REDUCIBILITIES

WE WILL USE REDUCTIONS FROM ONE LANGUAGE TO ANOTHER TO SHOW THAT OTHER LANGUAGES ARE NOT RECURSIVE OR NOT R.E.

DEFINITION LET L_1, L_2 BE LANGUAGES OVER Σ^*

$L_1 \leq_M L_2$ IF THERE IS A COMPUTABLE FUNCTION $f: \Sigma^* \to \Sigma^*$ SUCH THAT FOR ALL $x \in \Sigma^*$

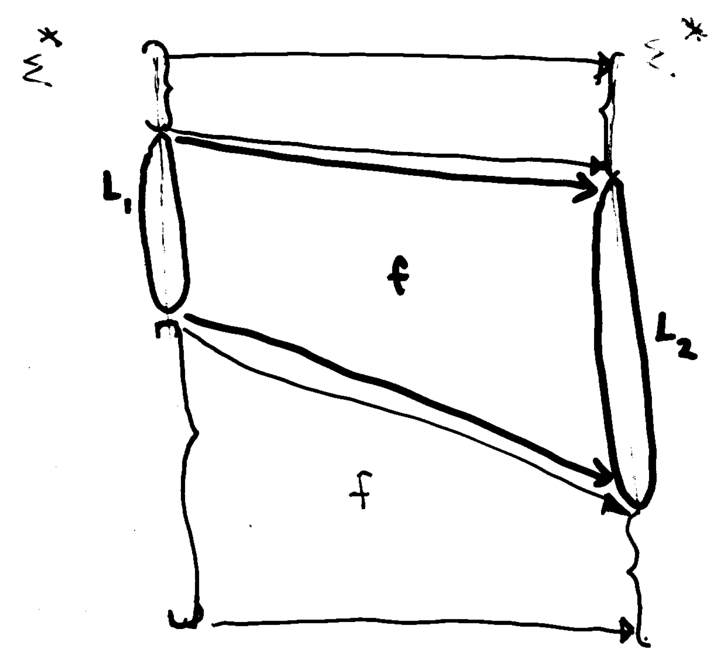
$$x \in L_1 \iff f(x) \in L_2$$

THEOREM SUPPOSE $L_1 \leq_M L_2$. THEN

1.) $\overline{L_1} \leq_M \overline{L_2}$ WHERE $\overline{L_i} = \{w \mid w \in \Sigma^*$ AND w NOT IN $L_i\}$

2.) IF L_2 IS RECURSIVE, THEN L_1 IS RECURSIVE
(L_1 NOT RECURSIVE $\Rightarrow$ L_2 NOT RECURSIVE)

3.) IF L_2 RE, THEN L_1 IS RE.
(L_1 NOT RE $\Rightarrow$ L_2 NOT RE.)



MORE EXAMPLES OF REDUCTIONS ($\Sigma = \{0,1\}$ UNLESS NOTED OTHERWISE)

HB $= \{\langle M \rangle \mid M$ HALTS ON THE BLANK INPUT TAPE$\}$

HALT $= \{\langle M, w\rangle \mid M$ HALTS ON $w\}$

THEOREM BOTH HB AND HALT ARE R.E. BUT ARE NOT RECURSIVE.

PROOF

1. HB $\leq_M$ HALT. $f(\langle M \rangle) \to (\langle M, \varepsilon\rangle)$

   THEREFORE, HALT IS NOT RECURSIVE IF HB IS NOT RECURSIVE.

2. TO SEE THAT HB IS NOT RECURSIVE, WE WILL SHOW THAT $\underbrace{A_{TM}}_{\langle M,w\rangle \text{ WHERE } M \text{ ACCEPTS } w} \leq_M \underbrace{HB}_{\langle M\rangle \text{ WHERE } M \text{ HALTS ON BLANK}}$

f on x:

ASSUME $x \in \Sigma^*$ OF THE FORM $\langle M, w\rangle$. Otherwise $f(x) = 0x$ ← illegal encoding

LET $f(x) = \langle M' \rangle$ WHERE M' WORKS AS FOLLOWS ON THE ALL-BLANK INPUT:

M' on any input including all-blank input:
- M' writes w on the tape and simulates M on w
- If and when M halts and accepts, M' halts and accepts
- If and when M halts and rejects, M' goes into an infinite loop

Show (a) f is computable

(b) $x \in A_{TM} \iff f(x) \in HB$ ← $x = \langle M, w\rangle$ FOLLOWS BECAUSE M ACCEPTS W IF AND ONLY IF M' HALTS ON ALL BLANK INPUT

3) TO SEE THAT HB AND HALT ARE BOTH RE, WE NEED TO SHOW THAT HALT IS R.E. SINCE $HB \leq_M$ HALT, IT FOLLOWS THAT HB IS ALSO RE

M' on <M,W>:

- SIMULATE M ON W
- HALT AND ACCEPT IF M ENTERS A HALT STATE DURING SIMULATION OF M ON W.

