Mapping Reducibility

CSE 480
Computational Theory

Introduction

- What is a *computable function*?
  A function $f : \Sigma^* \to \Sigma^*$ is a **computable function**
  if some Turing Machine M, on every input w,
  halts with just $f(w)$ on its tape.

- What is a *mapping*?
  A **mapping** is a function from one thing to another thing.
  In our current context, the things are languages.

- What is a *mapping reduction*?
  It's a more formal notion of reduction than "plain old" reduction, namely:

  A **mapping reduction** from $A \subseteq \Sigma^*$ to $B \subseteq \Sigma^*$
  (denoted $A \leq_m B$) is a computable function $f : \Sigma^* \to \Sigma^*$
  such that for all $x \in \Sigma^*, x \in A \iff f(x) \in B$.

Typical Use of Mapping Reductions

1. Let A be a language known to be undecidable ("old" or "existing" language).
2. Let B be the language that must be shown to be undecidable ("new" language).
3. Find a mapping reduction f from A to B.
4. Now, if B has a decider D_B, then we can decide membership in A as follows:
   - On input w, in order to check if w is in A, find out if $D_B(f(w))$ accepts or not. If it accepts, then w is in A, and if it rejects, then w is not in A.

Mapping Reduction from A_{TM} to $Halt_{\mathrm{TM}}$

The mapping reduction function f, in effect, generates the *text* of the program M' from the text of M. Given that function, a decider for A_{TM} could then be obtained in 4 easy steps:

M' = **"**On input x:
Run M on x
If the result is "accept," then "accept"
If the result is "reject," then loop**"**

1. Here is the initial tape:

M	w	

2. Build M' and put it on the tape:

M	w	... build M' that incorporates M here ...	

3. Put w on the tape

M	w	... build M' that incorporates M here ...	put w here	

4. Run $D_{Halt_{\mathrm{TM}}}$ on M' and w and return its decision:
   - $D_{Halt_{\mathrm{TM}}}(M', w)$ accepts $\Longrightarrow$ M' halts on $w \Rightarrow M$ accepts w
   - $D_{Halt_{\mathrm{TM}}}(M', w)$ rejects $\Longrightarrow$ M' loops on w $\Longrightarrow$ M rejects w

Mapping Reduction from A_{TM} to $\overline{E_{\text{TM}}}$

Show that $\overline{E_{\text{TM}}} = \{\langle M \rangle \mid M \text{ is a } TM \text{ and } L(M) \neq \emptyset\}$ is undecidable through a mapping reduction that maps $\langle M, w\rangle$ into $\langle M'\rangle$ as follows:

M' = "On input x:
- If $x \neq w$ then goto $reject_{M'}$
- Run M on w
- If M accepts w, goto $accept_{M'}$
- If M rejects w, goto $reject_{M'}$"

Given (an alleged) $D_{E_{\text{TM}}}$, how a decider for A_{TM} *could be* built:

1. Build above M' and put it on the tape:

M	w	... build M' that incorporates M and w here ...	

2. Run $D_{E_{\text{TM}}}$ on M' and return *the opposite* of its decision
   - $D_{E_{\text{TM}}}(M')$ accepts $\implies$ $L(M')$ is empty $\implies$ M rejects w.
   - $D_{E_{\text{TM}}}(M')$ rejects $\implies$ $L(M')$ is not empty $\implies$ M accepts w.

Mapping Reduction from A_{TM} to $Regular_{\text{TM}}$

Similarly, we can prove $Regular_{\text{TM}}$ to be undecidable by building this mapping reduction:

M' = "On input x:
- If x is of the form 0^n1^n then goto $accept_{M'}$
- Run M on w
- If M accepts w, goto $accept_{M'}$
- If M rejects w, goto $reject_{M'}$"

The (alleged) Decider for A_{TM}:

- $D_{Regular_{\text{TM}}}(M')$ accepts $\implies$ $L(M')$ is regular $\implies$ $L(M') = \Sigma^*$ $\implies$ M accepts w.
- $D_{Regular_{\text{TM}}}(M')$ rejects $\implies$ $L(M')$ is not regular $\implies$ $L(M') = 0^n1^n$ $\implies$ M rejects w.

Clarification

Note that the *TM* M' is *not* constructed for the purpose of actually running it on some input — a common confusion. We construct M' only for the purpose of feeding its description into the decider for $\mathit{Regular}_{\mathrm{TM}}$ that we have assumed to exist. Once this decider returns its answer, we can use it to obtain the answer to whether M accepts w. Thus, we can decide A_{TM}, a contradiction.