MA/CSSE 474 Day 32 Announcements and Summary

Announcements:

1) Exam 3 Feb 9. Will cover Sections 11.1-11.8, 12.1-12.4, 12.6 13.1-13.5, 13.8, 14.1-14.3, 17.1-17.3
   HW 10-14, Lectures 19-31.
2) You may have noticed that the course material has become more difficult lately. So you should expect some harder questions on this exam.
3) HW15 Feb 11, HW16 Feb 15, HW17 no turn-in Final Exam Feb 25, 8:00 AM GM room.

Main ideas from today

1) **Exercise:** Use multiple tapes to multiply two natural numbers represented in binary. Description can be high-level.

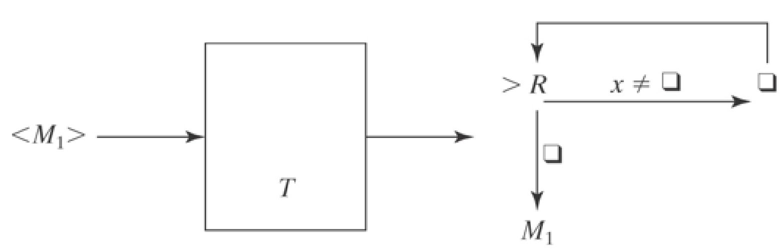
2) **Encoding a TM** M = (K, Σ, Γ, δ, s, H) as a string <M>:
   i) **Encoding the states**: Let i be $\lceil \log_2(|K|) \rceil$.
      (1) Number the states from 0 to $|K|$-1 in binary (i bits for each state number):
      (2) The start state, s, is numbered 0; Number the other states in any order.
      (3) If t' is the binary number assigned to state t, then:
         (a) If t is the halting state y, assign it the string yt'.
         (b) If t is the halting state n, assign it the string nt'.
         (c) If t is the halting state h, assign it the string ht'.
         (d) If t is any other state, assign it the string qt'.
   ii) **Encoding the tape alphabet**: Let j be $\lceil \log_2(|\Gamma|) \rceil$.
      (1) Number the tape alphabet symbols from 0 to $|\Gamma|$ - 1 in binary.
      (2) The blank symbol is number 0.
      (3) The other symbols can be numbered in any order
   iii) **Encoding the transitions**:
      (1) (state, input, state, output, move_direction)
      (2) Example: (q000,a000,q110,a000,$\rightarrow$)
   iv) **Encoding s and H** (already included in the above)
   v) A **special case** of TM encoding
      (1) One-state machine with no transitions that accepts only ε is encoded as (q0)
   vi) **Encoding other T**Ms: It is just a list of the machine's transitions:
      (1) Detailed example on slide
   vii) Consider the alphabet Σ = {(,), a, q, y, n, h, 0, 1, comma, $\rightarrow$, $\leftarrow$}. Is the following question decidable?
      (1) Given a string w in Σ*, is there a TM M such that w = <M> ?

3) We can **enumerate all TMs**, so that we have the concept of "the ith TM"
4) We can have **processes** (TMs?) **whose input and outputs are TM encodings**:

Input: a TM M_1 that reads its input tape and performs some operation P on it.

Output: a TM M_2 that performs P on an empty input tape.

5) **Encoding multiple inputs**: <x_1, x_2, …x_n>



6) **Specification of U**, the Universal Turing Machine (UTM):
   a) U starts with <M,w> on its input tape, then simulates M's action when it has input w:
   b) Ualts iff *M* halts on *w*.
   c) If *M* is a deciding or semideciding machine, then:
      i) If *M* accepts, U accepts.
      ii) If *M* rejects, U rejects.
   d) If *M* computes a function, then *U*(<*M*, *w*>) must equal *M*(*w*).

7) **Operation of U**
   a) Three tapes:
      i) M's tape
      ii) <M>
      iii) M's state
   b) Initialize U:
      i) start with <M,W> on tape 1
      ii) Move the <M> p0art to tape 2, leaving <w> on tape 1.
      iii) Figure out how many bits in encoded states, and use this to write <s> on tape 3.
   c) U simulates a move of M. Repeat:
      i) On tape 2 find a quintuple on tape 2 (if any) that matches the current state and tape symbol
      ii) Perform the transition by appropriately changing tapes 1 and 3
      iii) If no matching quintuple on tape 2, halt
      iv) If U halts, report the same info that M would report.

8) **The Church-Turing Thesis**: If it is computable, it can be computed by a Turing Machine.