Problem Reduction

Sipser Ch 5.1

**Theorem** (Sipser, Theorem 5.2). The language

\[ E_{\text{TM}} = \{ \langle M \rangle : M \text{ is a TM and } L(M) = \emptyset \} \]

is undecidable.

**Proof.** (Sipser reduces from \(A_{\text{TM}}\) but we’ll reduce from \(HALT_{\text{TM}}\) instead.) Suppose \(E_{\text{TM}}\) is decidable and let \(R\) be a TM that decides it. We’ll construct a TM \(S\) to decide \(HALT_{\text{TM}}\), which will give the desired contradiction since \(HALT_{\text{TM}}\) is known to be undecidable. Recall that \(HALT_{\text{TM}}\) is the language

\[ \{ \langle M, w \rangle : M \text{ is a TM and } M \text{ halts on input string } w \} \]

Our TM \(S\) works as follows.

\(S=\) “On input \(\langle M, w \rangle\):

1. Construct an encoding \(\langle M_w \rangle\) of a TM \(M_w\) that works as follows.

   \(M_w=\) ‘On input \(x\):

   (a) If \(x \neq w\), loop.

   (b) If \(x = w\), run \(M\) on \(w\). If \(M\) halts, accept.’

2. Run \(R\) on \(\langle M_w \rangle\).

3. If \(R\) accepts, reject. If \(R\) rejects, accept.’

It remains to be shown that \(S\) in fact decides \(HALT_{\text{TM}}\). So suppose \(\langle M, w \rangle\) is an input to \(S\).

First suppose \(M\) halts on \(w\). Then \(M_w\) accepts \(w\); therefore, \(L(M_w) \neq \emptyset\). Hence, \(R\) rejects \(\langle M_w \rangle\). Thus, \(S\) accepts \(\langle M, w \rangle\).

Next suppose \(M\) loops on \(w\). Then \(M_w\) loops on all input; therefore, \(L(M_w) = \emptyset\). Hence, \(R\) accepts \(\langle M_w \rangle\). Thus, \(S\) rejects \(\langle M, w \rangle\).

Therefore, \(S\) decides \(HALT_{\text{TM}}\). \(\square\)
Theorem (Cf. Sipser, Theorem 5.3). The language

\[ \text{CFL}_{\text{TM}} = \{ \langle M \rangle : M \text{ is a TM and } L(M) \text{ is a CFL} \} \]

is undecidable.

Proof. By reduction from \( A_{\text{TM}} \). Suppose \( \text{CFL}_{\text{TM}} \) is decidable and let \( R \) be a TM that decides it. We’ll construct a TM \( S \) to decide \( A_{\text{TM}} \), which will give the desired contradiction since \( A_{\text{TM}} \) is known to be undecidable. Recall that \( A_{\text{TM}} \) is the language

\[ \{ \langle M, w \rangle : M \text{ is a TM and } M \text{ accepts input string } w \} \]

Our TM \( S \) works as follows.

\( S = \) “On input \( \langle M, w \rangle \):

1. Construct an encoding \( \langle M_w \rangle \) of a TM \( M_w \) such that

\( M_w = \) ‘On input \( x \):

(a) If \( x \) has the form \( yy \), where \( y \in \Sigma^* \), accept.

(b) If \( x \) does not have that form, run \( M \) on \( w \). If \( M \) accepts \( w \), accept. If \( M \) rejects \( w \), reject.’

2. Run \( R \) on \( \langle M_w \rangle \).

3. If \( R \) accepts, accept. If \( R \) rejects, reject.’

It remains to be shown that \( S \) in fact decides \( A_{\text{TM}} \). So suppose \( \langle M, w \rangle \) is an input to \( S \). First suppose \( M \) accepts \( w \). Then \( M_w \) accepts all input, i.e., \( L(M_w) = \Sigma^* \), certainly a context free language. Hence, \( R \) accepts \( \langle M_w \rangle \). Thus, \( S \) accepts \( \langle M, w \rangle \).

