Exercise Sheet 2
Undecidability and Rice’s Theorem

David Carral

October 23, 2019
Exercise 1. Using an oracle that decides the halting problem, construct a decider for the language $\{\langle M, w \rangle | M$ is a TM that accepts $w \}$. 

**Definition.** An *Oracle Turing Machine* (OTM) is a TM $M$ with a special tape, called the *oracle tape*, and distinguished states $q_?$, $q_{yes}$, and $q_{no}$. For a language $O$, the *oracle machine* $M^O$ can, in addition to the normal TM operations, do the following: Whenever $M^O$ reaches $q_?$, its next state is $q_{yes}$ if the content of the oracle tape is in $O$, and $q_{no}$ otherwise.

**Solution.**

- Let $H = \{\langle M, w \rangle | M$ is a TM that halts on input $w \}$.
- We define an oracle machine $N^H$ that, on input $\langle M, w \rangle$, does the following:
  - Construct the TM $M'$, which is obtained by extending $M$ in the following manner:
    - Add a fresh state $q_\infty$ to $M'$.
    - For every tape symbol $a$, add $\langle q_\infty, a \rangle \mapsto \langle q_\infty, a, R \rangle$ to the transition function of $M'$.
    - For every non-accepting state $q$ and every tape symbol $a$ such that $\delta(q, a)$ is undefined, add $\langle q, a \rangle \mapsto \langle q_\infty, a, R \rangle$ to the transition function of $M'_\infty$.
  - Use the oracle tape of $N^H$ to determine whether $M'$ halts with input $w$. If that is the case, output *accept*; otherwise, *reject*. 
Exercise 2. A *useless state* in a Turing machine is one that is never entered on any input string. Consider the problem of determining whether a Turing machine has any useless states. Show that this language is undecidable.

Solution.

- Suppose for a contradiction that there is a TM $\mathcal{U}$ such that $L(\mathcal{U}) = \{\langle M \rangle \mid M \text{ contains some useless state}\}$.
- Using $\mathcal{U}$, we construct a TM $\mathcal{E}$ that solves the empty word problem (which is undecidable).
- On input $\langle M \rangle$, the TM $\mathcal{E}$ performs the following computation:
  - Write down the encoding of a TM $M'$ that (1) deletes the content in the input string, (2) places the head at the beginning of the tape, and (3) executes $M$. Note that, $M$ accepts the empty word iff $L(M') = \Sigma^*$ iff $L(M') \neq \emptyset$.
  - Produce the encoding of a TM $M''$ which results from pruning all useless states in $M'$. Note that, we can construct this TM using $\mathcal{U}$.
  - *Accept* iff $M''$ contains some final state.

Remark. We assume that once a TM goes into a final state it halts and accepts.
Exercise 3. Show the following: “If a language $L$ is Turing-recognisable and $\overline{L}$ is many-one reducible to $L$, then $L$ is decidable.”

Remark. $\overline{L} = \{w \mid w \notin L\}$

Definition. Consider some languages $P$ and $Q$ defined over the alphabet $\Sigma$. Then, $P$ is many-one reducible to $Q$ if there exists a total computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that $w \in P$ iff $f(w) \in Q$ for all $w \in \Sigma^*$.


(a) Premise: $L$ is Turing-recognisable.

(b) Premise: $\overline{L}$ is many-one reducible to $L$.

(c) By (a) and (b): $\overline{L}$ is Turing-recognisable.

(d) By (a) and (c): $L$ and $\overline{L}$ can be enumerated.

(e) By (d): Given some word $w$, we can enumerate all words in $L$ and $\overline{L}$ in parallel. Eventually, we will be able to determine whether $w$ is in $L$ or not.
Exercise Sheet 2

Exercise 4. Let \( L = \{\langle M \rangle \mid M \text{ a TM that accepts } w^r \text{ whenever it accepts } w \} \), where \( w^r \) is the word \( w \) reversed. Show that \( L \) is undecidable.


- Let \( \mathcal{P} \) be the property containing a language \( L' \) iff \( w \in L' \iff w^r \in L' \) for every \( w \in \Sigma^* \). Note that, a property is a set of languages, i.e., a set of sets of words.
- \( L \) is the set of all TM encodings \( \langle M \rangle \) that accept some language in \( \mathcal{P} \).
- Since \( \mathcal{P} \) is non-trivial, \( L \) is undecidable by Rice’s Theorem.

Definition 4.2: Let \( \mathcal{P} \) be a set of languages. A language \( L \) has the property \( \mathcal{P} \) if \( L \in \mathcal{P} \). Property \( \mathcal{P} \) is a non-trivial property of recognisable languages if there are TM-recognisable languages that have it and others that do not have it.

Theorem 4.1 (Rice’s Theorem): If \( \mathcal{P} \) is a non-trivial property of recognisable languages, then the following problem is undecidable:

\[
\mathcal{P}\text{-ness} = \{\langle M \rangle \mid L(M) \in \mathcal{P}\}
\]
Exercise Sheet 2

Exercise 5. Consider the following languages $L$ and $L'$:

$$L = \{ \langle M, w \rangle \mid M \text{ accepts } w \} \quad L' = \{ \langle M \rangle \mid M \text{ does not accept any word} \}$$

Show that there cannot exist a many-one reduction from $L$ to $L'$.

Definition. Consider some languages $P$ and $Q$ defined over the alphabet $\Sigma$. Then, $P$ is many-one reducible to $Q$ if there exists a total computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that $w \in P$ iff $f(w) \in Q$ for all $w \in \Sigma^*$.

Solution.

(a) Suppose for a contradiction that $L \leq_m L'$.

(b) By (a): $\overline{L} \leq_m \overline{L'}$.

(c) $\overline{L'}$ is semi-decidable (see Exercise 10 on the previous exercise sheet).

(d) By (b) and (c): $\overline{L}$ is semi-decidable.

(e) $L$ is semi-decidable (discuss).

(f) By (d) and (e): $L$ is decidable.

(g) $L$ is undecidable (discuss).

(h) By (f) and (g): Contradiction!
Exercise Sheet 2

Exercise 6. Show that every Turing-recognisable language can be mapping-reduced to the following language.

\[ L = \{ \langle M, w \rangle \mid M \text{ is a TM that accepts the word } w \} \]

Definition. Consider some languages \( P \) and \( Q \) defined over the alphabet \( \Sigma \). Then, \( P \) is \textit{many-one reducible} to \( Q \) if there exists a total computable function \( f : \Sigma^* \rightarrow \Sigma^* \) such that \( w \in P \) iff \( f(w) \in Q \) for all \( w \in \Sigma^* \).

Solution.
(a) Let \( L' \) be a semi-decidable language.
(b) By (a): There is some TM \( M \) that recognises \( L' \).
(c) Let \( f \) be the Turing-computable function mapping a word \( w \) to \( \langle M, w \rangle \).
(d) By (c): For every \( w \in \Sigma^* \), \( w \in L' \) iff \( \langle M, w \rangle \in L \).
(e) By (c) and (d): \( L' \leq_m L \).