21-1/

Turing Machines

- Acceptors = (may diverge) = Σ_1
- Deciders = (always halt) = Σ_0

union ($A \in X \wedge B \in X \rightarrow A \cup B \in X$)?

or Σ_0 : On input w, run A(w) if accs, then ACC

run B(w) ''

reject

Σ_1 : interleave A(w) and B(w)

or non-det choose A(w) or B(w)

---

intersect Σ_0 : if (Aw) then B(w) o w reject

Σ_1 : same strategy

---

complement (if $\underbrace{x \in A}_{\text{x is accepted}}$ then $\underbrace{x \notin A^c}_{\text{x is rejected or diverges}}$)

Σ_0 : return $\neg A(w)$

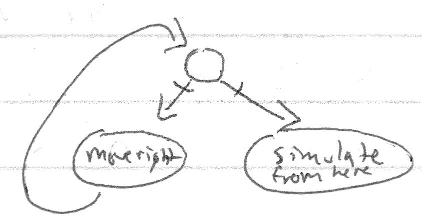
~~Σ_1 : not possible~~

---

concat: ($w = xy \in A \circ B$ if $x \in A$ and $y \in B$)

Σ_0 : $\underbrace{\text{guess}}_{\text{how many? } |w|+1}$ the division between x and y, run A(x) and B(y) until both accept

reject

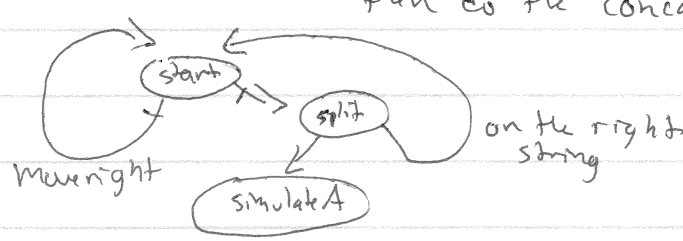


Σ_1 : totally fine

Kleene Star: $w \in A^*$ if $w = w_0 \ldots w_n$ where $w_i \in A$

Σ_0 : $n \in [1, |w|]$, guess the number of divisions then do the concat

Σ_1 : works



21-2/ In 1900, David Hilbert chair of International Congress of Mathematicians

"We must devise an algorithm that tests whether a polynomial has integral roots"

assignment of variables, σ s.t. poly / σ = 0

intuition

every term mentions some vars and multiplies coefficients them

term

$6xz + 7yx = 92$

variables

he assumed it was possible

Σ_1 solution = for $n = 0, 1, 2, 3, \ldots$

$\{ p \mid$ p is a poly over x & y and has an integral root $\}$

for $x = -n, +n$
for $y = -n, +n$
if $poly(x, y) = 0$,
accept
ow
continue

Σ_0 solution = has to figure out a maximum n to stop at

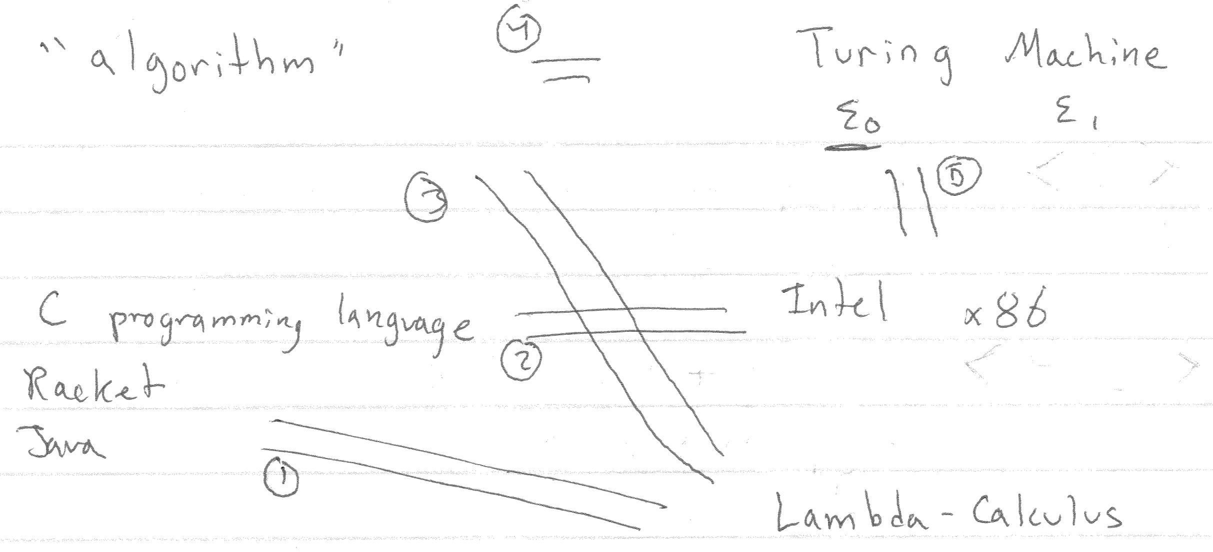
If p only mentions one variable x, that is

