16 April 2018: Universal Turing machine, Diagonalization

Def. (Univ TM) A universal Turing machine is a Turing machine, $\mathcal{U}$, with alphabet $\Sigma \supseteq \{\triangleright, \sqcup, \emptyset, 1, (,), `,', `;'\}$ such that when the input to $\mathcal{U}$ is in the format $x;y$ where

- x is the description of a Turing machine M
- y is the binary encoding of an input to M

then the outcome of the computation $\mathcal{U}(x;y)$ equals the outcome of the computation $M(y)$.
Here "outcome" denotes an element of $\{\text{yes}, \text{no}, \text{halt}, \nearrow\}$ reflecting the termination status (or lack thereof) for a computation.

Remark about abuse of notation: in the definition above "y" refers to the input string of M and to the encoding of that string in binary.

How $\mathcal{U}$ works: It's a multi-tape TM with 5 tapes.

1. Input tape: contains $x;y$. Never overwrite. Read only.
2. Description tape: contains x. (Description of M.) Read only after init.
3. Working tape: stores the binary encoding of M's tape throughout the simulation, with commas separating each binary encoded symbol.
4. State tape: stores binary encoding of M's current state in the simulation.
5. Special tape: stores the binary encodings of $\{\text{halt}, \text{yes}, \text{no}\} \cup \{\leftarrow, \rightarrow, -\}$.
   Read only after init.

For plain-English description of how it works, and pseudocode, consult lecture notes.

$\exists$ a single-tape machine $\tilde{\mathcal{U}}$ that simulates the 5-tape machine $\mathcal{U}$.

$\tilde{\mathcal{U}}$ is the one that really ought to be called a universal TM.

Diagonalization and Undecidability

Def If M is a Turing machine with alphabet Σ and x is a string over Σ,

"M accepts x" means $M(x) = \text{yes}$. (The computation of M on input x ends in the 'yes' state.)

For any machine M,

$$L(M) \triangleq \{x \mid M \text{ accepts } x\}.$$

$L \subseteq \Sigma^*$ is recursively enumerable (r.e.) if $\exists$ machine M such that $L = L(M)$.

$L \subseteq \Sigma^*$ is decidable if $\exists$ machine M s.t. $\forall x \in L \quad M(x) = \text{yes}$, $\quad \forall x \notin L \quad M(x) = \text{no}$.

Difference between these definitions: when L is r.e. M is allowed to not accept string x by running forever. When L is decidable, M is required to say "no" after some finite amt of time if $x \notin L$.

This difference is crucial. Consider, for example,

$$L = \{ x;y \mid U(x;y) \neq \nearrow \}.$$

L = "the set of all pairs $x;y$ such that
x describes a TM M, and
y describes an input to M, and
M terminates when processing that input."

If U' is a universal TM rewired to always stop in state "yes" when U transitions to {yes, no, halt} then $L = L(U')$. So L is r.e.
We will see that L is not decidable.