

Turing-Recognizability & Mapping Reducibility

Cor. If $A \leq_m B$ and A not Turing-recognizable, then

B is not Turing-recognizable.

Proof: Suppose B were T. R. By the previous th.,
since $A \leq_m B$, then A would be T.R. Contradiction.

Th. EQ_{TM} is not Turing-recognizable.

Proof: We show that $A_{TM} \leq_m \overline{EQ}_{TM}$

F: on input $\langle M, w \rangle$ construct Rej and M2 so
that $L(Rej) = \emptyset$

M2: on input x, run M on w. If M accepts, accept.
 $L(M2)$ is either Σ^* (if M accepts w) or $\emptyset$ (ow)

$M \text{ accepts } w \iff \emptyset = L(Rej) \neq L(M2)$

1

Turing-Recognizability & Mapping Reducibility

Th. $\overline{EQ}_{TM}$ is not Turing-recognizable.

Proof: We show that $A_{TM} \leq_m EQ_{TM}$

Want $\langle M, w \rangle \rightarrow \langle M1, M2 \rangle$

$M \text{ accepts } w \iff L(M1) = L(M2)$

G: on input $\langle M, w \rangle$ construct Acc and M2 so
that $L(Acc) = \Sigma^*$

M2: on input x, run M on w. If M accepts, accept.
 $L(M2)$ is either Σ^* (if M accepts w) or $\emptyset$ (ow)

$M \text{ accepts } w \iff \Sigma^* = L(Acc) = L(M2)$

EQ_{TM} and $\overline{EQ}_{TM}$ are both Turing-unrecognizable

2

Complexity

Decidability tells us answer (eventually)

but gives no idea of how long to compute!

Complexity: measure resources (time, space, power,...) to solve problem

We concentrate on $\text{TIME}(n)$ = # of steps, as function of length n of input string

Decidability

What can (conceptually) be solved, given unlimited resources?

Complexity

What can (practically) be solved, given limited resources?

3

Example: Travelling Salesman Problem

Collection $\{C_1, C_2, \dots, C_n\}$ of cities, distance k

Distance matrix $D(i,j)$ gives distance (in miles) between C_i and C_j .

Problem: Is there a tour of all cities with distance less than k ?

Solvable problem:

Can consider every possible tour (wo repet.)

For each tour, figure out total distance

Stop if find one of distance less than k .

Complexity:

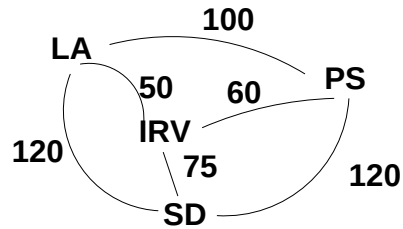
n cities, $(n-1)!$ choices for tours

$n = 50, 49! \sim 10^{45}$

not practically computable!

4

Example: Travelling Salesman Problem



SD

SD

5

Time Complexity

Def. Let M be a deterministic TM that always halts.

The *running time or time complexity* of M is the function $f : \mathbb{N} \rightarrow \mathbb{N}$ where $f(n)$ is the maximum no. of steps of the TM M on any input of length n .

We say that M runs in time $f(n)$.

Ex. $\{0^n 1^n \mid n \geq 0\}$

M on input w :

1. Scan the tape and reject if 0 to right of 1.
2. Repeat the following while both 0's and 1's:
 - a. Scan across tape, crossing out one 0 and one 1.
3. If 0's remain and all 1's crossed out, reject.
If 1's remain and all 0's crossed out, reject.
OW accept.

6

How many steps required?

1. Scan the tape and reject if 0 to right of 1.

$O(n)$ 000 0 1111..11

1 scan of tape: n steps to determine right form; n steps to move head to left = $2n$ steps

2. Repeat the following while both 0's and 1's:

n

$O(n^2)$ a. Scan across tape, crossing out one 0 and one 1.

$n/2$ scans, n steps each: $n^2/2$ steps

3. If 0's remain and all 1's crossed out, reject.

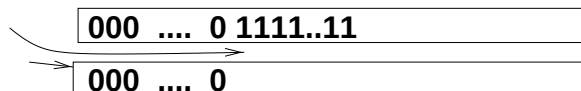
$O(n)$ If 1's remain and all 0's crossed out, reject.
OW accept.

1 scan of tape: n steps.

Total Time: $O(n) + O(n^2) + O(n) = O(n^2)$

7

What about two tapes?



$O(n)$ 1. scan across tape, reject if 0 to rt. of 1 at same time, write 0's on second tape.

2. Scan across the 1's on Tape 1 till end.

? For each 1 on Tape 1, cross off 0 on Tape 2.
if all 0's are crossed off before all 1's read, reject!

3. If 0's remain and all 1's crossed out, reject.

$O(n)$ If 1's remain and all 0's crossed out, reject.
OW accept.

Total Time: $O(n) + O(n) + O(n) = O(n)$

8

Complexity Models: Resources Matter

Linear time in single tape model

Quadratic time in two-tape model

Turns out time difference:

Det. single tape

$$O(t(n)^2)$$

Deterministic

$$O(2^{t(n)})$$

Det. multitape

$$O(t(n))$$

Nondeterministic

$$O(t(n))$$

All deterministic models differ by polynomial time

Exponential time difference in det. vs nondet.

Polynomial difference: small Exponential: large

9

The Class P

Def. P is the class of languages decidable on a deterministic, single tape TM in time bounded by some polynomial (in size of input)

P contains all problems that are realistically solvable

P turns out to be invariant over many models of computation

May still have problems in P we don't think are

realistically solvable (e.g. $O(n^v)$ where v is very large no.)

Gives us fundamental way of grouping problems:

P

not in P

10

The Class NP

*Many problems have as their only solution brute-force search
May not be solvable in polynomial time*

**Def. NP is the class of languages decidable on
a nondeterministic, single tape TM in time bounded
by some polynomial (in size of input)**

*Ex. Travelling Salesman problem: solution < k miles
Only known solution is to try all tours- exponential
time for deterministic TM*
**Nondeterministic TM can guess the tour, then
check if its length is less than k in polynomial
time!**