COMPUTABILITY

- can we use computers to solve any problem?

PROBLEMS - decision : YES/NO

- input finite strings, finite alphabet

- Ex: membership

$$L \subseteq \Sigma^*$$

set of strings where answer is YES

Q: given $w \in \Sigma^*$ $w \in L$? (yes or no)

Σ^n - all strings of length n formed using Σ

ex: $\{0,1\}^n$ all binary n-bit strings

ex $\Sigma = \{a, b\}$

$L1 = \{w \in \{a\}^*, w \in \{b\}^*\}$

language of words that only contain a or only contain b.

- EX 3SAT

$$\Sigma = \{\wedge, \vee, x_1 \ldots x_n, \bar{x}_1 \ldots \bar{x}_n\}$$

YES/NO - Does it have a satisfying assignment

- EX Clique

Graph $= (V, E)$ $|V| = m$ $|E| = \binom{m}{2} = \frac{m(m-1)}{2}$

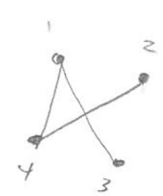
$\Sigma = \{0,1\}$

represent any graph of m vertices as an 1/0 string of length $\binom{m}{2}$

no edge

0 1 0 0 1 0 1 0 1

edge



12	13	14	23	24	34
0	1	1	0	1	0

YES/NO: does it have a clique of size k?

COMPUTER - every computer

- by an algorithm

TURING - formalized notion of algorithms

- turing machines

Σ : finite set of symbols

tape : infinite

head : looks at a symbol on tape

- change symbol
- move L / R | STAY
- change state

$\Sigma = \{a, b, c, *, \dots\}$

Q : finite set of states

Transition Tables : head action given pair :

# of lines in transition tables

$|\Sigma| * |Q|$

(symbol , state)

↓

(symbol , state, → right / ← left / – stay)

$\Sigma = \{a, b, \sqcup\}$

STATES : S0, S1, S2, SYES, SNO

(a, S0) → (a, S1, →)
(b, S0) → (b, S2, →)
(a, S1) → (a, S1, →)
(b, S2) → (b, S2, →)
(a, S2) → (a, SNO, –)
(b, S1) → (b, SNO, –)
($\sqcup$, S0) → ($\sqcup$, SYES, –)
($\sqcup$, S1) → ($\sqcup$, SYES, –)
($\sqcup$, S2) → ($\sqcup$, SYES, –)

COMPUTABILITY:

define "problems" to be YES/NO questions on finite strings.

let $\Sigma = \{a, b, c\}$ (alphabet of interest) (for example).

$\Sigma^* :=$ set of all finite-length strings on the alphabet Σ

→ aa, abab, abcc, $\underbrace{ccc \ldots ccc}_{99\ c\text{'s}}$ are all strings in Σ^*

$\Sigma^n :=$ all length-n strings on the alphabet Σ

→ $\Sigma^2 = \{aa, ab, ba, bb, bc, cb, ca, ac, cc\}$ (for example)

$$\Sigma^* = \bigcup_{n \geq 1} \Sigma^n$$

i.e. set of all finite length strings on Σ = set of all 1 length-strings $\cup$
set " " 2 " " $\cup$
3
⋮

Notation:
" := " = "defined to be"

EXAMPLE

$\Sigma = \{a, b\}$

$L1 = \{\omega \in \{a\}^*, \omega \in \{b\}^*\}$ = all strings of finite length of the form aaa...a or bb...b.

Subsets of Σ^* are called languages.

L1 is a language on Σ, and a subset of Σ^*.