$$p(x) = \sum_{i=0}^{m} c_i x^i \quad (\text{order } m)$$

then $\max n = m \times \frac{c_{max}}{c_0}$ where

$c_{max} = \max(c_0, \ldots, c_m)$

If a poly mentions more than 1, Matijasevič's theorem proves no bound exists



Church - Turing Thesis
(invented L.C) (invented TM)

(1) real languages are λ-calculus
 └ we <u>can</u> prove this (Racket, ML, Haskell, Javalite)

(2) programming languages can be faithfully compiled to real hardware
 └ we <u>can</u> prove this (Compcert, CakeML, VLisp)

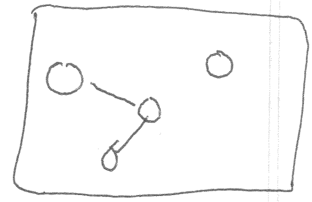
(5) Turing machines and real hardware are the same
 └ we know this <u>false</u> (no infinite tape)
 └ on accepting/rejecting (Σ_0) there's a fixed tape
  └ then imagine

(3) + (4) "algorithm" means Turing Machine or λ-calculus
 → Church-Turing Thesis ←

<u>cannot</u> be proved
must be taken on faith

21-4

Check if a graph is connected



=

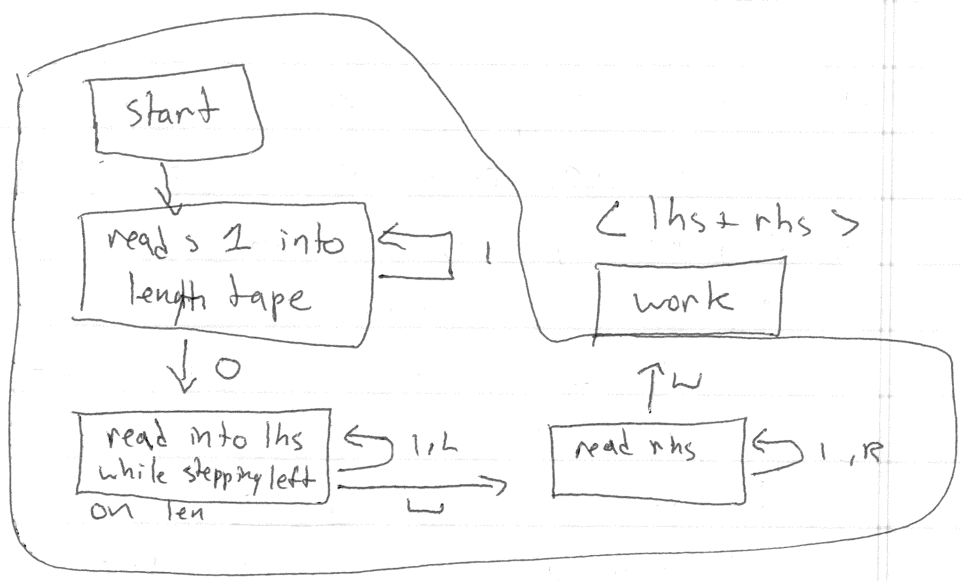
$\langle ? \rangle$ encoding bar turns a math structure into Σ^*

$\langle 3+4 \rangle = 011 + 100 \qquad \Sigma = \{0, 1, +\}$

$= \underbrace{0}_{\text{num}}\, 011 \underbrace{1}_{+} 0100$

$= \underbrace{111}_{\substack{\text{how many}\\ \text{bits}}} \underbrace{0}_{\substack{\text{bits}\\ \text{done}}}\ \underbrace{011}_{\text{lhs}}\ \underbrace{100}_{\text{rhs}} \qquad \Sigma = \{0, 1\}$

decoder



$\{ \langle G \rangle \mid G$ is a connected graph $\}$

- do BFS mark nodes
- if at end, note unmarked $\Rightarrow$ reject