CSE 396 Lecture Thu 4/28 Spring 2016

<u>Key Definition</u>: A language A $\left\{\begin{array}{l}\text{many-one reduces} \\ \underline{\text{mapping-reduces}}\end{array}\right.$ to a language B, written $\underline{A \leq_m B}$, if there is a total computable function $f: \Sigma^* \to \Sigma^*$ such that:

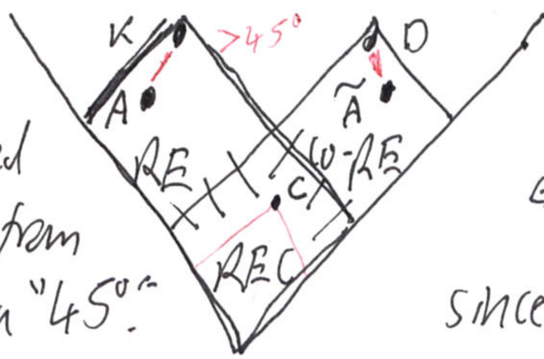
$$\text{for all } x \in \Sigma^*, \quad x \in A \iff f(x) \in B.$$

Note: This is the same as $x \in \tilde{A} \iff f(x) \in \tilde{B}$, so

$A \leq_m B \iff \tilde{A} \leq_m \tilde{B}$. "Mapping Reductions are Mirror-Image."

My <u>Visual Convention</u>

$A \leq_m B$ is indicated by making the angle from A up to B steeper than "45°."



Note $A \leq_m K$

$\iff \tilde{A} \leq_m D$

since $D = {\sim}K$.

essentially, ignoring invalid codes.

<u>Key Lemma</u>: For all languages A, B:

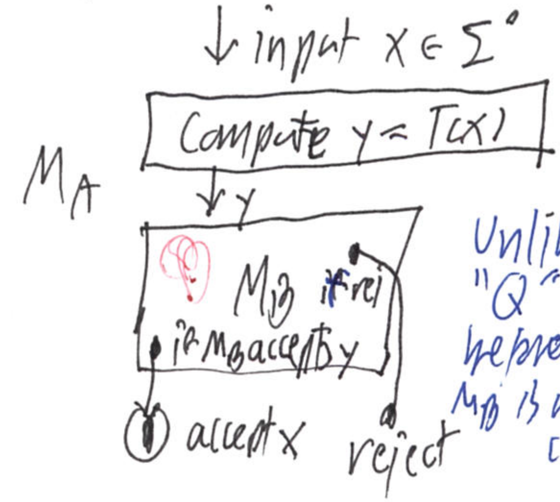
(a) If $A \leq_m B$ and B is decidable, then A is decidable

(b) If $A \leq_m B$ and $B \in RE$, then $A \in RE$, i.e. A too is recognizable.

(c) If $A \leq_m B$ and $B \in$ co-RE, then A too is in $\underline{\text{co-RE}}$.

<u>Proof</u> of (c) first assuming (b): If $A \leq_m B$ and $B \in$ coRE, then $\tilde{B} \in RE$ and $\tilde{A} \leq_m \tilde{B}$. By (b), $\tilde{A} \in RE$, so finally $A \in$ coRE.

Proof of (a): Take some total TM M_B s.t. $L(M_B) = B$. Take a total TM T that computes f. Build M_A as follows:



Unlike "Q" here, M_B is real code.

Correctness: M_A is total and for all x,
M_A accepts x $\iff$ M_B accepts y
by construction $\iff$ M_B accepts $f(x)$
by $L(M_B) = B$ $\iff$ $f(x) \in B$
by reduction $\iff$ $x \in A$. $\therefore L(M_A) = A$.
Since M_A is total, A is decidable.

For (b), even if M_B merely recognizes B, M_A still recognizes A. ⊠

Remark ⟹ Chapter 7 (Thm 7.31)

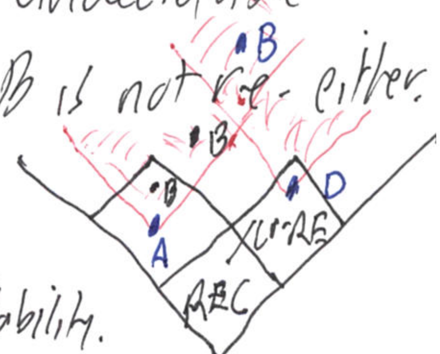
In case (a), you might think the total running time $t(n)$ for M_A equals the sum of the runtime $t_1(n)$ for $y = T(x)$ and the ($n = |x|$) runtime $t_2(?)$ for M_B. But note: "?" $= |y|$ which might be $\approx t(n)$. Best estimate for the time by M_A is $t_2(t_1(n))$. Still polynomial if both $t_1, t_2 = n^{O(1)}$.

Corollary — The Contrapositive: If $A \leq_m B$, then:

(a'): if A is undecidable then B is undecidable.

(b'): if A is not r.e. (cognizable) then B is not r.e. either.

(c'): if $A \notin$ co-RE, then $B \notin$ co-RE.



Part (a') is how we use reductions to show undecidability.

And: if $A \notin$ co-RE and $A \leq_m B$ and $D_{TM} \leq_m B$ too, then B is in an intersection of two "upward reducibility cones" so B is neither r.e. nor co-r.e.. neither r.e. nor co-r.e

Example: $B = ALL_{TM}$ on p4.

Examples of Reductions: A simple "f" first:

① $K \le_m A_{TM}$ via the function $f(u) = \langle u,u\rangle$.

