Top Hat 4703

Three Reduction "Toolkit" Ideas.

1. "Waiting for Godot" style Reductions

Consider the following kind of "code cleanup" problem:

**INSt:** An OOP program $P$ and a class $C$ in $P$.

**Ques:** Is there any possible input $x$ so that $P(x)$ eventually creates an object of class $C$?

**Con:** [Or, can we just delete class $C$ by now?]

The language $CC$ is r.e.

**Theorem:** $ATM \leq_m CC$, so the CC problem is undec.

Suppose we had a decider $R$ for (all cases of) the CC problem.

Then we would get a decider $Q$ for the ATM problem as follows:

**Construction:**

1. Input $(M, w)$
2. Build $P$ as shown.
3. Run the decider $R$ on $(P)$.
4. If $R$ accepts, accept; else reject.

$f$ is computable because the Turing Kit is a fixed routine.

The reduction is correct because for all $M$ and $w$:

$L(M, w) \in ATM \Rightarrow$ $M$ accepts $w$ on any $x$, $P(x)$ creates a $C$.

$L(M, w) \notin ATM \Rightarrow$ for all $x$, $P(x)$ does not create a $C$.

Because ATM is undecidable, it follows that CC is undec.
"All or Nothing Switch": Revisit $A_{TM} \leq_{m} N_{E_{TM}}$

\[\langle M, w \rangle \xrightarrow{f} M':\]

\[
\begin{align*}
\langle M, w \rangle \in A_{TM} & \implies (\forall x) M' \text{ accepts } x \iff L(M') \neq \emptyset \\
\langle M, w \rangle \notin A_{TM} & \implies L(M') = \emptyset
\end{align*}
\]

So we also have $\implies \langle M' \rangle \notin A_{TM}$

$A_{TM} \leq_{m} A_{TM}$ via the same reduction for $f$.

Also:

\[\langle M, w \rangle \in A_{TM} \implies L(M') = \Sigma^* \implies M' \text{ accepts } \langle M' \rangle\]

\[\langle M, w \rangle \notin A_{TM} \implies L(M') = \emptyset \implies M' \text{ does not accept } \langle M' \rangle\]

\[\therefore A_{TM} \leq_{m} K_{TM} \]

Since we already had $K_{TM} \leq_{m} A_{TM}$,

We write $A_{TM} \equiv_{m} K_{TM}$ and say they are mapping-equivalent.

Is $A_{TM}$ equivalent? Is it e.g.? Let's go to Technique 3.

**Technique 3**: The All or Nothing Switch can be composed with other routines.

\[\langle M, w \rangle \xrightarrow{f} M': \]

\[
\begin{align*}
\langle M, w \rangle \in A_{TM} & \implies L(M') = \{\text{palindromes}\} \\
\langle M, w \rangle \notin A_{TM} & \implies L(M') = \emptyset
\end{align*}
\]

\[\therefore A_{TM} \leq_{m} \text{ "Given a TM } M', \text{ is } L(M') \text{ non regular?" ; REG}_{TM} \text{ in text HW is undecidable.}\]
"Delay Switch": Show $\text{DTM} \leq_m \text{ALL-TM}$

which is the same as showing $K_{\text{TM}} \leq_m \neg \text{ALL-TM}$.

\[ \text{Input } x, \ n = 1 \times |x| \]

\[ \langle M, w \rangle \xrightarrow{f} M' = \text{Simulate } M(w) \text{ for up to } n \text{ steps. Did it accept?} \]

- yes
  - Accept $x$
  - reject $x$

For correctness:

\[ \langle M \rangle \in \text{DTM} \implies M \text{ does not accept } w = \langle M \rangle \]

\[ \langle M \rangle \in K \implies \text{for all } n, M \text{ does not accept } \langle m \rangle \text{ within } n \text{ steps} \]

\[ \implies \text{for all } x, M'(x) \text{ does not take the yes branch} \]

\[ \implies L(M') = \Sigma^* \implies \langle M' \rangle = f(\langle M \rangle) \in \text{ALL-TM} \]

\[ \langle M \rangle \in \text{DTM} \implies M \text{ does not accept } w = \langle M \rangle \]

\[ \implies \text{there is an } n \text{ s.t. } M \text{ accepts } \langle M \rangle \text{ within } n \text{ steps} \]

\[ n \in K \implies \text{on any } x, |x| \geq n, M'(x) \text{ detects this acceptance and so takes the yes branch and rejects } x. \]

\[ \implies L(M') \neq \Sigma^*, \text{ indeed } L(M') \text{ is finite, } \neg f(M') \in K \]

Since $K \leq_m \text{ALL-TM}$ and $K$ is not co-c.e., $\text{ALL-TM}$ is not co-c.e.

Since $D \leq_m \text{ALL-TM}$ and $D$ is co-c.e., $\text{ALL-TM}$ is not c.e.

\[ \therefore \text{ALL-TM is neither c.e. nor co-c.e.} \]
Def: A language $B$ is hard for a class $C$ of languages if for all $A \in C$, $A \leq^m B$.

If also $B \in C$, then $B$ is complete for $C$.

Theorem: $A_{TM}$ is complete for $RE$ under $\leq^m$.

Proof: $A_{TM} \in RE$. \hspace{1cm} \checkmark \hspace{1cm} Let any $A \in RE$ be given. \hspace{1cm} Goal: show $A \leq^m A_{TM}$.

By $A \in RE$, we can by defn take a TM $M_A$ s.t. $\langle LM_A, A \rangle = A_{TM}$.

Given any $x \in \Sigma^*$, map

$s(x) = \langle M_A, x \rangle$. Then

$x \in A \iff M_A\text{ accepts } x \iff \langle M_A, x \rangle = s(x) \in A_{TM}$.

So $A \leq^m A_{TM}$ via $s$, and since $A \in RE$ is arbitrary, $A_{TM}$ is RE-complete.

Fact: If $B$ is complete and $C \equiv^m B$, then $C$ is complete (because $\leq^m$ is transitive).

HALTING PROBLEM

Input $\langle M, w \rangle$

Question: Does $M(w)$ halt?

$A_{TM} \leq^m A_{TM}$:

Input $w$:

$\langle M, w \rangle \in \text{Sim}(M(w), 1)$

Input $\langle M, w \rangle$:

Simulates $M$ on $w$:

Accepts $\langle M, w \rangle$ if $M(w) \downarrow$

$M(w) \downarrow \iff M'(w) \downarrow$

$M(w) \uparrow \iff M'(w) \uparrow$

$M'$ is $M$ with added accept state.