M' ACCEPTS HALT, THUS HALT IS R.E.

$HB = \{\langle M\rangle \mid M$ HALTS ON BLANK TAPE$\}$

$\overline{HB} = \{w \mid w \notin HB\} = \{w \mid$ w IS NOT OF FORM $\langle M\rangle$ OR w IS OF FORM $\langle M\rangle$, AND M DOES NOT HALT ON ALL-BLANK TAPE$\}$

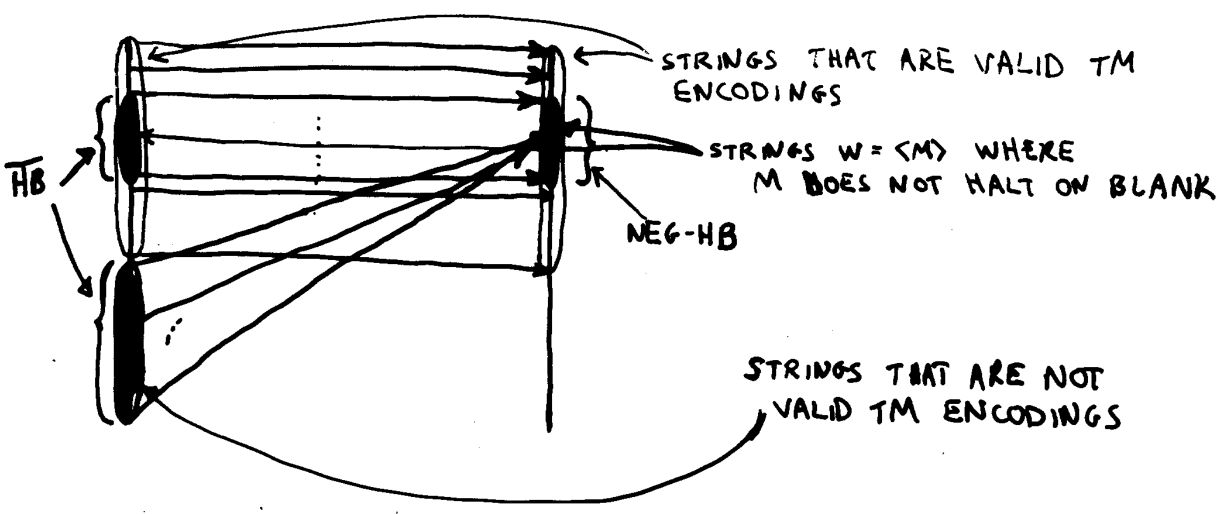
NEG-HB $= \{\langle M\rangle \mid M$ DOES NOT HALT ON BLANK TAPE$\}$

THEOREM $\overline{HB}$ AND NEG-HB ARE NOT R.E.

Proof BY HOMEWORK PROBLEM 5, IF $\overline{HB}$ IS RE AND HB IS RE, THEN HB IS RECURSIVE. SINCE WE HAVE ALREADY SHOWN HB IS RE BUT NOT RECURSIVE, IT FOLLOWS THAT $\overline{HB}$ IS NOT RECURSIVELY ENUMERABLE.

TO SEE THAT NEG-HB IS ALSO NOT RE, WE WILL SHOW $\overline{HB} \leq_M$ NEG-HB. DEFINE f ON w AS FOLLOWS.

f ON w:
- CHECK IF w IS A VALID TM ENCODING. IF NOT, $f(w) = \langle M^*\rangle$ WHERE M^* IS A PARTICULAR TM THAT GOES INTO AN INFINITE LOOP ON ALL INPUTS
- OTHERWISE (w IS A VALID TM ENCODING) $w = \langle M\rangle$ FOR SOME M. THEN $f(w) = w$.



Claim: $x \in \overline{HB}$ IF AND ONLY IF $f(x) \in$ NEG-HB, AND F IS COMPUTABLE. THUS NEG-HB IS NOT RE.

NOTE THERE IS A SOMEWHAT SIMPLER PROOF THAT NEG-HB IS NOT RE, USING A MORE GENERAL FORM OF A REDUCTION WHERE THE TM FOR NEG-HB IS USED AS A SUBROUTINE.

ALTERNATIVE PROOF THAT NEG-HB NOT RE:

ASSUME FOR SAKE OF CONTRADICTION THAT NEG-HB IS RE. LET M ACCEPT NEG-HB. (NOTE THAT M MIGHT NOT ALWAYS HALT.) WE CONSTRUCT A TM M' FOR $\overline{HB}$ AS FOLLOWS.

