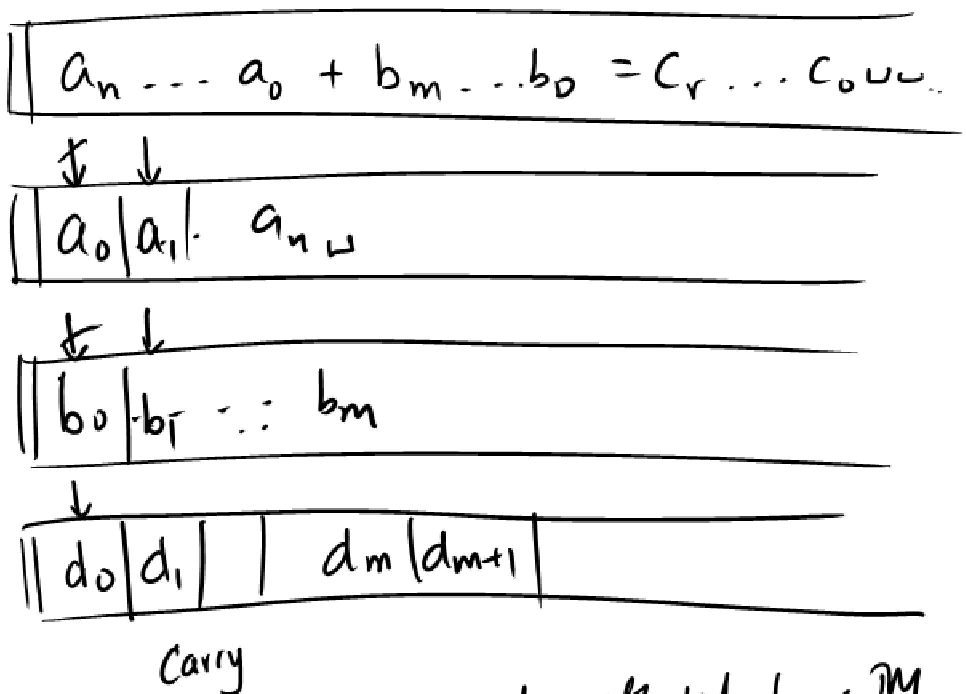
More decidable problems,
encodings, simulating "real" computers,
Recognizability vs decidability

## Arithmetic

$$L = \{ a_n \ldots a_1 + b_m \ldots b_0 = c_r \ldots c_0 \mid a_i, b_j, c_k \in \{0, \ldots 9\}, \llbracket c_r \ldots c_0 \rrbracket = \llbracket a_n \ldots a_0 \rrbracket + \llbracket b_m \ldots b_0 \rrbracket \}$$

$$\Sigma = \{0, \ldots 9, +, =\}$$

$$\llbracket a_n \ldots a_0 \rrbracket = \sum_{i=0}^{n} a_i \cdot 10^i$$



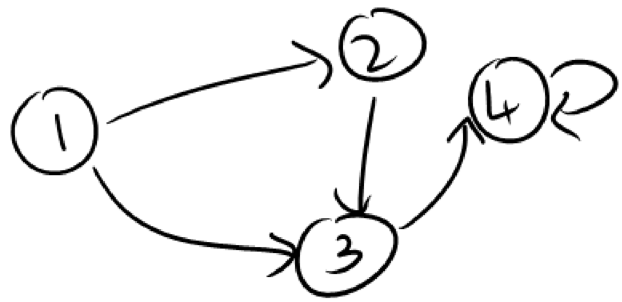

All arithmetic ops can be effected by a TM.

$\langle G=(V,E)\ ,\ s,\ t\rangle$

$E \subseteq V \times V\ ,\ s, t \in V$


$n, m, (n_1, n_1'), (n_2, n_2'), \dots (n_m, n_m')$

$L = \{ \langle n\rangle, \langle m\rangle, \langle n_1, n_1'\rangle \dots \langle n_m, n_m'\rangle, n_s, n_t \mid$ Graph rep. by this string has $t$ reachable from $s \}$




"4, 5, (1, 2), (2, 3) (1, 3), (3, 4), (4, 4),
1, 4"

Decision problem $\xrightarrow{\text{Encoding}}$ Language

"As far as the encoding is 'natural' the language being TM-decidable or not will not change"

Encodings don't matter for TM-dec/
TM-recog.

But they do matter

- DFA
- Resource-bounded TM.

Problem. Given $\langle x \rangle$, find a prime factor of $x$

$\langle x \rangle$ - binary — no one knows ∄ a P-TIME m/c.

