Reduction Proofs

CS 234

0 Introduction

This document contains examples of good reduction proofs. These are not the only ways to write good proofs.

These proofs may contain footnotes explaining different thought processes that occurred in their construction, to help show you how to think about writing proofs. Commentary may also be provided at the end about alternative approaches.

1 Proofs

INF is Undecidable

Theorem 1. *Let* $INF = \{i \mid |\mathcal{L}(M_i)| = \infty\}$. *The language* INF *is undecidable.*

Proof. This property can be shown via a reduction from the diagonal language D to INF.

Suppose for the sake of contradiction that INF is decidable. Then there is some total Turing machine N that decides INF.

Now use N to define the machine P as given by the following pseudocode:

```
1  P(i) =
2      let Q(x) = M_i(i) in
3      let q = index(Q) in
4      N(q)
```

Now note the following facts about P:

Firstly, P is total. Values are only defined without interesting computation until line 4. Then in line 4, N is run, but N is total by assumption so this process will terminate.

Secondly, $\mathcal{L}(P) = D$. This fact follows from the following two cases:

- Suppose $i \in D$. Then the following chain of implications holds:

$$\begin{aligned}
i \in D &\implies M_i(i) \textit{ accepts} && D\ def \\
&\implies \forall x.\, Q(x) \textit{ accepts} && Q\ def \\
&\implies |\mathcal{L}(Q)| = \infty && |\Sigma^*| = \infty \\
&\implies |\mathcal{L}(M_q)| = \infty && q\ def \\
&\implies N(q) \textit{ accepts} && N\ def \\
&\implies i \in \mathcal{L}(P) && P\ def
\end{aligned}$$

- Suppose $i \notin D$. Then the following chain of implications holds:

$$\begin{aligned}
i \notin D &\implies M_i(i) \textit{ rejects} && D\ def \\
&\implies \forall x.\, Q(x) \textit{ rejects} && Q\ def \\
&\implies |\mathcal{L}(Q)| \neq \infty && |\emptyset| = 0 \neq \infty \\
&\implies |\mathcal{L}(M_q)| \neq \infty && q\ def \\
&\implies N(q) \textit{ rejects} && N\ def \\
&\implies i \notin \mathcal{L}(P) && P\ def
\end{aligned}$$

Thus, $i \in D \iff i \in \mathcal{L}(P)$, so $\mathcal{L}(P) = D$, and P is total. These facts mean that P decides D, witnessing that D is decidable. However, D is known to be undecidable (see Theorem 14.4 in the textbook). Thus, there is a contradiction, and the assumption that INF is decidable must be false. Therefore, INF is in fact undecidable.

□

SAME is Undecidable

Theorem 2. *Let $SAME = \{(a, b) \mid \mathcal{L}(M_a) = \mathcal{L}(M_b)\}$. The language $SAME$ is undecidable.*

Proof. This property can be shown via a reduction from $HALT$ to $SAME$.

Suppose for the sake of contradiction that $SAME$ is decidable. Then there is some total Turing machine S that decides $SAME$.

Now use S to define the machine H as given by the following pseudocode:

```
1   H(i,x) =
2       let A(y) = run M_i(x) then accept in
3       let a = index(A) in
4       let B(z) = accept in
5       let b = index(B) in
6       S(a,b)
```

Now note the following facts about H:

Firstly, H is total. Values are only defined without interesting computation until line 6. Then in line 6, S is run, but S is total by assumption so this process will terminate.

Secondly, $\mathcal{L}(H) = HALT$. This fact follows from the following two cases:

- Suppose $(i, x) \in HALT$. Then the following chain of implications holds:

$$\begin{aligned}
(i,x) \in HALT &\implies M_i(x) \textit{ halts} && \textit{HALT def} \\
&\implies \forall y.\, A(y) \textit{ accepts} && \textit{A def} \\
&\implies \mathcal{L}(A) = \Sigma^* && \mathcal{L} \textit{ def} \\
&\implies \mathcal{L}(A) = \mathcal{L}(B) && \mathcal{L}(B) = \Sigma^* \\
&\implies \mathcal{L}(M_a) = \mathcal{L}(M_b) && a, b \textit{ def} \\
&\implies S(a,b) \textit{ accepts} && \textit{S def} \\
&\implies (i,x) \in \mathcal{L}(H) && \textit{H def}
\end{aligned}$$

- Suppose $(i, x) \notin HALT$. Then the following chain of implications holds:

$$\begin{aligned}
(i,x) \notin HALT &\implies M_i(x) \textit{ loops} && \textit{HALT def} \\
&\implies \forall y.\, A(y) \textit{ loops} && \textit{A def} \\
&\implies \mathcal{L}(A) = \emptyset && \mathcal{L} \textit{ def} \\
&\implies \mathcal{L}(A) \neq \mathcal{L}(B) && \mathcal{L}(B) = \Sigma^* \\
&\implies \mathcal{L}(M_a) \neq \mathcal{L}(M_b) && a, b \textit{ def} \\
&\implies S(a,b) \textit{ rejects} && \textit{S def} \\
&\implies (i,x) \notin \mathcal{L}(H) && \textit{H def}
\end{aligned}$$

Thus we know that $\mathcal{L}(H) = HALT$ and H is total. In other words, H decides $HALT$. However, $HALT$ is known to be undecidable (see

Theorem 14.7 from the textbook). Thus, there is a contradiction, and the assumption that $SAME$ is decidable must be false. Therefore, $SAME$ is in fact undecidable.

$\square$