Next suppose \( M \) does not accept \( w \), i.e., \( M \) either rejects or loops on \( w \). In either case, \( L(M_w) = \{ yy : y \in \Sigma^* \} \), a non-context free language. Hence, \( R \) rejects \( \langle M_w \rangle \). Thus, \( S \) rejects \( \langle M, w \rangle \).

Therefore, \( S \) decides \( A_{\text{TM}} \). \( \square \)
Theorem. The language

\[ F_{\text{FINITE}} = \{ \langle M \rangle : M \text{ is a TM and } L(M) \text{ is finite} \} \]

is undecidable.

Proof. By reduction from \( A_{\text{TM}} \). Suppose \( F_{\text{FINITE}} \) is decidable and let \( R \) be a TM that decides it. We'll construct a TM \( S \) to decide \( A_{\text{TM}} \), which will give the desired contradiction since \( A_{\text{TM}} \) is known to be undecidable. Our TM \( S \) works as follows. \( S= \) “On input \( \langle M, w \rangle \):

1. Construct an encoding \( \langle M_w \rangle \) of a TM \( M_w \) such that
   \( M_w = \) "On input x:
   (a) Run \( M \) on \( w \). If \( M \) accepts \( w \), accept. If \( M \) rejects \( w \), reject.'

2. Run \( R \) on \( \langle M_w \rangle \).

3. If \( R \) accepts, reject. If \( R \) rejects, accept.”

It remains to be shown that \( S \) in fact decides \( A_{\text{TM}} \). So suppose \( \langle M, w \rangle \) is an input to \( S \). First suppose \( M \) accepts \( w \). Then \( M_w \) accepts all input, i.e., \( L(M_w) = \Sigma^* \), an infinite language. Hence, \( R \) rejects \( \langle M_w \rangle \). Thus, \( S \) accepts \( \langle M, w \rangle \).

Next suppose \( M \) does not accept \( w \), i.e., \( M \) either rejects or loops on \( w \). In either case, \( L(M_w) = \emptyset \), a finite language. Hence, \( R \) accepts \( \langle M_w \rangle \). Thus, \( S \) rejects \( \langle M, w \rangle \).

Therefore, \( S \) decides \( A_{\text{TM}} \). \( \square \)
Theorem. The language

\[ \text{DECIDER}_{\text{TM}} = \{ \langle M \rangle : \text{TM } M \text{ is a decider} \} \]

is undecidable.

Proof. By reduction from \( \text{HALT}_{\text{TM}} \). Suppose \( \text{DECIDER}_{\text{TM}} \) is decidable and let \( R \) be a TM that decides it. We’ll construct a TM \( S \) to decide \( \text{HALT}_{\text{TM}} \), which will give the desired contradiction since \( \text{HALT}_{\text{TM}} \) is known to be undecidable. Our TM \( S \) works as follows.

\( S \) = “On input \( \langle M, w \rangle \), where \( M \) is a TM and \( w \) is a string:

1. Construct an encoding \( \langle M_w \rangle \) of a TM \( M_w \) that works as follows.

   \( M_w = \) ‘On input string \( x \):

   (a) If \( x \neq w \), accept.

   (b) If \( x = w \),

      i. Run \( M \) on \( w \).

      ii. If \( M \) accepts, accept. If \( M \) rejects, reject.’

2. Run \( R \) on \( \langle M_w \rangle \).

3. If \( R \) accepts, accept. If \( R \) rejects, reject.”

It remains to be shown that \( S \) in fact decides \( \text{HALT}_{\text{TM}} \). So suppose \( \langle M, w \rangle \) is an input to \( S \).

First suppose \( M \) halts on \( w \). Then \( M_w \) halts on all input \( x \). Therefore, \( M_w \) is a decider. Hence, \( R \) accepts \( \langle M_w \rangle \). Thus, \( S \) accepts \( \langle M, w \rangle \).

Next suppose \( M \) loops on \( w \). Then \( M_w \) loops on some input, specifically, it loops on \( w \). So \( M_w \) is not a decider. Hence, \( R \) rejects \( \langle M_w \rangle \). Thus, \( S \) rejects \( \langle M, w \rangle \).

Therefore, \( S \) decides \( \text{HALT}_{\text{TM}} \). \( \square \)