CS 374 Lab 9: Church-Turing Thesis

Date: February 17, 2016.

Problem 1. [Category: Design] A k-tape Turing machine is a TM that can read and write onto k-tapes. It starts out with the input being written on the first tape, and the remaining $k-1$ tapes being blank. Formally, $M = (Q, \Sigma, \Gamma, \square, \delta, \text{start}, \text{accept}, \text{reject})$, where Q is a finite set of states, Σ is the input alphabet, Γ is the tape alphabet, $\square \in \Gamma \setminus \Sigma$ is the blank symbol, δ is the transition function, and start, accept and reject are the start, accept and reject states, respectively.

1. What is the domain and co-domain of the function δ?
2. What does a configuration of such a machine look like?
3. Given an arbitrary k-tape TM M, sketch out the construction of a 1-tape TM N that accepts the same language as M.

Problem 2. [Category: Design] A Random Access Machine (RAM) is a computing device that has finite many registers that can store numbers, and infinitely many memory locations each of which can store an arbitrary natural number. Initially the RAM has a program (sequence of instructions) stored in the first few cell of the the memory; all other memory locations, and all registers initially contain 0. A RAM program consists of the following instructions.

- `add X, Y`: Add the contents of registers X and Y and store the result in X.
- `loadc X, I`: Place the constant I in register X.
- `load X, M`: Load the contents of memory location M into register X.
- `loadI X, M`: Load the contents of the location "pointed to" by the contents of M into register X.
- `store X, M`: store the contents of register X in memory location M.
- `jmp M`: The next instruction to be executed is in location M.
- `jmz X, M`: If register X is 0, then jump to instruction M.
- `halt`: Halt execution.

Prove that an RAM can be simulated by a k-tape TM (for some k).