Computabilty and Complexity Theory

Turing Machines has finitely many control states and
- One read only input tape
- One write only output tape
- Finitely many read-write work tapes

One a given input a Turing Machine may
(a) (Halt and) accept
(b) Halt but not accept } not accepts
(c) Not halt.

For a Turing Machine M,

$$L(M) = \{ w \mid M \text{ accepts } w \}$$

$$f_M(x) = \begin{cases} \text{String } w \text{ written on output tape} & \text{if } M \text{ halts on } x \\ \text{undefine} & \text{if } M \text{ does not halt on } x \end{cases}$$

A language/problem A is recursively enumerable/semi decidable if there is a TM M s.t. $A = L(M)$

A language/problem A is recursive/decidable if there is a TM M that halts on all inputs and $A = L(M)$.

Properties of r.e. and recursive languages

(a) If A is recursive then A is r.e.

(b) If A is recursive then $\bar{A}$ is recursive

(c) A is recursive iff A and $\bar{A}$ are r.e.

($\Leftarrow$) Dovetailing

Encoding Any (structured) object can be encoded as a binary string.

$\langle O_1, O_2 \dots O_n \rangle$ – binary string encoding $O_1, O_2 \dots O_n$

$\langle M \rangle$ – binary encoding of TM M.

M_x – TM M s.t $\langle M \rangle = x$.

Universal TM There is TM U that solves the following problem (membership)

$$L(U) = MP = \{ \langle M, w \rangle \mid M \text{ accepts } w \}$$

MP is r.e.

Theorem $\overline{K} = \{ x \mid x \notin L(M_x) \}$ is not r.e.

Proof Consider any TM M.

$\langle M \rangle \in L(M)$ iff $\langle M \rangle \notin \overline{K}$

$\langle M \rangle \in (L(M) \setminus \overline{K}) \cup (\overline{K} \setminus L(M)) \neq \emptyset$

$L(M) \neq \overline{K}$

Reduction A many one/mapping reduction from A to B is a computable (total) function $f : \Sigma \to \Sigma$ s.t.

$\forall w.\ w \in A$ iff $f(w) \in B$.

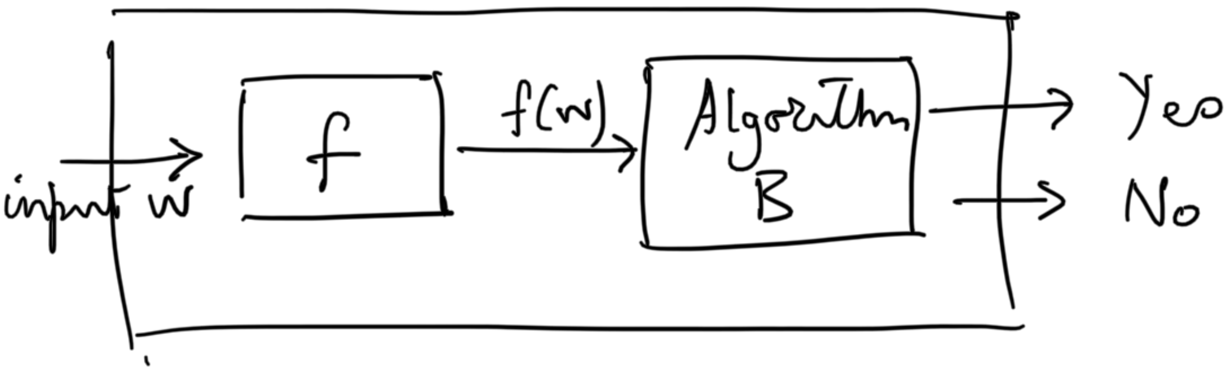
If there is reduction from A to B, denote

$A \le_m B$

Theorem If $A \le_m B$ and B is r.e (recursive) then A is r.e. (recursive).

If $A \le_m B$ and A is not r.e (undecidable) then B is not r.e (undecidable).

Proof An algorithm for A



Proposition If $A \le_m B$ and $B \le_m C$ then $A \le_m C$.

If $A \le_m B$ then $\bar{A} \le_m \bar{B}$

Proof Assume $A \le_m B$ and $B \le_m C$.