M' ON INPUT w

- IF $w = \langle M'' \rangle$ FOR SOME M'' (THAT IS, w IS A VALID TM ENCODING), SIMULATE M ON INPUT $w = \langle M'' \rangle$, AND ACCEPT w IF M ACCEPTS w.
- OTHERWISE (IF w NOT A VALID TM ENCODING), ACCEPT w.

CLAIM M' ACCEPTS $\overline{HB}$. BUT THIS CONTRADICTS FACT THAT $\overline{HB}$ IS NOT RE. THUS NEG-HB IS NOT RE.

MORE EXAMPLES

	RECURSIVE	RE
A_{TM}	NO	YES
L_D	NO	NO
$\overline{L_D}$	NO	YES
HB, HALT	NO	YES
NE	NO	YES
NEG-NE, $\overline{NE}$, $\overline{HB}$, NEG-HB	NO	NO
NEG-ALL, ALL, $\overline{ALL}$	NO	NO
INF, $\overline{INF}$, NEG-INF	NO	NO

$NE = \{\langle M\rangle \mid L(M) \neq \phi\}$ ← valid encodings of TM's, M, where M does not accept the empty language.

$\overline{NE} = \{w \mid w \notin NE\}$; NEG-NE $= \{\langle M\rangle \mid L(M) = \phi\}$

$ALL = \{\langle M\rangle \mid L(M) = \Sigma^*\}$

$\overline{ALL} = \{w \mid w \notin ALL\}$; NEG-ALL $= \{\langle M\rangle \mid L(M) \neq \Sigma^*\}$

$INF = \{\langle M\rangle \mid$ M accepts an infinite language$\}$

$\overline{INF} = \{w \mid w \notin INF\}$; NEG-INF $= \{\langle M\rangle \mid L(M)$ is finite$\}$

NOTE THE NEG-VERSIONS, NEG-NE, NEG-ALL, NEG-INF ARE ALSO NOT RE BY PREVIOUS ARGUMENT FOR NEG-HB.

LEMMA $NE = \{\langle M\rangle \mid L(M) \neq \phi\}$ IS RE, BUT NOT RECURSIVE.

PROOF

① TO SHOW THAT NE IS RE, WE WILL DESCRIBE A TM M_0 SUCH THAT $NE = L(M_0)$.

M_0 on input $\langle M\rangle = w$

- IF w NOT A VALID TM ENCODING, REJECT.
- OTHERWISE $w = \langle M\rangle$.

SIMULATE M ON ALL INPUTS OF LENGTH ≤ 1 FOR 1 STEP

SIMULATE M ON ALL INPUTS OF LENGTH ≤ 2 FOR 2 STEPS

⋮

SIMULATE M ON ALL INPUTS OF LENGTH $\leq i$ FOR i STEPS

⋮

IF M EVER ACCEPTS AN INPUT IN ABOVE SIMULATIONS, M_0 HALTS AND ACCEPTS.

THE ABOVE TECHNIQUE IS CALLED DOVE-TAILING.

* IF $L(M) \neq \phi$ THEN THERE IS SOME STRING w AND SOME NUMBER t SUCH THAT M HALTS AND ACCEPTS w AFTER t STEPS.
THEN M_0 WILL HALT AND ACCEPT WHEN IT SIMULATES M ON w FOR $\max(|w|, t)$ MANY STEPS.

* OTHERWISE IF $L(M) = \phi$, THEN EVERY SIMULATION BY M_0 OF M ON SOME STRING WILL NOT ACCEPT, SO M_0 NEVER ACCEPTS $w = \langle M\rangle$.

THUS $NE = L(M_0)$ SO NE IS RE.

② TO SHOW THAT NE IS NOT RECURSIVE, WE WILL SHOW

$$\underbrace{HB}_{\{\langle M\rangle \mid M \text{ HALTS ON BLANK INPUT}\}} \leq_M \underbrace{NE}_{\{\langle M\rangle \mid L(M) \neq \phi\}}$$

<u>f on input w</u>:

(a) CHECK IF W IS A VALID TM ENCODING. IF NOT, $f(w) = w$.

(b) OTHERWISE $W = \langle M \rangle$.
CONSTRUCT $\langle M' \rangle = f(\langle M \rangle)$ SUCH THAT $\langle M \rangle \in HB$
IF AND ONLY IF $\langle M' \rangle \in NE$.

M' on input α:
- ERASE INPUT α, AND RUN M ON BLANK TAPE
- IF M HALTS ON BLANK TAPE, M' HALTS AND ACCEPTS α

*NOTE: $\langle M' \rangle$ DEPENDS ON $\langle M \rangle$.

