MA/CSSE 474 Day 34 Summary

1) Recap of TM encoding from last class (a place for your notes, if any).

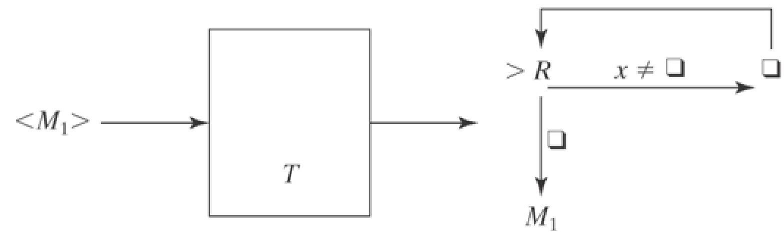
We can **enumerate all TMs**, so that we have the concept of "the ith TM"

2) 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.

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



4) **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*).

5) 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.

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

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

8) **Recap: TMs as** language **recognizers**. Let $M = (K, \Sigma, \Gamma, \delta, s, \{y, n\})$.
   a) *M **accepts*** a string *w* iff $(s, \square w) \vdash_M^* (y, w')$ for some string w'.
   b) *M **rejects*** a string *w* iff $(s, \square w) \vdash_M^* (n, w')$ for some string w'.
   c) *M **decides*** a language $L \subseteq \Sigma^*$ iff for any string $w \in \Sigma^*$ it is true that:
      i) $w \in L \rightarrow M$ accepts *w*, and
      ii) $w \notin L \rightarrow M$ rejects *w*.
   d) A language *L* is ***decidable*** (in the set ***D***) iff there is a Turing machine *M* that decides it.
   e) *M **semidecides*** *L* iff, for any string $w \in \Sigma_M^*$:
      i) $w \in L \rightarrow M$ accepts *w*
      ii) $w \notin L \rightarrow M$ does not accept *w*. For each string *w*, *M* may either reject or fail to halt.
   f) A language *L* is ***semidecidable*** (in the set ***SD***) iff there is a Turing machine that semidecides it.
   g) Clearly SD contains D, and SD is properly contained in the set of all languages (countability argument)
9) **Complement:** relative to what universe?
   a) For now, we will usually consider that universe to be the set of strings that fit the syntax of the problem.
   b) Define the ***complement*** of any language *L* whose member strings include at least one Turing machine description to be with respect to a universe of strings that are of the same syntactic form as *L*.
   c) If L_1 = {<*M*, *w*> : TM *M* halts on input string *w*}, then $\neg L_1$ = {<*M*, *w*> : TM *M* does not halt on input string *w*}.
10) **The language H** = {<*M*, *w*> : TM *M* halts on input string *w*} Is H decidable?
   a) Of course we can decide halting for specific simple TMs. Or can we? (Collatz conjecture, 1937, still no proof).
   b) It's easy to see that H is semidecidable.
      M'_H(<*M*, *w*>) =
         1. Run *M* on *w*.
         2. Accept.
      i) M'_H accepts <*M*, *w*> iff M halts on input w.
      ii) *So* M'_H semidecides H