Let f and g be the reductions

Observe $f \circ g$ is a reduction from $A \le_m C$

If f is a reduction from A to B

then f is also a reduction from $\bar{A}$ to $\bar{B}$

Proposition $\overline{K} \le_m \overline{MP}$

$\overline{MP} = \{ \langle M, w \rangle \mid w \notin L(M) \}$

$\overline{K} = \{ x \mid x \notin L(M_x) \}$

$f(x) = \langle M_x, x \rangle$ is reduction from $\overline{K}$ to $\overline{MP}$

$x \in \overline{K}$ iff $x \notin L(M_x)$ iff $\langle M_x, x \rangle \in \overline{MP}$ iff $f(x) \in \overline{MP}$

Corollary Since $\overline{K}$ is not r.e. $\overline{MP}$ is not r.e.

Since MP is r.e., K is r.e.

MP is not decidable / recursive.

Hardness / Completeness

A is r.e.-hard iff $\forall B \in$ R.E. $B \leq_m A$.

A is r.e.-complete iff A is r.e-hard and A is r.e.

Theorem MP is r.e-complete.

Proof Consider A that is r.e.

Let M be TM $L(M) = A$.

Need to show $A \leq MP$.

$f(x) = \langle M, x \rangle$ is computable.

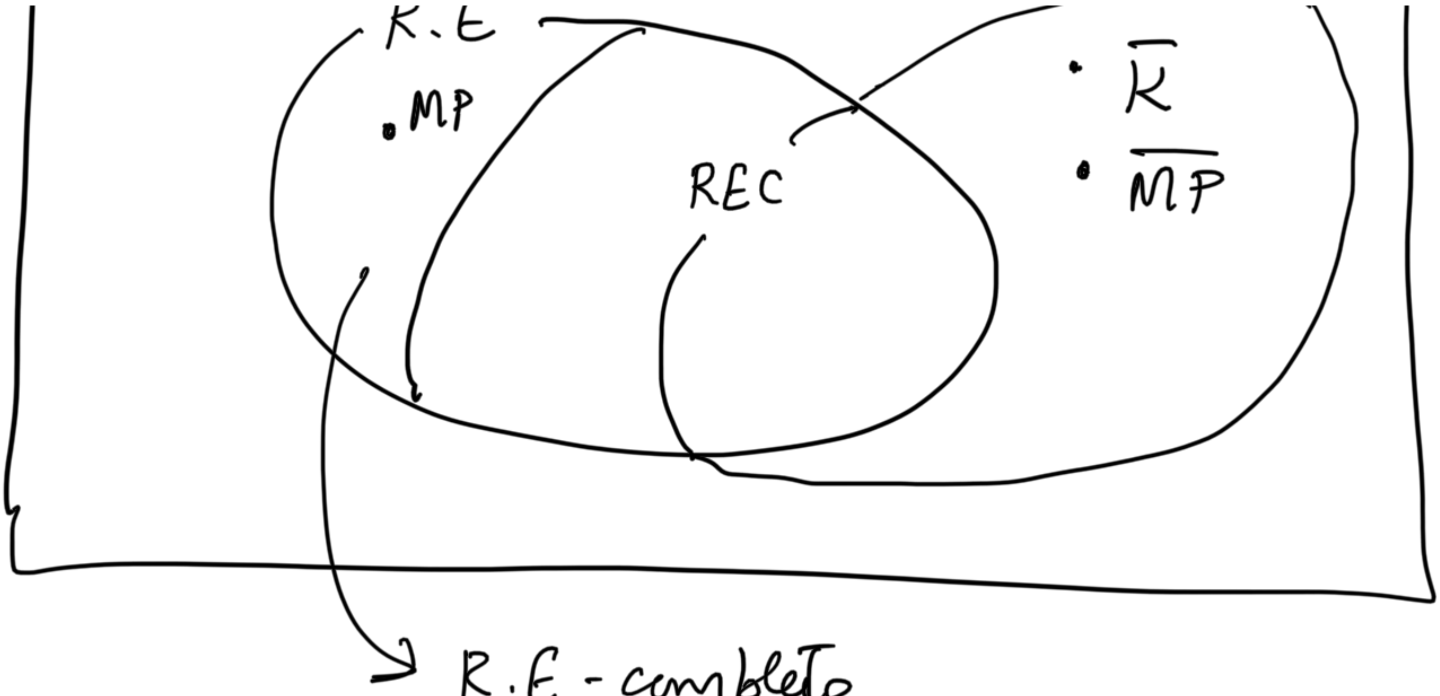
$x \in A$ iff $x \in L(M)$ iff $\langle M, x \rangle \in MP$ iff $f(x) \in MP$.