<u>CLAIM</u> M' WILL EITHER ACCEPT ALL STRINGS, AND THIS HAPPENS IF M HALTS ON BLANK TAPE, OR M' WILL ACCEPT NO STRINGS, AND THIS HAPPENS WHEN M DOES NOT HALT ON BLANK TAPE.
THUS $w \in HB \iff f(w) \in NE$

<u>CLAIM</u> f IS COMPUTABLE. STEP (a) IS CLEARLY COMPUTABLE. STEP (b) IS COMPUTABLE BECAUSE WE CAN COMPUTE $\langle M' \rangle$ FROM $\langle M \rangle$.

$\therefore$ NE IS NOT RECURSIVE SINCE HB IS NOT RECURSIVE.

LEMMA INF, $\overline{\text{INF}}$, NEG-INF ARE NOT R.E

INF = $\{\langle M\rangle \mid L(M) \text{ IS INFINITE}\}$

PROOF

① TO SHOW $\overline{\text{INF}}$ NOT RE, WE WILL SHOW HB $\leq_M$ INF.
THEN HB $\leq_M$ INF IMPLIES $\overline{\text{HB}} \leq_M \overline{\text{INF}}$, AND THUS IT FOLLOWS THAT $\overline{\text{INF}}$ NOT RE, BECAUSE $\overline{\text{HB}}$ NOT RE.

TO SHOW HB $\leq_M$ INF, WE CONSTRUCT f SUCH THAT $x \in \text{HB} \Leftrightarrow f(x) \in \text{INF}$.

f on x

a) If x NOT A VALID TM ENCODING, $f(x) = x$
b) Otherwise $x = \langle M\rangle$. Let $f(\langle M\rangle) = \langle M'\rangle$ WHERE M' IS AS FOLLOWS.

M' on α:
- M' SIMULATES M ON BLANK TAPE
- IF M HALTS ON BLANK TAPE, M' ACCEPTS α

Claim

f IS COMPUTABLE AND
$x \in \text{HB} \Leftrightarrow f(x) \in \text{INF}$.

THUS $\overline{\text{INF}}$ NOT RE.

$x \in \text{HB} \Rightarrow f(x) \in \text{INF}$:

$x \in \text{HB}$ IMPLIES x A VALID TM ENCODING, $\langle M\rangle = x$, AND $\langle M\rangle$ HALTS ON BLANK TAPE.
THUS M' ON ALL α HALTS AND ACCEPTS.
THUS $L(M') = \Sigma^*$, AND THUS $f(x) = \langle M'\rangle \in \text{INF}$.

$f(x) \in \text{INF} \Rightarrow x \in \text{HB}$. ($x \notin \text{HB} \Rightarrow f(x) \notin \text{INF}$):

$x \notin \text{HB}$ MEANS EITHER (i) x NOT A VALID ENCODING OR (ii) $x = \langle M\rangle$ AND M DOES NOT ACCEPT BLANK. IF (i), THEN $f(x) = x$ AND $f(x) \notin \text{INF}$. IF (ii) THEN M' ACCEPTS NO STRINGS, SO $L(M') = \phi$, AND THUS $f(\langle M\rangle) = \langle M'\rangle \notin \text{INF}$.

② TO SHOW INF NOT RE, WE WILL SHOW $HB \leq_M \overline{INF}$

THIS IMPLIES $\overline{HB} \leq_M INF$.

<u>f on x</u>

a) IF x NOT A VALID TM ENCODING, $f(x) = \langle M^* \rangle$
where M^* is SOME TM ACCEPTING ALL INPUTS

b) OTHERWISE $x = \langle M \rangle$. LET $f(\langle M \rangle) = \langle M' \rangle$, WHERE M' IS AS FOLLOWS.

- M' SIMULATES M ON BLANK TAPE FOR $|\alpha|$ STEPS
- IF M HALTS IN $|\alpha|$ STEPS, M' REJECTS α
  OTHERWISE M' ACCEPTS α.

<u>CLAIM</u> f IS COMPUTABLE AND $x \in HB \leftrightarrow f(x) \in \overline{INF}$

- IF x IS NOT A VALID ENCODING, $x \notin HB$
  and $f(x) \notin \overline{INF}$

- IF x IS A VALID ENCODING, $x = \langle M \rangle$:
  a) THEN x HALTS ON BLANK AT TIME t
  IMPLIES THAT M' ACCEPTS EXACTLY THOSE STRINGS OF LENGTH $< t$. THUS $L(M')$ IS FINITE

  b) IF x DOES NOT HALT ON BLANK TAPE (EVER) THEN M' ACCEPTS ALL STRINGS.
  THUS $L(M')$ IS INFINITE.