Computability Theory: Decidability and Recognizability

- Encoding Turing Machines and the Universal TM
- Computability
- Halt: Undecidable Problems using Diagnolization
- Accept: Undecidable Problems using Diagnolization
- Turing Reductions
- Mapping Reductions
- Undecidable and Unrecognizable Problems
- Rice Theorem

Imdad ullah Khan

Mapping Reduction

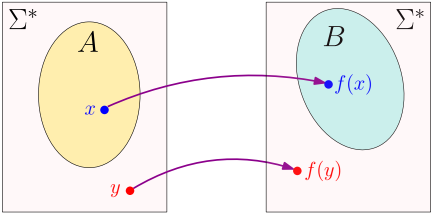
Mapping Reduction

Mapping Reduction is a general method to prove undecidability

A function $f : \Sigma^* \mapsto \Sigma^*$ is computable, if there is a Turing machine M, which on input w halts with $f(w)$ on the tape

A language $A \subset \Sigma^*$ is mapping reducible to a language $B \subset \Sigma^*$ if there is a computable function $f : \Sigma^* \mapsto \Sigma^*$ such that for every

$$\forall\, w \in \Sigma^*, \qquad w \in A \iff f(w) \in B$$



Denoted by $A \leq_m B$

A is mapping reducible to B

Note: not enough if
$w \in A \implies f(w) \in B$

Mapping Reduction

Let $A_{DFA} = \{\langle D, w\rangle : D \text{ a DFA}, w \in L(D)\}$

Let L be any regular language.

$L \ \leq_m \ A_{DFA}$

If L is a regular language, then there is a DFA D such that $L = L(D)$

The following computable function yields a mapping reduction

$$f(w) \ := \ \langle D, w\rangle$$

$w \in L \iff D \text{ accepts } w \iff \big[\langle D, w\rangle = f(w)\big] \in A_{DFA}$

Thus, f is a mapping reduction from L to A_{DFA}

Mapping Reduction

$A_{NFA} \leq_m A_{DFA}$

For every NFA N, there is a DFA D such that $L(D) = L(N)$

▷ Recall subset construction and ϵ-closure

The following function yields a mapping reduction from A_{NFA} to A_{DFA}

For $\langle N, w \rangle$, let D be the DFA such that $L(D) = L(N)$, then

$$f(\langle N, w \rangle) := \langle D, w \rangle$$

Suppose $A \leq_m B$. If B is decidable, then A is decidable

Informally, if the harder problem B is decidable, then so should be A

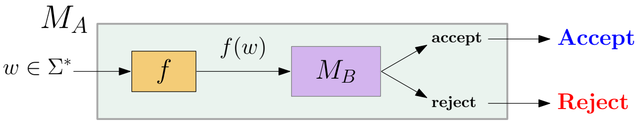
Proof: Let M_B be TM deciding B and $f : A \mapsto B$ be mapping reduction

We construct M_A to decide M_B that works as follows:

$M_B(w)$: Compute $f(w)$, Run $M_B(f(w))$ and output its answer

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

$$\implies M_B(w) \text{ accepts/rejects} \iff M_A(w) \text{ accepts/rejects}$$



TM_A maps input string w to $f(w)$, runs M_B on $f(w)$, and accept/reject accordingly

Mapping Reduction

Suppose $A \leq_m B$. If B is recognizable, then A is recognizable

Informally, if the harder problem B is recognizable, then so should be A

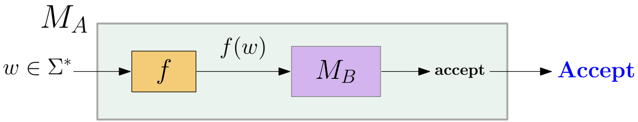
Proof: Let M_B be TM recognizing B and $f : A \mapsto B$ be mapping reduction

We construct M_A to recognize M_B that works as follows:

$M_B(w)$: Compute $f(w)$, Run $M_B(f(w)$ if it halts, then output its answer

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

$$\implies M_B(w) \text{ accepts } \iff M_A(w) \text{ accepts}$$



M_A maps input string w to $f(w)$, runs M_B on $f(w)$, and accept if M_B does

Mapping Reduction

The contrapositives of the above theorems yield negative results

Suppose $A \leq_m B$. If B is decidable, then A is decidable

Suppose $A \leq_m B$. If A is undecidable, then B is undecidable

Suppose $A \leq_m B$. If B is recognizable, then A is recognizable

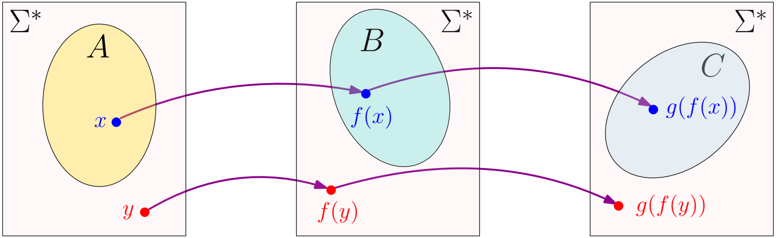
Suppose $A \leq_m B$. If A is unrecognizable, then B is unrecognizable

Mapping Reduction

We can get additional negative and positive results by transitivity

If $A \leq_m B$ and $B \leq_m C$, then $A \leq_m C$

$$w \in A \iff f(w) \in B \iff g(f(w)) \in C$$



Proving the Halting problem undecidable using mapping reduction

HALT $= \{\langle M, w\rangle : M$ halts on input $w\}$ is undecidable

Proof: Need mapping reduction from an undecidable problem to HALT

The only undecidable problem we know is A_{TM}

Define $f : A_{TM} \mapsto$ HALT as:

$f(\langle M, w\rangle) = \langle M', w\rangle$,

M' works as follows: Run M on w and accept w if M does, else loop

we have

$$\langle M, w\rangle \in A_{TM} \iff \langle M', w\rangle \in \text{HALT}$$

$\overline{\text{HALT}}$ is unrecognizable

$\overline{\text{HALT}} = \{\langle M, w\rangle : M \text{ does not halt on } w\}\}$

Proof: Need mapping reduction from an unrecognizable problem to $\overline{\text{HALT}}$

The only unrecognizable problem we know (so far) is $\overline{A_{TM}}$

The same mapping reduction $f : A_{TM} \mapsto \text{HALT}$ is also a mapping reduction from $\overline{A_{TM}}$ to $\overline{\text{HALT}}$ ▷ why?

$$\langle M, w\rangle \in A_{TM} \iff \langle M', w\rangle \in \text{HALT}$$

HALT $\leq_m$ A_{TM}

Define f : HALT $\mapsto A_{TM}$ as:

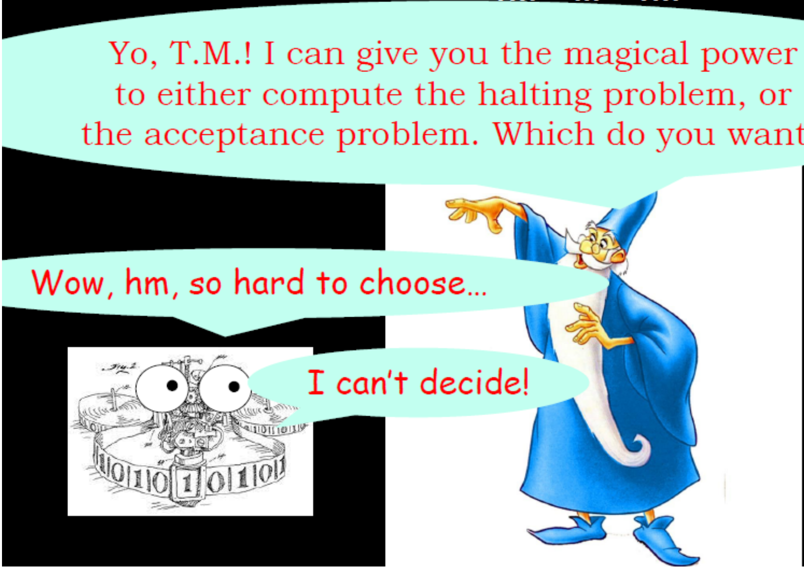
$f(\langle M, w \rangle) = \langle M', w \rangle$,

M' works as follows: Run M on w and accept w if M halts, else loop

we have

$$\langle M, w \rangle \in \text{HALT} \iff \langle M', w \rangle \in A_{TM}$$

Corollary: $HALT_{TM} \equiv_m A_{TM}$



slide credit: Ryan Williams@MIT