$HP = \{ M \# x \mid M \text{ halts on } x \}$ is not recursive

HP is recursively enumerable.

Membership Problem.

$MP = \{ M \# x \mid x \in L(M) \}$.

MP is r.e.

Is MP recursive ?

Suppose $\exists$ a total TM K s.t $L(K) = MP$

Given a TM M and input x. To check if M halts on x

Build a new TM N_M that does the following.

- Similar to M, N_M accepts if M accepts or rejects.

For all $x \in \Sigma^*$, N_M accepts x iff M halts on x.

For any M & x, to check if M halts on x

Construct N_M and run K on input $N_M \# x$.

By assumption K is a total TM. But then we can construct a total TM K' s.t $L(K') = HP$

This is a contradiction.

HP is not recursive, HP is r.e

MP is not recursive, MP is r.e

Lemma. if A is r.e and $\bar{A}$ is r.e then A is recursive

Therefore

- $\overline{HP}$ is not r.e
- $\overline{MP}$ is not r.e.

Properties of TMs.

- Is it decidable if a TM M takes more than 100 steps on input ϵ?

Simulate M on input ϵ for 100 steps - Universal TM.

- Is it decidable if a TM M accepts ϵ.
- Is it decidable if for a TM M, $L(M) = \phi$?
- Is it decidable if for a TM M, $L(M) = \Sigma^*$?

- Is it decidable if for a TM M, $L(M)$ is regular?
- Is it decidable if for a TM M, $L(M)$ is a CFL?
- Is it decidable if for a TM M, $L(M)$ is recursive?

Some Undecidable problems involving TMs.

Is it decidable if a TM M accepts ϵ.

Suppose $\exists$ a total TM K that can decide if a given TM M accepts ϵ. We can then decide the halting problem.

Given a TM M and string x, to determine if M halts on x.

Construct M_1 that on input y works as follows.

1. Erases its input y.
2. writes x on the tape.
3. Runs M on input x.

M and x are hard-coded in M_1.

4. Accepts if M halts on x.

if M halts on x, M_1 accepts y, $\forall$ strings y.

$$\therefore \quad L(M_1) = \begin{cases} \Sigma^* & \text{if } M \text{ halts on } x \\ \phi & \text{if } M \text{ does not halt on } x. \end{cases}$$

Run K with input M_1.

if K accepts $\Rightarrow \epsilon \in L(M_1) \Rightarrow L(M_1) = \Sigma^* \Rightarrow M$ halts on x

if K rejects $\Rightarrow \epsilon \notin L(M_1) \Rightarrow L(M_1) = \phi \Rightarrow M$ does not halt on x.

Given a TM M and string x, to determine if M halts on x.

Construct M_1 that on input y works as follows.

1. Erases its input y.
2. writes x on the tape.
3. Runs M on input x.

M and x are hard-coded in M_1

4. Accepts if M halts on x.

if M halts on x, M_1 accepts y, $\forall$ strings y.

$$\therefore \quad L(M_1) = \begin{cases} \Sigma^* & \text{if } M \text{ halts on } x \\ \phi & \text{if } M \text{ does not halt on } x. \end{cases}$$

Run K with input M_1.

if K accepts $\Rightarrow \epsilon \in L(M_1) \Rightarrow L(M_1) = \Sigma^* \Rightarrow M$ halts on x

if K rejects $\Rightarrow \epsilon \notin L(M_1) \Rightarrow L(M_1) = \phi \Rightarrow M$ does not halt on x.

- Is it decidable if for a TM M, $L(M) = \phi$?
- Is it decidable if for a TM M, $L(M) = \Sigma^*$?

Same construction as above.

- Is it decidable if for a TM M, $L(M)$ is regular?

Choose a set A that is r.e. but not recursive.
Eg. $A = HP$ or $A = MP$. Let N-TM where $L(N) = A$.

Suppose $\exists$ a total TM K that can decide, given an arbitrary TM M if $L(M)$ is regular.
Then using K we can decide the halting problem.

Given M and x to determine if M halts on x.

Construct a TM M_2 which on input y does the following:
↳ with multiple tracks.

1. Writes y on one of the tracks
2. Writes x on a separate track.
3. Runs M on input x

} M and x are hard coded in M_2

4. if M halts on x then M_2 runs N on y.
(y is M_2 original input)
M_2 accepts if N accepts y.

if M does not halt on x then M_2 does not accept any string

$$L(M_2) = \begin{cases} A & \text{if } M \text{ halts on } x. \\ \phi & \text{if } M \text{ does not halt on } x \end{cases}$$

A is not recursive, not CFL, not regular.
ϕ is regular, CFL and recursive.

Techniques to show undecidability :

Diagonalization and Reduction.

To show problem B is undecidable - reduce HP to B.

