Mapping Reducibility

Sipser 5.3 (pages 206–210)

Computable functions

- Definition 5.17: A function \( f : \Sigma^* \rightarrow \Sigma^* \) is a **computable function** if some Turing machine \( M \), on every input \( w \), halts with just \( f(w) \) on its tape.

- Example: The increment function

\[
inc++ : \{1\}^* \rightarrow \{1\}^*
\]

is Turing-computable
Incrementing

• $\text{inc}++: \{1\}^* \rightarrow \{1\}^*$

$$\text{Finite control}$$

$$\text{Infinite tape}$$

Transforming machines

$F =$ "On input $<M>$:

1. Construct the machine
   $M_\infty =$ "On input $x$:
   1. Run $M$ on $x$.
   2. If $M$ accepts, accept.
   3. If $M$ rejects, loop."

2. Output $<M_\infty>$."
Mapping reducibility

• Definition 5.20:
  Language $A$ is mapping reducible to language $B$, written $A \leq_m B$, if there is a computable function $f: \Sigma^* \rightarrow \Sigma^*$, where for every $w$,$$
  w \in A \iff f(w) \in B
$$

Problem reduction

• Theorem 5.22:
  If $A \leq_m B$ and $B$ is decidable, then $A$ is decidable.
And... the contrapositive

- Theorem 5.22:
  If $A \leq_m B$ and $B$ is decidable,
  then $A$ is decidable.

- Corollary 5.23:
  If $A \leq_m B$ and $A$ is undecidable,
  then $B$ is undecidable.

A familiar mapping reduction...

$A_{TM} = \{<M, w> | M \text{ is a TM and } M \text{ accepts } w\}$

$HALT_{TM} = \{<M, w> | M \text{ is a TM & } M \text{ halts on input } w\}$
\[ A_{TM} \leq_m HALT_{TM} \]

\( F = \) "On input \(<M>:\)"
1. Construct the machine
   \( M_w = \) "On input \(x:\)"
      1. Run \(M\) on \(x\).
      2. If \(M\) accepts, accept.
      3. If \(M\) rejects, loop."
2. Output \(<M_w>\)."

Similarly...

• Theorem 5.28:
  If \(A \leq_m B\) and \(B\) is Turing-recognizable, then \(A\) is Turing-recognizable.

• Theorem 5.29:
  If \(A \leq_m B\) and \(A\) is not Turing-recognizable, then \(B\) is not Turing-recognizable.
**Solvable, half-solvable, hopeless**

\[ A_{TM} \subseteq A_{TM,\emptyset} \subseteq E_{TM} \]

- Turing-recognizable
- Turing-decidable
- co-Turing-recognizable

\[ EQ_{TM} = \{ <M_1, M_2> | L(M_1) = L(M_2) \} \]

is hopeless

- Theorem 5.30:
  \( EQ_{TM} \) is neither Turing-recognizable nor co-Turing-recognizable
- Proof:
  - What if we show \( A_{TM} \leq_m EQ_{TM} \)?
$A_{TM} \leq_m EQ_{TM}$

- $G = "On \text{ input } <M,w>:"
  1. Construction the following two machines:
     $M_1 = "On \text{ any input:}"
        1. Accept"
     $M_2 = "On \text{ any input:}"
        1. Run $M$ on $w$.
        2. If it accepts, accept:"
     2. Output $<M_1,M_2>:"

$EQ_{TM}$ is not Turing-recognizable

- Theorem 5.30:
  $EQ_{TM}$ is neither Turing-recognizable nor co-Turing-recognizable
- Proof:
  Show $A_{TM} \leq_m \overline{EQ_{TM}}$
\[ A_{TM} \leq_m \overline{EQ_{TM}} \]

- \( G = "\text{On input } <M, w>:" \)
  1. Construction the following two machines:
     - \( M_1 = "\text{On any input:} \)
       1. Reject."
     - \( M_2 = "\text{On any input:} \)
       1. Run \( M \) on \( w \).
       2. If it accepts, accept."
  2. Output \( <M_1, M_2>." \)

---

**Solvable, half-solvable, hopeless**

- Turing-recognizable
- Turing-decidable
- co-Turing-recognizable