16 April 2021 Universal Turing Machine

Announcements

1. Problem Set 8 will be due Thurs, Apr. 29.
2. Wellness Days Fri, 4/23 + Mon, 4/26. No class those days.
3. Homework drop policy: one of your HW scores will be increased to perfect (unless AI deduction), we'll choose the HW where this increase is biggest.
4. It's safe to talk about Prelim 2 with classmates now.

Recap: Turing machine (multi-tape) has:

- finite alphabets of symbols $\Sigma \subset \Gamma$ (Σ: input, Γ: work)

  $\sqcup$, $\vdash$ are symbols in $\Gamma \setminus \Sigma$.
- finite set of states Q
- finite set of infinite tapes for storing symbols.
- transition function

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

The MTM operates as follows.

- initialize in state $s \in Q$
  each read/write head at left end
  of ~~one~~ of the t tapes,
  reading $\vdash$.
  first tape contains input, x.
  (contents: $\vdash x \sqcup \sqcup \sqcup \ldots$)
  All other tapes $\vdash \sqcup \sqcup \sqcup \ldots$
- each step: read t-tuple of symbols
  at current positions.
  Consult δ to get
  - new state
  - symbol to write on current
    position on each tape
  - direction to move each head.
- State set Q has two special states,
  r and a.
  If machine ever reaches...
  state a: halt and accept input x
  state r: halt and reject x.

A MTM is called <u>total</u> if it always
either accepts or rejects its input.
The other alternative is it could run forever ("loop").

If M is a Turing machine,

$$L(M) = \{ x \in \Sigma^* \mid M \text{ accepts } x \}$$

A subset $L \subseteq \Sigma^*$ is called:

1. Recursively enumerable (r.e.) if $\exists$ Turing machine M such that $L = L(M)$.

2. Recursive if $\exists$ Turing machine M which is <u>total</u> and $L = L(M)$.

In other words:

- L is recursive means $\exists$ an algorithm that takes $x \in \Sigma^*$, is guaranteed to run in finite time, and tells whether $x \in L$.
- L is re. means $\exists$ an algorithm that will eventually say $x \in L$ if that's true, will never say $x \in L$ if that's false, but may run forever and say nothing.

<u>Example</u>: Halting problem. Given a program (e.g. in Java) and an input, decide if the program terminates on that input.

Pseudocode for Turing machines

[4820 Spring 2021 conventions, not standardized math.]

A piece of pseudocode representing a TM may use a finite number of variables, each of which is either:

1. A single element of Γ
2. A non-negative integer.

Control flow: allows if-then-else, while loops, repeat-until, for i in range(a,b), for $i = 0,1,2,\ldots\ldots$ (infinite loop).

Conditionals: test integers for $=$, $<$, $>$
test Γ elements for $=$

Assignment statements: allowed

Arithmetic: modify integers by $+1$, -1.

Reading data: can access symbol at current position on any tape, or move a ~~tape~~ head left/right.

Calling functions: allowed but only bounded-depth stack.
Arrays disallowed.
Pointers disallowed.

```
PRIME_LENGTH(x):   // test if length of x is prime
    // One tape
    Move right
    if reading ⊔:      // x has length ∅
        reject x
    Move right.
    if reading ⊔:      // x has length 1
        reject.
    //  main loop: test for divisors.
    length = 0
    repeat {Move right} until reading ⊔
    Move left
    repeat {
        Move left
        length = length + 1
    } until reading ⊢
    move right.
```

strlen function

Compute length of x

```
    for d = 2, 3, ..., length-1:
        count1 = 0
        count2 = 0
        while count1 < length:
            count1 = count1 + 1
            count2 = count2 + 1
```

mod(length, d)

Compute length % d

if count2 = d:
    count2 = 0
// count1 = length, count2 = length mod d
if count2 = 0:
    reject x
// reached end of for-loop, no divisor found
accept x.

Interpreting pseudocode as a MTM.

State set: $Q = \{\text{program lines}\} \times \Gamma^v$
where v denotes number of Γ-valued variables in the program.

Tapes: Any tapes mentioned in the program + one additional tape for each integer variable.
The int variables are stored in unary. (k stored as string 1^k.)