:**bold** number represents
position of head.

1 1 1 1 0 1 B

0 **1** 1 1 0 1 B

0 0 **1** 1 0 1 B

0 0 0 **1** 0 1 B

0 0 0 0 **0** 1 B

0 0 0 0 **1** 1 B ;Rewinds and
HALTS; the result is 48 in
reverse order.

- So the successor's output on
111101 was 000011 which is
the reverse binary
representation of 48.

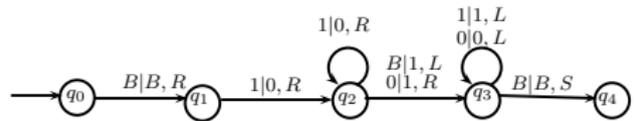


Figure 3: Successor TM for binary number.

Primitive recursive function: Zero (Z)

- The Zero function is PR, and its effect is to make tape contents equal to zero.
- Thus, if contents of tape is 1^n or it is 1^m , after running Zero (Z) function, the result on tape is 1, which stands for zero. The following is TM for zero function.

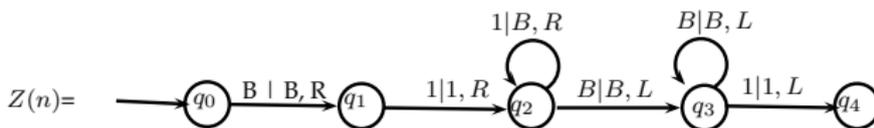


Figure 4: Turing machine computing the zero.

Primitive recursive function: Pred (n)

- The predecessor function (pred) is a partial recursive function. Its effect is to make tape contents reduced by 1.
- Thus, if contents of tape is 1^{n+1} , after running pred function, the result on tape is 1^n . However, if it is zero, $\text{pred}(1) = 1$ (zero), because predecessor of 0 is zero only, since we are dealing with positive integers.

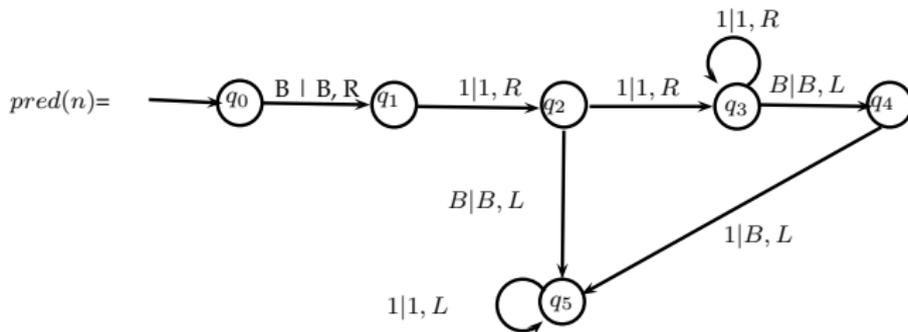


Figure 5: Turing machine computing the $\text{pred}(n)$ function.

Concatination of functions.

- To build complex TMs or to compute the complex functions, these primitive recursive functions are called as library function, are joined in sequential order. For example, to make tape zero (initialize), then add 2 into the present value, the functions, Z, S, S are called in sequence.
- When the functions are concatenated like this, the final (accepting/halting) state of the previous function becomes the start of state of the next function, as shown in the figure below.

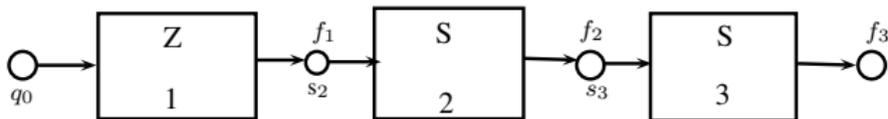


Figure 6: using more than one PR functions.

Visualization (Simulation) of TM.

- Considering calling the successor function. Let the tape content is 111, we can show the computation as shown in figure 7.

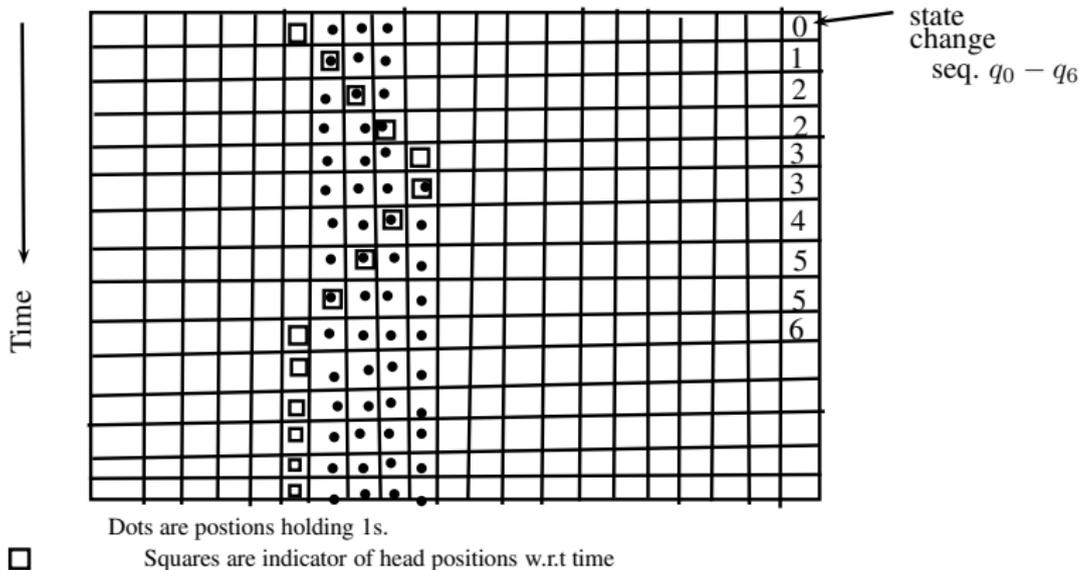


Figure 7: Simulation (visualization) of TM.