That is, Yes instances of HP become Yes instances of B

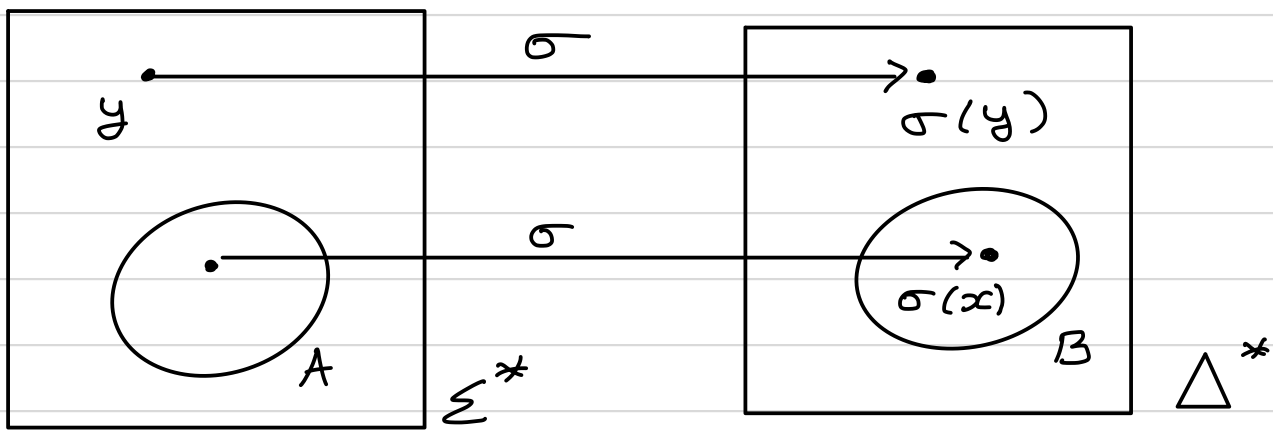
No instances of HP become No instances of B.

If B is decidable, use the decision procedure of B as a subroutine to get a decision procedure for HP – contradiction.

Reduction.

Given $A \subseteq \Sigma^*$, $B \subseteq \Delta^*$, a (many-one) reduction of A to B is a computable function.

$\sigma : \Sigma^* \to \Delta^*$ s.t $\forall x \in \Sigma^*$, $x \in A$ iff $\sigma(x) \in B$



σ should be computable by a total TM.
A total TM that on any input x writes $\sigma(x)$ on the tape and halts.

σ need not be one-to-one or onto.

$A \leq_m B$ - A reduces to B via map σ.

Observation. $\leq_m$ between sets is transitive.

if $\underset{\sigma}{A \leq_m B}$ and $\underset{\tau}{B \leq_m C}$ then $\underset{\tau \circ \sigma}{A \leq_m C}$

Example 1. Given M is $\epsilon \in L(M)$?

$A = \{M \# x \mid M \text{ halts on } x\} = HP$

$B = \{M \mid \epsilon \in L(M)\}$.

σ is the computable function $M \# x \mapsto M_1$.

Example 2. Given M, is $L(M)$ regular?

Given M we constructed M_2 s.t

$$L(M_2) \begin{cases} \text{is not regular if } M \text{ halts on } x. \\ \text{is } \phi \quad \text{if } M \text{ does not halt on } x. \end{cases}$$

$A = \{M \# x \mid M \text{ halts on } x\} = HP$

$B = \{M \mid L(M) \text{ is regular}\}$

σ is the computable function $M \# x \mapsto M_2$

Theorem.

1. if $A \leq_m B$ and B is r.e then A is r.e.

↳ if $A \leq_m B$ and A is not r.e then B is not r.e.

2. if $A \leq_m B$ and B is recursive then A is recursive

↳ if $A \leq_m B$ and A is not recursive then B is not recursive.

Proof.

1. Suppose $A \leq_m B$ via σ and B is r.e. $B = L(M)$

Construct a TM N s.t $L(N) = A$ as follows:

On input x,
1. Compute $\sigma(x)$
2. Run M on $\sigma(x)$
3. Accept if M accepts.

N accepts x iff M accepts $\sigma(x)$ iff $\sigma(x) \in B$ iff $x \in A$

Definition of $A \leq_m B$.

Theorem.

1. if $A \leq_m B$ and B is r.e then A is r.e.

↳ if $A \leq_m B$ and A is not r.e then B is not r.e.

2. if $A \leq_m B$ and B is recursive then A is recursive

↳ if $A \leq_m B$ and A is not recursive then B is not recursive.

Proof.

1. Suppose $A \leq_m B$ via σ and B is r.e. $B = L(M)$

Construct a TM N s.t $L(N) = A$ as follows:

On input x,
1. Compute $\sigma(x)$
2. Run M on $\sigma(x)$
3. Accept if M accepts.

