14 Apr 2021

Turing Machines

Announcements:

1. Lecture notes for this part of the course are on Canvas. (Author: Dexter Kozen)

2. Problem Set 8 will be released Friday, due a week later.

3. You can drop your lowest homework grade. (One homework scores will be increased to perfect score. It will be the one where this increase gives greatest # additional points. Even if you didn't hand it in.)

Uncomputable problems are those that cannot be solved by any algorithm.

Ex. (1) Given a program in Java, is there an input that would cause it to run forever?

(2) Given a multi-variable polynomial with integer coefficients, can we set the variables to integer values to make it evaluate to zero?

To prove ~~$\exists$~~ an algorithm to solve a problem we need to standardize on a mathematical definition of "algorithm" that is broad enough that it's believed to capture every algorithm that will ever run on a computer.

The standard definition is embodied by the TURING MACHINE.
(invented in 1930's before computers)

A Turing machine is a finite state machine sitting on an infinite tape that stores data, with the ability to move left and right on the tape and read data.

Formally a Turing machine is specified by:

- alphabets (sets of symbols for writing data)

finite
- Σ = input alphabet
- $\Gamma \supseteq \Sigma$, working alphabet (contains all input symbols and maybe others)

$\sqcup, \vdash \in \Gamma$

$\sqcup$: blank symbol; $\vdash$: tape endpoint markers

- states

Q = finite set of states

s = starting state

t = accept state

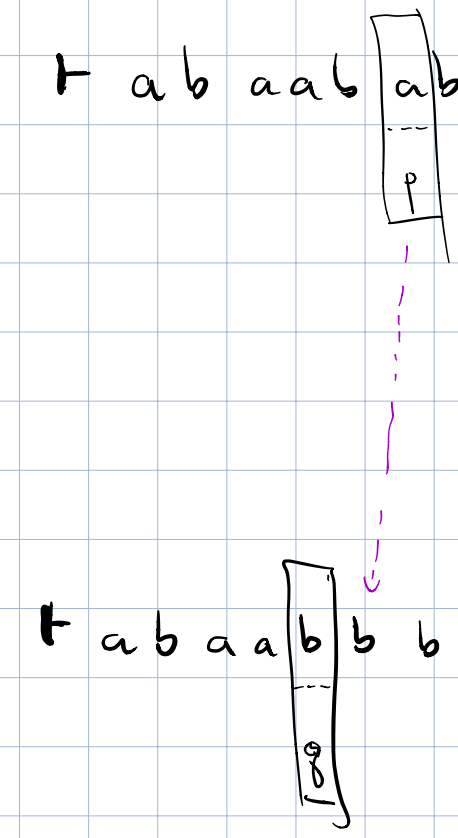
r = reject state

- transition rule, δ

  (akin to the operating system)
  tells the TM how to run
  and process data.

One processing step:

a. Read symbol in present location.
b. Overwrite with a possibly different symbol.
c. Move tape head
d. Change internal state.



That means δ defines a function

$$\delta : Q \times \Gamma \longrightarrow Q \times \Gamma \times \{-1, 0, 1\}$$

move left, stay put, move right

Key features:

"Turing complete"

- a computer has a finite amount of internal state but potentially infinite external storage.
- the computer's behavior is specified by a finite # of instructions
- these instructions (pieces of code) can be treated as data for another program
- the computer is capable of simulating itself running any other program. ("universality")

Generalization to multiple tapes.

A multi-tape Turing machine (MTM) has one set of states but it has a finite number (t) of tapes. Each has a read-write head that moves independently of the others.

That means

t-tuple of symbols being read on each tape

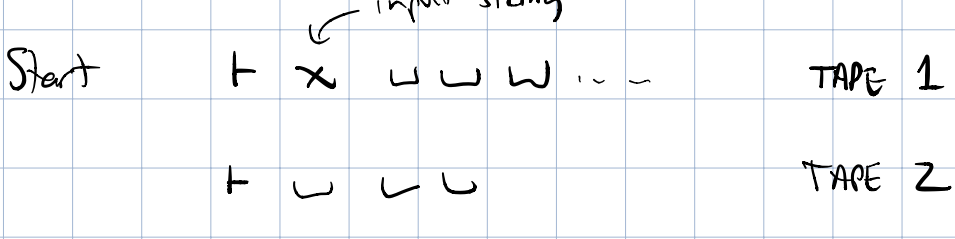
$$\delta : Q \times \Gamma^t \longrightarrow Q \times \Gamma^t \times \{-1, 0, 1\}^t$$

- Q: new state
- Γ^t: symbol to write on each tape
- $\{-1, 0, 1\}^t$: direction to move on each tape.

A single-tape TM can simulate a MTM. (See Kozen's notes.)

Examples.

(1) Test if input string is a palindrome. (Reads the same backwards as forwards.)



Step 1. Copy x to tape 2.

Copying state, C.

In state C, both heads move right.

Tape 1 overwrites what it reads.

Tape 2 writes what is on tape 1.

End when read ␣ on tape 1.

Step 2. Return tape 1 to the first symbol of x.

Returning state, ℓ.

In state ℓ, tape 1 moves left, tape 2 stays put.

Symbols aren't changed.

Ends when tape 1 reads $\vdash$.

Step 3. Compare x with backwards x.

Comparing state, e.

In state e, Tape 1 moves R, Tape 2 moves L.

If symbols mismatch, transition to reject state r

If they match, remain in e.

Accept (state t) when we reach $\vdash$ on tape 2.

Assume $\Sigma = \{0,1\}$ $\Gamma = \{0, 1, \vdash, \sqcup\}$

State \ Symbols	$\begin{pmatrix} \vdash \\ \vdash \end{pmatrix}$	$\begin{pmatrix} \supset \\ \sqcup \end{pmatrix}$	$\begin{pmatrix} 1 \\ \sqcup \end{pmatrix}$	$\begin{pmatrix} \sqcup \\ \sqcup \end{pmatrix}$	$\begin{pmatrix} 0 \\ 0 \end{pmatrix}$
s	$(c, \begin{smallmatrix} \vdash \\ \vdash \end{smallmatrix}, \begin{smallmatrix} +1 \\ +1 \end{smallmatrix})$				
c		$(c, \begin{smallmatrix} 0 \\ 0 \end{smallmatrix}, \begin{smallmatrix} +1 \\ +1 \end{smallmatrix})$	$(c, \begin{smallmatrix} 1 \\ 1 \end{smallmatrix}, \begin{smallmatrix} +1 \\ +1 \end{smallmatrix})$	$(\ell, \begin{smallmatrix} \sqcup \\ \sqcup \end{smallmatrix}, \begin{smallmatrix} -1 \\ -1 \end{smallmatrix})$	
ℓ					$(\ell, \begin{smallmatrix} 0 \\ 0 \end{smallmatrix}, \begin{smallmatrix} -1 \\ 0 \end{smallmatrix})$
e					

More examples in Kozen's notes.