Turing Machine Variants

We saw some examples of languages (regular, context-free, non-context-free) for which we constructed Turing machines recognizing them

In fact, they decided them!

Our current model of a TM is a <u>DFA</u> with an infinite two-way tape

For finite-state automata

- Adding non-determinism made no difference ($DFA \equiv NFA$)
- Adding extra memory did make a difference ($DFA \neq PDA$)
- Accessing the memory differently did make a difference ($DFA + \text{stack} \neq DFA + \text{queue}$)

What are the answers to similar questions for TMs?

What functionality can one add to our existing definition of TMs?

→ Adding a stay operation for the tape head

So, $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$, where

$$\delta : Q \times \Gamma \to Q \times \Gamma \times \{L, R, S\}$$

Does this accord us extra computational power?

$\delta_1 : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R\} \longrightarrow M_1$

$\delta_2 : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{L, R, S\} \longrightarrow M_2$

$\delta_2(q, a) = (q', b, S)$ $\quad M_2 = (Q, \Sigma, \Gamma, \delta, s, t, r)$

$\delta_1'(q, a) = (q_{new}, b, R)$ $\quad M_1' = (Q', \Sigma, \Gamma, \delta_1', s, t, r)$

$\delta_1'(q_{new}, c) = (q', c, L)$ $\quad c \in \Gamma \quad Q' = Q \cup \ldots$

What functionality can one add to our existing definition of TMs?

→ Adding a stay operation for the tape head:

So, $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$, where

$$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, S\}$$

Does this accord us extra computational power? No!

Replace every stay transition with
one that moves right, and one that moves back left

So, showing that two TM variants are equal in computational power requires us to show that we can simulate one with the other.

What functionality can one add to our existing definition of TMs?

→ Adding an end marker to the left, before the start of the input

So, $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$, where

$\vdash \in \Gamma$, and

for every $q \in Q$, if $(q, \vdash) \in \mathrm{dom}(\delta)$, there is a $q' \in Q$ s.t.

$$\delta(q, \vdash) = (q', \vdash, R)$$

the tape head does not move left of the end marker

Does this give us extra computational power?

What functionality can one add to our existing definition of TMs?

→ Adding an end marker to the left, before the start of the input

So, $M = (Q, \Sigma, \Gamma, \delta, s, t, r)$, where

$\vdash \in \Gamma$, and

for any $q \in Q$, there is a $q' \in Q$ and $c \in \Gamma$ s.t.

$$\delta(q, \vdash) = (q', c, R)$$

the tape head does not move left of the end marker

Does this give us extra computational power? No!

At every step, the two-way infinite tape can overwrite at most one cell to the left of the existing string on the tape

I could just shift everything one step to the right, and overwrite the now empty leftmost (not $\vdash$) cell.

How does one shift tape contents to the right?

At every step, the two-way infinite tape can overwrite
at most one cell to the left of the existing string on the tape

I could just shift everything one step to the right, and
overwrite the now empty leftmost (not $\vdash$) cell.

How does one shift tape contents to the right? From the right first!
Where is the right end of the current tape contents? Unclear!
Could have a long sequence of intermediate blanks.
Cannot stop at the first (or tenth or thousandth) blank!

At every step, the two-way infinite tape can overwrite
at most one cell to the left of the existing string on the tape
I could just shift everything one step to the right, and
overwrite the now empty leftmost (not $\vdash$) cell.

How does one shift tape contents to the right? From the right first!
Where is the right end of the current tape contents? Unclear!
Could have a long sequence of intermediate blanks.
Cannot stop at the first (or tenth or thousandth) blank!
Add a right marker.
If ω is the string on my two-way tape, replace it by $\omega\#$, where $\#$ is not in my tape alphabet (nor is it $\vdash$)
Now, scroll right on the tape till you hit $\#$, at which point you shift each letter right till you hit $\vdash$, where you do nothing.