Proposition If A is r.e. hard then A is undecidable

Proof Since A is r.e-hard and $MP \in$ R.E

$MP \leq_m A$.

Since MP is undecidable, A is undecidable

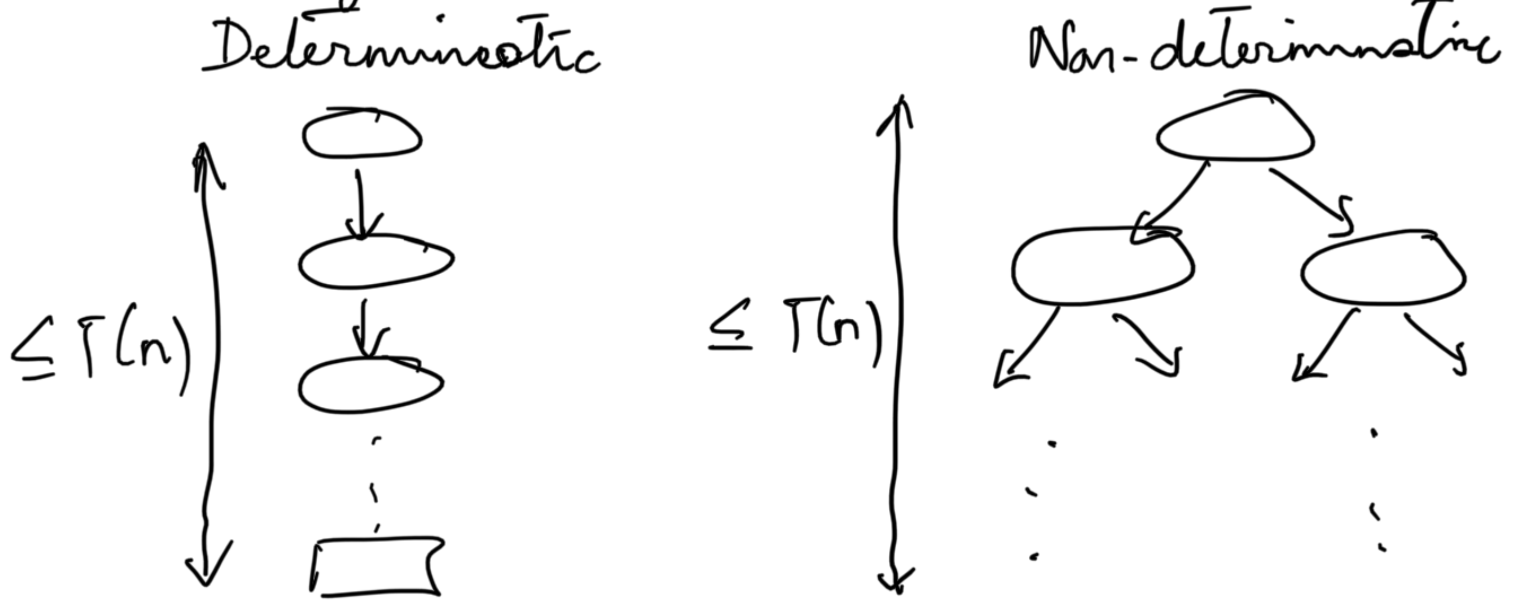


R.E-complete

$co RE = \{ A \mid \overline{A} \text{ is r.e.} \}$

Complexity Theory

Definition A (deterministic / non-deterministic) TM M runs in time $T(n)$ if for every input w of length n, all computations of M on w take at most $T(n)$ of steps.



A (deterministic / non-deterministic) TM M uses $S(n)$ space if on any input w of length n, the total number of work tape cells used by M in any computation of w is at most $S(n)$.

→ A cell is used by TM if it is either written to / read from during the computation.

→ Cells reused

$DTIME(T(n)) = \{ A \mid \exists$ DTM M s.t. M runs in $T(n)$ time & $L(M) = A \}$

$DSPACE(S(n)) = \{ A \mid \exists$ DTM M s.t. M uses $S(n)$ space and $L(M) = A \}$

$NTIME(T(n))$ and $NSPACE(S(n))$