MA/CSSE 474 Day 33 Summary

1) **TM extensions**. For each extension, we can show that every extended machine has an equivalent basic machine.
   a) **Multi-track TM. Implementation:** Input symbols are tuples of the input symbols from the tracks
   b) **Multiple-tape TM**
      i) The transition function for a k-tape Turing machine:

2) Example: Use two tapes to add two natural numbers (represented in binary)

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

3) ***Theorem** (adding tapes adds no computing power):* Let $M = (K, \Sigma, \Gamma, \delta, s, H\}$ be a k-tape Turing machine for some $k > 1$. Then there is a standard TM M' where $\Sigma \subseteq \Sigma'$, and:
   (1) On input x, M halts with output z on the first tape iff M' halts in the same state with z on its tape.
   (2) On input x, if M halts in n steps, M' halts in $O(n_2)$ steps.
      (a) Proof by construction:
         (i) Treat the single tape as if it were multi-track. This gives M' a large number of tape symbols.

4) **Encoding a TM** $M = (K, \Sigma, \Gamma, \delta, s, H)$ as a string <M>:
   i) **Encode 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'.

   Example:

   q0000
   q0001 q0010 y0011
   n0100 q0101 q0110
   q0111 q1000

   ii) **Encode 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 always symbol number 0.
      (3) The other symbols can be numbered in any order.

   Example: $\Gamma = \{ \square, b, c, d \}$.

$\square$ =	a00
b =	a01
c =	a10
d =	a11

iii) **Encode a transition**: (state, input, state, output, move_direction). **Example:** (q000,a000,q110,a000,→)
iv) **Encode 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) **Encode other T**Ms: The encoding is just a list of the machine's transitions (in any order). Details on slides.

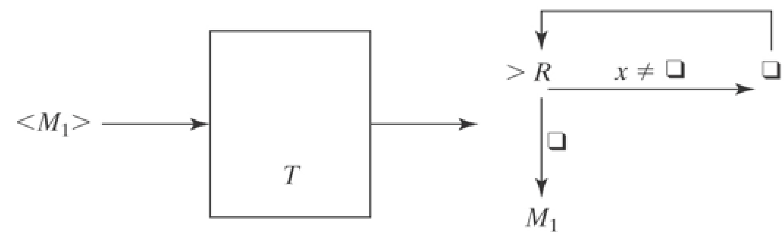
vii) Consider the alphabet Σ = {(,), a, q, y, n, *h,* 0, 1, comma, →, ←}. Is the following question decidable?
(1) Given a string w in Σ*, is there a TM M such that w = <M> ?

5) We can **enumerate all TMs**, so that we have the concept of "the ith TM"
6) 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.

7) **Encoding multiple inputs**: $<x_1, x_2, \ldots x_n>$



8) **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) U halts 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*).

9) 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> part 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.

10) **How long does U take to run?**

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