Problem 1

REVERSE = \{ M \mid M \text{ is a TM with the property: for all } w, M(w) \text{ accepts iff } M(w^R) \text{ accepts}\}.

REVERSE is undecidable.

Problem 2.1 UNDECIDABLE

\{ (M, w) \mid M \text{ is a TM that on input } w, \text{ tries to move its head past the left end of the tape} \}

Problem 2.2 DECIDABLE

\{ (M, w) \mid M \text{ is a TM that on input } w, \text{ moves its head left at least once, at some point} \}

Problem 2.1 UNDECIDABLE

\{ (M, w) \mid M \text{ is a TM that on input } w, \text{ tries to move its head past the left end of the tape} \}

Proof: Reduce from \( A_{\text{TM}} \) to the above language

On input \((M, w)\), make a TM \( N \) that marks the leftmost tape cell, shifts input \( w \) over one square, then simulates \( M(w) \). If \( M \) moves to the marked cell, \( N \) moves the head back to the right. If \( M \) accepts, \( N \) tries to move its head past the left end of the tape.

\((M, w)\) is in \( A_{\text{TM}} \) if and only if \((N, w)\) has the property
Problem 2.2 DECIDABLE

\((M, w) \mid M\) is a TM that on input \(w\), moves its head left at least once, at some point\)

On input \((M, w)\), run the machine for \(|Q_M| + |w| + 1\) steps:

- **Accept** if \(M\)’s head moved left at all
- **Reject** otherwise

*(Why does this work??)*

Problem 3

Let \(L\) be a language over Turing machines. Assume that \(L\) satisfies the following properties:

1. (Semantic) For any TMs \(M_1\) and \(M_2\), where \(L(M_1) = L(M_2)\), \(M_1 \in L\) if and only if \(M_2 \in L\)

2. (Nontrivial) There are TMs \(M_{YES}\) and \(M_{NO}\), where \(M_{YES} \in L\) and \(M_{NO} \notin L\)

Prove that \(L\) is undecidable

Examples and Non-Examples

Semantic Properties P

- \(M\) accepts \(\varepsilon\)
- \(L(M) = \{\varepsilon\}\)
- \(L(M)\) is empty
- \(L(M)\) is regular
- \(M\) accepts exactly 154 strings

Not Semantic!

- \(M\) halts and rejects \(\varepsilon\)
- \(M\) tries to move its head off the left end of the tape, on input \(\varepsilon\)
- \(M\) never moves its head left on input \(\varepsilon\)
- \(M\) has exactly 154 states
- \(M\) halts on all inputs

Let \(L = \{M \mid P(M)\) is true\}

is undecidable

Rice’s Theorem

Let \(L\) be a language over Turing machines. Assume that \(L\) satisfies the following properties:

1. (Semantic) For any TMs \(M_1\) and \(M_2\), where \(L(M_1) = L(M_2)\), \(M_1 \in L\) if and only if \(M_2 \in L\)

2. (Nontrivial) There are TMs \(M_{YES}\) and \(M_{NO}\), where \(M_{YES} \in L\) and \(M_{NO} \notin L\)

Then \(L\) is undecidable

*“Every nontrivial semantic property of Turing machines is undecidable”*

Extremely Powerful Stuff
Theorem: There is a computable function

\begin{align*}
&\rightarrow \\
L(M) &\text{ contains at least 154 strings}
\end{align*}

Examples and Non-Examples

\begin{itemize}
  \item \(L(M)\) contains at most 154 strings
  \item \(L(M)\) contains at least 154 strings
\end{itemize}

Is there a generic condition for unrecognizability?

Rice's Theorem, Part II

Let \(L\) be a language over Turing machines. Assume that \(L\) satisfies the following properties:

1. (Semantic) For any TMs \(M_1\) and \(M_2\), where \(L(M_1) = L(M_2)\), \(M_1 \in L\) if and only if \(M_2 \in L\).

2. (Non-monotone) There are TMs \(M_{\text{YES}}\) and \(M_{\text{NO}}\), where \(M_{\text{YES}} \in L\), \(M_{\text{NO}} \not\in L\), and \(L(M_{\text{YES}}) \subset L(M_{\text{NO}})\).

Then \(L\) is unrecognizable.

“Every non-monotone semantic property of Turing machines is unrecognizable.”

Idea: Give a mapping reduction from \(-A_{TM}\) to \(L\).

Examples and Non-Examples

<table>
<thead>
<tr>
<th>Monotone Properties (P)</th>
<th>Non-Monotone</th>
</tr>
</thead>
<tbody>
<tr>
<td>(L(M)) is infinite</td>
<td>(L(M)) is finite</td>
</tr>
<tr>
<td>(L(M) = \Sigma^*)</td>
<td>(L(M) = {\emptyset})</td>
</tr>
<tr>
<td>(L(M)) contains at least 154 strings</td>
<td>(L(M)) is regular</td>
</tr>
<tr>
<td>(L(M)) contains 11111</td>
<td>(L(M)) is not regular</td>
</tr>
<tr>
<td></td>
<td>(L(M)) contains at most 154 strings</td>
</tr>
</tbody>
</table>

Monotone: \(\forall M_{\text{YES}}, M_{\text{NO}},\)
If \(M_{\text{YES}} \in L\)
and \(L(M_{\text{YES}}) \subset L(M_{\text{NO}})\)
then \(M_{\text{NO}} \in L\).

\(L = \{M \mid P(M) \text{ is true}\}\)
is unrecognizable.

Reduction from \(-A_{TM}\): On input \((M,w)\):

Output \(M_w(x) := \text{Run } M_{\text{YES}}(x), M_{\text{NO}}(x), M(w)\) in \(\|\)
If \((M \text{ accepts } w) \& (M_{\text{NO}} \text{ accepts } x)\), ACCEPT
If \((M_{\text{YES}} \text{ accepts } x)\), ACCEPT’

If \(M\) accepts \(w\), then \(L(M_w) = L(M_{\text{NO}})\), since \(L(M_{\text{YES}}) \subset L(M_{\text{NO}})\). We have \(M_w \not\in L\).

If \(M\) does not accept \(w\), then \(L(M_w) = L(M_{\text{YES}})\) Since \(M_{\text{YES}} \in L\), we have \(M_w \in L\)

\((M, w) \in A_{TM} \text{ if and only if } M_w \not\in L\)

Self-Reference and the Recursion Theorem

\begin{align*}
\text{Theorem: There is a computable function} \\
q : \Sigma^* \rightarrow \Sigma^* \text{, where for any string } w, \\
q(w) \text{ is the description of a TM } P_w \text{ that on any input, prints out } w \text{ and then accepts}
\end{align*}

\begin{align*}
w &\rightarrow Q \rightarrow P_w \\
P_w &\downarrow \\
w &\rightarrow w
\end{align*}
Another Way of Looking At It

Suppose in general we want to design a program that prints its own description. How?

“Print this sentence.”

Print two copies of the following, the second copy in quotes:

“Print two copies of the following, the second copy in quotes:”

The Recursion Theorem

Theorem: Let $T$ be a Turing machine that computes a function $t : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$. There is a Turing machine $R$ that computes a function $r : \Sigma^* \rightarrow \Sigma^*$, where for every string $w$,

$$r(w) = t(R, w)$$

- $(a, b) \rightarrow T \rightarrow t(a, b)$
- $w \rightarrow R \rightarrow t(R, w)$