N accepts x iff M accepts $\sigma(x)$ iff $\underline{\sigma(x) \in B \text{ iff } x \in A}$

Definition of $A \leq_m B$.

2. Recall: A is recursive iff A is r.e and $\overline{A}$ is r.e.

Suppose $A \leq_m B$ via σ and B is recursive.

Then $\overline{A} \leq_m \overline{B}$ via σ [follows from the definition]

if B is recursive then both B and $\overline{B}$ are r.e.

By part 1, both A and $\overline{A}$ are r.e. $\Rightarrow$ A is recursive

Example 1. $FIN = \{ M \mid L(M) \text{ is finite} \}$ is not r.e

$\overline{FIN}$ is not r.e.

We give a reduction: $\left.\begin{array}{l} \overline{HP} \leq_m FIN \text{ and} \\ \overline{HP} \leq_m \overline{FIN} \end{array}\right\}$ (a)

$$\overline{HP} = \{ M \# x \mid M \text{ does not halt on } x \}.$$

(a) and Theorem $\Rightarrow$ FIN is not r.e., $\overline{FIN}$ is not r.e.

$\overline{HP} \leq_m FIN$: From $M \# x$, construct a TM $M_1 = \sigma(M \# x)$ s.t

M does not halt on x iff $L(M_1)$ is finite.

M_1 on input y works as follows.

1. Erases the input y
2. Writes x on the tape
3. Runs M on input x
4. Accept if M halts on x.

} Descriptions of M and x are hard-coded in M_1.

Example 1. $FIN = \{M \mid L(M) \text{ is finite}\}$ is not r.e.
$\overline{FIN}$ is not r.e.

We give a reduction: $\left.\begin{array}{l}\overline{HP} \le_m FIN \text{ and} \\ \overline{HP} \le_m \overline{FIN}\end{array}\right\}$ (a)

$$\overline{HP} = \{M\#x \mid M \text{ does not halt on } x\}.$$

(a) and Theorem $\Rightarrow$ FIN is not r.e., $\overline{FIN}$ is not r.e.

$\overline{HP} \le_m FIN$: From $M\#x$, Construct a TM $M_1 = \sigma(M\#x)$ s.t
M does not halt on x iff $L(M_1)$ is finite.
M_1 on input y works as follows.

1. Erases the input y
2. Writes x on the tape
3. Runs M on input x
4. Accept if M halts on x.

$\}$ Descriptions of M and x are hard-coded in M_1.

if M does not halt on x then M_1 never reaches (4).
$\therefore$ M_1 does not accept its input y.

M halts on $x \Rightarrow L(M_1) = \Sigma^* \Rightarrow L(M_1)$ is infinite
M does not halt on $x \Rightarrow L(M_1) = \phi \Rightarrow L(M_1)$ is finite.
$\therefore$ $\overline{HP} \le_m FIN$.

Note. To produce a description of M_1, σ does not need to simulate M – so σ is Computable.

2. $\overline{HP} \leq_m \overline{FIN}$

By definition of $\leq_m$, if $\overline{A} \leq_m \overline{B}$ via σ then $A \leq_m B$ via σ.

Suffices to show that $HP \leq_m FIN$ via τ
i.e., given M and x, construct $M_2 = \tau(M\#x)$ s.t
M halts on x iff $L(M_2)$ is finite.

M_2 on input y works as follows:

1. Save y on one of the tracks
2. Write x on a separate track
3. Simulate M on x for $|y|$ steps

$\Big\}$ M and x are hard coded in M_2

Erase one symbol in y for each step of M on x

4. Accept if M has not halted in $|y|$ steps. otherwise reject.

2. $\overline{HP} \leq_m \overline{FIN}$

By definition of $\leq_m$, if $\overline{A} \leq_m \overline{B}$ via σ then
$A \leq_m B$ via σ.

Sufices to show that $HP \leq_m FIN$ via τ
i.e., given M and x, construct $M_2 = \tau(M\#x)$ s.t
M halts on x iff $L(M_2)$ is finite.

M_2 on input y works as follows:

1. Save y on one of the tracks

2. Write x on a separate track
3. Simulate M on x for $|y|$ steps

} M and x are hard coded in M_2

Erase one symbol in y for each step of M on x

4. Accept if M has not halted in $|y|$ steps. otherwise reject.

if M does not halt on $x \Rightarrow M_2$ halts and accepts y $(\forall y)$
If M halts on x then it halts after some n steps.
M_2 accepts y if $|y| < n$, rejects y if $|y| \geq n$.

M does not halt on $x \Rightarrow L(M_2) = \Sigma^* \Rightarrow L(M_2)$ is infinite

M halts on $x \Rightarrow L(M_2) = \{y \mid |y| <$ running time of M on $x\}$
$\Rightarrow L(M_2)$ is finite.

Note: The map τ can be computed by a total TM.