All <u>decision problems</u> can be viewed as problems of the kind "does string x belong to language L over Σ?"
alphabet

<u>3SAT (Problem)</u>:

variables: $x_1, x_2, \ldots, x_n$
negated vars: $\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}$ — literals

3-CNF formula: $\underbrace{(x_1 \overset{\text{"OR"}}{\vee} x_2 \overset{\text{"OR"}}{\vee} \overline{x_3})}_{\text{3 literals}} \overset{\text{"AND"}}{\wedge} \underbrace{(x_4 \vee x_2 \vee \overline{x_1})}_{\text{3 literals}} \cdots$

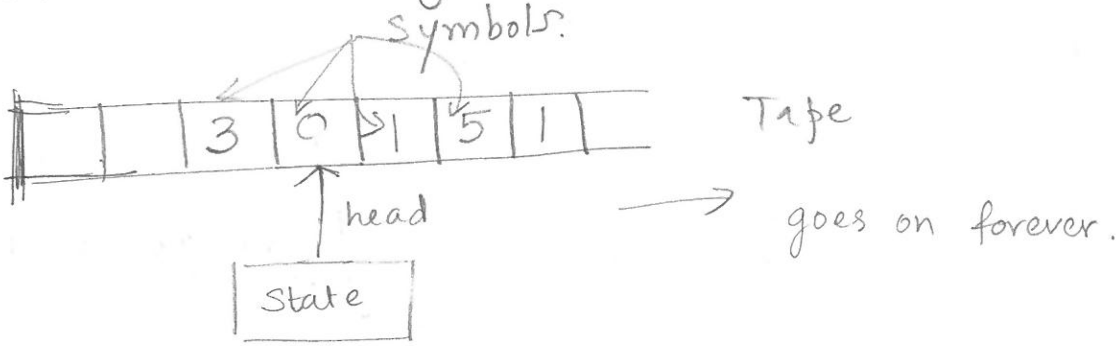
$\boxed{\text{Conjunctive normal form}}$

3SAT: Given a 3-CNF formula, does there exist an assignment of TRUE / FALSE to variables so that the 3-CNF formula evaluates to true?

— 3SAT has no known polytime algorithm. (NP-complete).

3SAT can be viewed as checking if a finite length string over $\Sigma = \{\vee, \wedge, x_1, \ldots, x_n, \overline{x_1}, \ldots, \overline{x_n}\}$ belongs to a language L = set of 3-CNF formulas that have a satisfying assignment.

All algorithms used to solve decision problems can be viewed as a "turing machine".



head can move left/right/stay.

can change symbol of box which head points to.

can change state

symbols come from a specified alphabet Σ.

A turing machine is specified by the alphabet Σ (finite), states Q (finite) and a transition table.

The transition table tells you what to do in a particular situation:

(symbol, state) $\longrightarrow$ (symbol, state, "move left" OR "move right" OR "stay")

new symbol you replace old symbol with

new state you are in.

The table has $|\Sigma||Q|$ lines.

Notation:

"To decide a language L" means to check if a given string x belongs to L

Ex. $L = \{ w \in \{a\}^* \text{ or } w \in \{b\}^* \}$

$\Sigma = \{a, b\}$

Q:
Write a turing machine that decides L.
— i.e. given a string x your turing machine decides if x has all a's or all b's.

Answer:

$\Sigma = \{ a, b, \triangleright, \sqcup \}$ # alphabets

$Q = \{ s_0, s_1, s_2, \text{ACCEPT}, \text{REJECT} \}$ # States

$(a, s_0) \longrightarrow (a, s_1, \text{"move right"})$

$(b, s_0) \longrightarrow (b, s_2, \text{"move right"})$
$(a, s_1) \longrightarrow (a, s_1, \text{"move right"})$
$(b, s_2) \longrightarrow (b, s_2, \text{"move right"})$
$(b, s_1) \longrightarrow (b, \text{"REJECT"}, \text{stay})$
$(a, s_2) \longrightarrow (a, \text{"REJECT"}, \text{stay})$

$(\sqcup, s_0) \longrightarrow (\sqcup, \text{"ACCEPT"}, \text{stay})$

$(\sqcup, s_1) \longrightarrow (\sqcup, \text{"ACCEPT"}, \text{stay})$.

$(\triangleright, s_0) \longrightarrow (\triangleright, s_0, \text{"move right"})$