The Universal Turing Machine

11/9/18

The universal TM U is a TM that is "programmable" - it will take two pieces of input

1. A "program" - i.e. a specification of a TM M
2. Input to M - encoded in some way (compatible w/ the specification in 1)

and output whatever M would output on input 2.

We need a way of giving specifications on TMs using a finite tape alphabet.

(This is annoying, b/c the size of alphabet of M can be ~~as big as~~ arbitrarily big, but the size of the alphabet of U has to be prescribed in advance)

We'll assume states of M are

State 1, State 2, ...

letters of M (tape alphabet) are

Letter 1, Letter 2, ...

Start

blank

We still need to specify - when we halt, which states are final states, and the transitions

---

Tape alphabet for U:

Specify states/letters in unary

$\Gamma_U = \{\square, 1, S, Y, N\}$

$(q_1, \ell_3) \longrightarrow (q_3, \ell_1, L)$

transition in M

$L = 1$

$R = 2$

S (1 S 111 S 111 S 1 S 1) S (next transition)

how this transition is encoded in the input for U

we also need to encode input:

if $w = l_3 l_2 l_1 l_3$, we encode as

111S11S1$111

I hope you've been convinced we can encode input to U appropriately.

How does U work?

- We'll let U have 3 tapes (w/ 3 independent heads)

Tape 1 - holds the input program (and never changes)

Tape 2 - holds the current tape of M, (encoded)

Tape 3 - holds the current state M is in (as a unary number)

Step 0: Read input, write appropriate things on tapes 1, 2, 3.

Repeat:

~~Tries to~~ Matches what is on tape 3 and ~~it~~ what is under the head on tape 2 w/ the correct transition in tape 1.

Once it finds the correct transition, it a) writes new state on tape 3

b) modifies tape 2

(we need a ~~4th tape~~ or a special tape symbol to keep track of where the head should go back to on tape 2)

(but we still need tape 4 or extra space on tape 3 to remember how many spaces we are moving tape 2 by.)