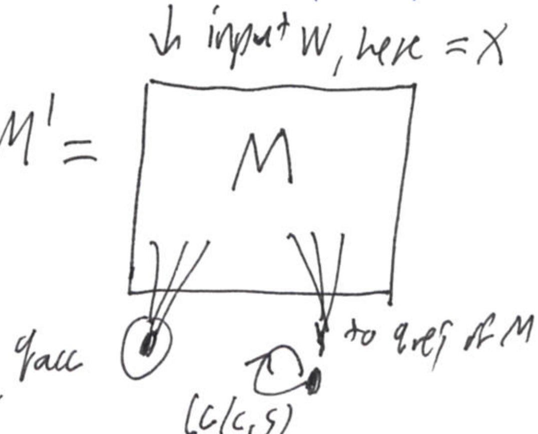
Correct since $K = \{u : u$ is the code of a TM M_u such that M_u accepts $u\}$

so $u \in K \iff M_u$ accepts $u \iff \langle M_u, u\rangle \in A_{TM}$

$\langle M_u, u\rangle$ ||| identified with $\langle u,u\rangle$

Annoying Issue Again: What if u is not a valid code?
OK, then $\langle u,u\rangle$ isn't valid either, so $u \notin K$ and $\langle u,u\rangle \notin A_{TM}$.

② $A_{TM} \le_m HALT_{TM}$ $\qquad M \xrightarrow{f} M' =$



$\langle M,x\rangle \qquad \langle M',x'\rangle$

x' ||| make $= x$

Construction

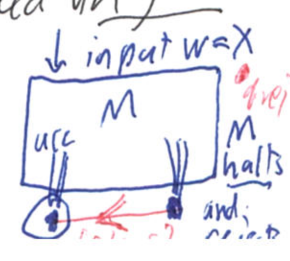
Computability: $f(\langle M,x\rangle) = \langle M',x'\rangle$ where $x' = x$ and M' is computed by adding $[q_{rej}, c/c, S, q_{rej}]$ for all $c \in \Gamma$ to the code of M. This is a "lexical transformation of code" and is ("easily") computable.

Correctness: $\langle M,x\rangle \in A_{TM} \equiv M(x)$ accepts $\iff M'(x)$ halts, because M' doesn't halt when M doesn't accept. $\iff \langle M',x\rangle \in HALT_{TM}$
$\iff f(M,x) \in HALT_{TM}$.

Thus $A_{TM} \le_m HALT_{TM}$. ☒

③ Also $HALT_{TM} \le_m A_{TM}$. Thinking of correctness first, we need an f' defined on instances $\langle M,x\rangle$ of the $HALT_{TM}$ problem so that $\langle M,x\rangle$ is in the $HALT_{TM}$ language : $M(x)\downarrow \iff M'$ accepts $x \equiv \langle M',x\rangle \in A_{TM}$ where $\langle M',x\rangle = f'(M,x)$.

$M \xrightarrow{f'} M' =$



Since $A_{TM} \leq_m HALT_{TM}$ <u>and</u> $HALT_{TM} \leq_m A_{TM}$, we write $A_{TM} \equiv_m HALT_{TM}$ and say they are <u>many-one mapping-equivalent</u>.

(ample

ALL$_{TM}$ (see bottom added note)

$A_{TM} \leq_m NE_{TM}$: $\langle M, x \rangle \xrightarrow{f} M' =$

↓ input w.

[Test tests $w \stackrel{?}{=} x$, but not necessary.]

Simulate $M(x)$

if & when M accepts x

accept w

NE_{TM}: INST: M

QUES: Is $L(M) \neq \emptyset$?

Visually, A_{TM} does NOT mapping reduce to E_{TM}!

<u>Computability</u>

This f is computable because it takes M and x and inserts it as a "job" into the code of M!

<u>Correctness</u>:

$\langle M, x \rangle \in A_{TM} \equiv M$ accepts $x \Rightarrow$ for all w, M' accepts $w \Rightarrow L(M') = \Sigma^*$
$\Rightarrow L(M') \neq \emptyset \Rightarrow \langle M' \rangle \in NE_{TM}$

$\langle M, x \rangle \notin A_{TM} \equiv$ M does not accept x $\Rightarrow$ for all w, M' does <u>not</u> accept w
$\Rightarrow L(M') = \emptyset \Rightarrow \langle M' \rangle \notin NE_{TM}$.

$\therefore A_{TM} \leq_m NE_{TM}$. ☒

$\therefore \langle M, x \rangle \in A_{TM} \Leftrightarrow f(M, x) = \langle M' \rangle \in NE_{TM}$, so $A_{TM} \leq_m NE_{TM}$. ☒

<u>Corollary</u>: $\widetilde{A_{TM}} \leq_m E_{TM}$ since $E_{TM} = \widetilde{NE_{TM}}$, so we get that since $\widetilde{A_{TM}}$ is not recognizable, <u>E_{TM} is not recognizable either</u>.

<u>Added</u>: By the <u>same</u> construction <u>and</u> analysis, $\boxed{A_{TM} \leq_m ALL_{TM}}$.

<u>Separately</u>, we can show $\boxed{E_{TM} \leq_m ALL_{TM}}$ too:

$M \xrightarrow{f'} M' =$ ↓ input w [Simulate $M(x)$ for $|w|$ steps on each $x \leq w$. If <u>any</u> acc, reject. Else if <u>no</u> accepts were found, accept.]

$M \in E_{TM} \Rightarrow$ for all w, $M'(w)$ never finds an accept $\Rightarrow L(M') = \Sigma^*$

$M \notin E_{TM} \Rightarrow$ for some (long enough) w, $M'(w)$ finds accept $\Rightarrow L(M') \neq \Sigma^*$.