$\langle x \rangle$ - unary $(1^{|x|})$ - ~~but~~ we can build a PTIME TM.

$$E_{DFA} = \{\langle M\rangle \mid M \text{ is a DFA and } L(M) = \emptyset\}$$

Encode DFAs as strings.

$$A = (Q, \Sigma, \delta, q_0, F)$$

Encode $\langle Q\rangle, \langle \Sigma\rangle, \langle q_0, \langle F\rangle$

| | $a$ |
|---|---|
| $q$ | $q'$ |

TM can decide $E_{DFA}$ (it does not matter how you encode DFAs)

# s-t connectivity

$n, m, \langle n_1, n_1' \rangle \ldots \langle n_m, n_m' \rangle, s, t$

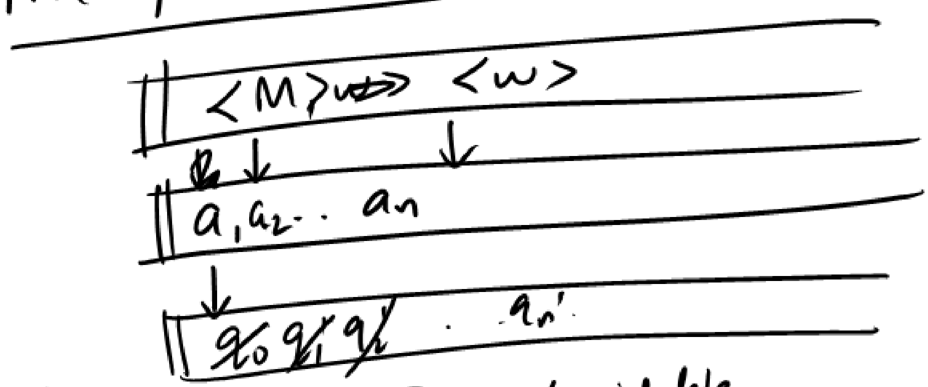
Edge-list $\langle n_1, n_1' \rangle \ldots \ldots \langle n_m, n_m' \rangle$

Done list $s, u_1, u_2, u_3$

To-do list $\cancel{s}, \cancel{u_1}, \cancel{u_2}, \cancel{u_3}, v_1, v_2, v_3, \ldots v_5, v_6 \ldots w_8 \ldots$

$A_{DFA} = \{ \langle M, w \rangle \mid M$ is a DFA, $w$ is an input for $M$, $M$ acc $w \}$

TM for $A_{DFA}$.




$A_{DFA}$ is TM-decidable.

$$EQ_{DFA} = \{ \langle M_1, M_2 \rangle \mid L(M_1) = L(M_2) \}$$

$- \ L(M_1) \subseteq L(M_2)$

$$L(M_1) \cap \overline{L(M_2)} = \phi$$

$- \ L(M_2 \subseteq L(M_1)$

$EQ_{DFA}$ is TM-decidable.

$A_{NFA} = \{ \langle M, w \rangle \mid$ M accepts w, M is an NFA $\}$.

I. Convert M to a DFA M'.
~~Check if~~ Simulate M' on w
And accept iff M' acc w.

II. TM could simulate M on w,
but at each prefix of w,
keep track of the set of states
the NFA could be in.

$A_{regex} = \{ \langle R, w \rangle \mid R$ is a regexp and $w \in L(R) \}$

TM deciding $A_{regex}$ -

- Convert $R$ to a DFA $M$
- Check if $M$ accepts $w$.

## TMs can simulate "real" computers.


RAM $r_1, \ldots r_K$

LOAD $r_1, r_2$

STORE $r_1, r_2$

JMP $r_1$

⋮

## TM simulating a RAM-machine

$(a_1, d_1), (a_2, d_2) \ldots\ldots (a_n, d_n)$

$(5, 123), (6, 124), (9, 12), \ldots\ldots$

$r_1 = 9$

LOAD $q_0\, r_1, r_2$

$r_2 := 12$

Real comp: Registers

\+ Instructions

\+ RAM-memory

\+ control.

\+ arithmetic (ALU)

Theorem.

$L$ is TM-decidabele

iff $L$ and $\overline{L}$ are TM-recognizable.

---

($\Rightarrow$) $L$ is decidable, $L$ is recognizable.

Since $L$ is decidable, $\overline{L}$ is decidable.

So $\overline{L}$ is also recognizable.

($\Leftarrow$) $M_1$ recog. $L$

$M_2$ recog $\overline{L}$

<u>Decider for M:</u>

Simulate $M_1$ & $M_2$ on $w$
in "parallel"
Accept if $M_1$ halts & accepts
Reject if $M_2$ halts & accepts.