Computability and Logic

HW 6

Due: Friday, April 10

1. Use the Turing-machine software (see guidelines on back) to create a Turing-machine that, when started with [n,m,k], will halt with [n,m-1,k] (you can assume m is greater than 0)
2. Use the Turing-machine software to create a Turing-machine that computes the max(x,y) function (i.e. returns the maximum of the two numbers x and y)
3. For each of the following, either use the software to create a Turing machine that accomplishes the desired task, or carefully explain why there cannot be such a machine.
    a. A Turing-machine that, started anywhere on the tape, will eventually halt if and only if the tape was completely blank (all 0)
    b. A Turing-machine that, started anywhere on the tape, will eventually halt if and only if the tape was not completely blank (all 0)
4. Use the software to create the 'Copy' Turing-machine: a Turing-machine that for all n, when started on [n], halts with [n,n].
5. Use the software to create the 'Double' Turing-machine: a Turing-machine that for all n, when started at the left most 1 of n consecutive 1's on an otherwise blank tape, halts at the left most 1 of 2n consecutive 1's on an otherwise blank tape. Or, what is the same thing: given the 'standard' encoding as defined on the back, create a Turing-machine that for all n, when started on [n], halts with [2n+1].

See specific guidelines and conventions and other things to think about on the back!

Please zip all your files when submitting electronically to me.

For all Turing-machines you create for the HW, follow the following guidelines:

- You cannot use symbols other than 0 and 1.
- You do not have to use an explicit halting state
- When computing functions over natural numbers, use the 'standard' convention of using the following tape/head configuration $[n_1, n_2, \ldots n_k]$ to represent a tuple of numbers $<n_1, n_2, \ldots n_k>$: On an otherwise all 0 tape, there is a block of n_1+1 consecutive 1's, followed by a single 0, followed by a block of n_2+1 consecutive 1's, followed by a single 0, …, followed by a block of n_k+1 consecutive 1's, the head is at the leftmost 1 of the leftmost block of 1's. Use this convention for input as well as output (i.e. if the answer is n, your machine should halt with [n], i.e. make sure the head is at the leftmost 1 of a block of n+1 consecutive 1's and the tape is otherwise all 0!)
- Organize the nodes and transitions so that your machine looks nice and is 'readable' (i.e. don't leave your machine a spaghetti of connections). Try to keep crossing connections at a minimum.
- Make sure to test the 'edge' cases (x = 0, y = 0, x = y, etc)
- Save your Turing-machine using the 'Save' option. Do *not* use the 'Save Graph' option! ('Save graph' saves the machine as an abstract mathematical graph, i.e. it saves the nodes and connections between them, but does *not* save the graphical layout of your machine!)