Reductions II

Thursday, March 25, 2021 11:28 AM

$P \leqslant Q$

Q is at least as P; so if P is undecidable then Q must also be <u>undecidable</u>.

x

$P \leqslant_{\underline{\underline{m}}} Q$

Suppose $L_1, L_2 \in \Sigma^*$.

$L_1 \leqslant_m L_2$

if $\exists$ a <u>total</u> <u>computable</u> function

$f : \Sigma^* \to \Sigma^*$

s.t. $\forall w \in \Sigma^*$

$w \in L_1$ iff $f(w) \in L_2$.

What is the difference $P \leqslant Q$, $P \leqslant_m Q$

- with $\leqslant$ you make a transformation of a P-problem and then you can do post processing and you can ask multiple questions to your Q-solver
- with $\leqslant_m$ you can only ask one Q question and all you get to do is report the answer; you cannot even negate the answer.

---

3. If $P \leqslant_m Q$ and Q is CE then P is CE
4. If $P \leqslant_m Q$ and P is NOT CE then Q cannot be CE
5. If $P \leqslant_m Q$ and P is not co CE then Q cannot be co CE.

x

$A_{TM} \leqslant EMPTY_{TM}$

BUT

IT IS NOT TRUE THAT

$A_{TM} \cancel{\leqslant_m} EMPTY_{TM}$

If there were such a reduction

$\overline{A_{TM}} \cancel{\leqslant_m} \overline{EMPTY_{TM}}$

would also hold.

But we know $\overline{A_{TM}}$ is <u>not</u> CE

However $\overline{EMPTY_{TM}}$ is CE.

<u>THM</u> $EQ_{TM} = \{\langle M_1, M_2\rangle \mid L(M_1) = L(M_2)\}$ is neither CE nor co CE

<u>PROOF</u> : Idea (1) we will show

$A_{TM} \leqslant_m EQ_{TM}$

A_{TM} is not co CE so EQ_{TM} cannot be co CE

(2) we will show $A_{TM} \leqslant_m \overline{EQ_{TM}}$

This shows $\overline{EQ_{TM}}$ is not co CE

i.e. EQ_{TM} cannot be CE.

(1) Input $\langle M, w\rangle$

Construct (a) M_1 ; input x ignore x and accept

so $L(M_1) = \Sigma^*$

(b) M_2 ; input x
- ignore x
- simulate M on w and if M(w) accepts then M_2 accepts x.

$L(M_1) = \Sigma^*$; $L(M_2) = \begin{cases} \Sigma^* & \text{if M accepts w} \\ \emptyset & \text{otherwise} \end{cases}$

M accepts w iff $L(M_1) = L(M_2)$.

(2) Given $\langle M, w\rangle$

Define M_1, M_2

M_1 : ignore input and reject

$L(M_1) = \emptyset$

M_2 : same as above

M does <u>not</u> accept w iff

$L(M_1) = L(M_2)$

So EQ_{TM} is neither CE nor co CE.

$INF = \{\langle M\rangle \mid |L(M)| = \infty\}$

Claim INF is not CE

$\overline{H_{TM}} \leqslant_m INF$

↓

definitely not CE (in fact it is co CE).

Given $\langle M, w\rangle$

Define M' with input x

M' works as follows:

1. Simulate M on w but stop the simulation after $|x|$ steps.
2. if M halts before the end of the simulation, reject x. else accept x.

If M does <u>not</u> halt on w then $L(M')$ is infinite else finite.

$\langle M, w\rangle \in \overline{H_{TM}} \iff \langle M'\rangle \in INF$

<u>RICE'S THM</u>

Let PROG be the set of all programs in some TURING COMPLETE language. We can enumerate this set effectively. For simplicity assume all programs are of type $\mathbb{N} \to \mathbb{N}$.

Given $P \in PROG$

$[\![P]\!] = \{(x, y) \mid P(x) = y\}$

$P(x) = y$ when P is run with input x it terminates and outputs y.

It is possible to have

$P_1 \neq P_2$ but $[\![P_1]\!] = [\![P_2]\!]$

We define $P_1 \sim P_2$ if $[\![P_1]\!] = [\![P_2]\!]$

$\sim$ is obviously an equivalence rel^n.

In terms of TM

$M_1 \sim M_2$ iff $L(M_1) = L(M_2)$

$Q : PROG \to \{T, F\}$

We call Q a <u>property</u> of programs

We call Q an <u>extensional</u> <u>property</u> if $P_1 \sim P_2$ then $Q(P_1) \iff Q(P_2)$.

<u>EXAMPLES</u> (i) This program has running time $O(n^2)$. <u>NO</u>

(ii) This program sorts its input <u>YES</u> it is extensional

(iii) This ~~prog~~ has a 100 lines of code. <u>NO</u>

An extensional property only depends on the <u>input-output correspondence</u>.

FUNCTIONAL SPEC

TWO TRIVIAL PROPERTIES

(i) $Q(P) = T$ for $\forall P \in PROG$

(ii) $Q(P) = F$ for $\forall P \in PROG$

Definitely extensional.

<u>THM (RICE)</u> <u>Every</u> non-trivial extensional property is undecidable.

<u>PROOF</u> Let Q be a <u>non</u>-<u>trivial</u> property i.e. $\exists P, P'$ s.t.

$Q(P) = T$ and $Q(P') = F$.

ASSUME $EMPTY = \{\langle M\rangle \mid L(M) = \emptyset\}$ does <u>not</u> satisfy Q.

i.e $\forall M$ $L(M) = \emptyset$ then $\neg Q(M)$.

Let M_0 be such that $Q(M_0) = T$

Of course $L(M_0) \neq \emptyset$.

I will show

$\underline{\underline{A_{TM}}} \leqslant_m L_Q = \{\langle M\rangle \mid Q(M) = T\}$

Input $\langle M, w\rangle$

<u>Assume</u> I have a gadget to solve L_Q i.e to answer whether $Q(\overline{M}) = T$ or F.

CONSTRUCT M':

input of M' is called x

1. simulate M on w
2. If M accepts w then simulate M_0 on x.

FEED $\langle M'\rangle$ to L_Q-gadget.

If M accept w then $L(M') = L(M_0)$ otherwise $L(M') = \emptyset$.

if $L(M') = L(M_0)$ and we know $Q(M_0)$ holds then $Q(M')$ must hold Because Q is extensional.

if $L(M') = \emptyset$ then we know $Q(M') = F$ by our (harmless) assumption on Q.

If we can decide membership in L_Q we have answered the question does M accept w.

Thus membership in L_Q must be <u>undecidable</u>.

<u>All</u> FUNCTIONAL SPECS are undecidable.