Tutorial 3: Decidability and Undecidability

CSC 463

February 7, 2020

1. Recall that Rice’s theorem states that if \( P \) is a non-trivial set of Turing machine encodings \( \langle M \rangle \) satisfying the condition that if \( M_1, M_2 \) are Turing machines with \( L(M_1) = L(M_2) \), then \( \langle M_1 \rangle \in P \) if and only if \( \langle M_2 \rangle \in P \). ¹, we can conclude that \( P \) is an undecidable set.

Prove Rice’s Theorem using the following steps.

(a) Explain why we can assume that \( P \) does not contain any Turing machine \( M \) with \( L(M) = \emptyset \) and there is a Turing machine \( M' \in P \) with \( L(M') \neq \emptyset \). (Hint: Recall that if \( A \) is decidable, so is the complement \( \bar{A} \).)

**Solution:** If \( P \) does contain a Turing machine with \( L(M) = \emptyset \), we know that by the conditions given that the complement \( \bar{P} \) is non-empty and does not contain any Turing machine with \( L(M) = \emptyset \). So replace \( P \) by \( \bar{P} \) in this case, and undecidability of \( \bar{P} \) implies undecidability of \( P \).

(b) Construct a reduction \( A_{TM} \leq_m P \) so that reduction maps instances \( \langle M, w \rangle \) to Turing machines \( \langle N \rangle \in P \), using \( M' \) from part (a), if and only if \( M \) accepts \( w \).

**Solution:** Let \( \langle M, w \rangle \) be an encoding of a Turing machine and a string \( w \in \Sigma^* \). By part (a), we also know that there is a Turing machine \( \langle M' \rangle \in P \) with \( L(M') \neq \emptyset \) and \( P \) does not contain any Turing machine with an empty language. Given a pair \( \langle M, w \rangle \), we can construct a Turing machine \( N \) with the following description:

i. On input \( x \), firstly simulate \( M \) on input \( w \).
ii. If \( M \) accepts, simulate \( M' \) on input \( x \) and accept \( x \) if \( M' \) accepts.
iii. If \( M \) rejects, reject \( x \).

Note that if \( M \) accepts \( w \), we have \( L(N) = L(M') \) and the assumption that \( \langle M' \rangle \in P \) implies that \( \langle N \rangle \in P \). Otherwise, if \( M \) rejects or loops on \( w \), we have \( L(N) = \emptyset \) and part (a) implies that \( \langle N \rangle \notin P \). Hence we have constructed a reduction from \( A_{TM} \) to \( P \).

(c) Conclude Rice’s Theorem from the reduction in part (b).

**Solution:** Since \( A_{TM} \) is undecidable, so is \( P \).

(d) Let \( L \) be the set of Turing machine encodings \( \langle M \rangle \) with less than 200 states. Show that \( L \) is decidable. Why does this not contradict Rice’s theorem?

**Solution:** We can design a Turing machine to read the encoding \( \langle M \rangle \) and count the number of states it has from the encoding, so \( L \) is decidable. This result does not contradict Rice’s theorem since even though \( L \) is non-trivial language consisting of Turing machine encodings, the second condition of Rice’s theorem does not apply and so we cannot use it to conclude anything about the decidability of \( L \). There are Turing machines \( M_1, M_2 \) deciding the same language, with \( M_1 \) having more than 200 states and \( M_2 \) having less than 200 states.

¹In other words, whether or not \( \langle M \rangle \in P \) depends on its language \( L(M) \) only
2. Let $T = \{ \langle M \rangle : M$ is a Turing machine with $|L(M)| = 3 \}$. Prove that $T$ is not semidecidable.

**Solution:** We construct a reduction from $A_{TM}$ to $T$ to show that $T$ is not semidecidable. Given a pair $\langle M, w \rangle$, let $M'$ be the Turing machine with the following description:

(a) On input $x$, if $x \in \{\epsilon, 0, 1\}$ accept.
(b) Otherwise, run $M$ on input $w$. Accept $x$ if $M$ accepts and reject $x$ if $M$ rejects.

If $M$ accepts $w$, then $L(M') = \Sigma^*$. Otherwise, $L(M') = \{\epsilon, 0, 1\}$. So $\langle M, w \rangle \in A_{TM}$ if and only if $\langle M' \rangle \notin T$. We have constructed a reduction $\overline{A_{TM}} \leq_m T$, so this shows that $T$ is not semidecidable since $\overline{A_{TM}}$ is not semidecidable.

3. Let $G_1$ and $G_2$ be context-free grammars. Show that the problem of testing whether or not $L(G_1) \subset L(G_2)$ is undecidable. You may assume that testing whether or not $L(G) = \Sigma^*$ for a context-free grammar $G$ is undecidable.

**Solution:** We give a reduction from $\text{ALL}_{CFG} = \{ \langle G \rangle : G$ is a CFG with $L(G) = \Sigma^* \}$ to $\text{S}_{CFG} = \{ \langle G_1, G_2 \rangle : G_1, G_2$ are CFGs and $L(G_1) \subset L(G_2) \}$. We map every encoding of a context-free grammar $G$ to $\langle A, G \rangle$ where $A$ is the context-free grammar with a single variable $S$ and rules $S \rightarrow a_i S$ for all $a_i \in \Sigma$, and $S \rightarrow \epsilon$. Note that $L(A) = \Sigma^*$, so if $L(A) \subset L(G)$, then $L(G) = \Sigma^*$. Hence, $\text{S}_{CFG}$ is undecidable as we have shown a reduction $\text{ALL}_{CFG} \leq_m \text{S}_{CFG}$.

4. Assume that $\Gamma = \{0, 1, \sqcup\}$ is the tape alphabet for all Turing machines in this problem. The **busy beaver function** $BB : \mathbb{N} \rightarrow \mathbb{N}$ is defined as follows. Let $M_k$ be the set of $k$-state Turing machines that halt when started on a blank tape. Define $BB(k)$ to be the maximum number of ones remaining on the tape when a machine $M \in M_k$ started on a blank tape halts.

Show that $BB$ is not a computable function (i.e. there is no Turing machine $M$ which on input $k$ in unary, outputs $BB(k)$ in unary and halts). You may assume that $BB$ is a strictly increasing function.

**Solution:** Assume that there is a Turing machine $M$ that computes $BB$. Then there is a Turing machine $M'$ that starting from a blank tape:

(a) Writes $k$ in unary.
(b) Doubles the number on its tape to obtain $2k$ in unary.
(c) Simulates $M$ to compute $BB(2k)$

By our assumption about $M$, $M'$ is a well-defined Turing machine that has $\leq k + c$ states for some constant $c$ (from Steps (b) - (c)) and a constant $k$ that we can choose (from Step (a)). Furthermore, the output produced has $BB(2k)$ ones. Since $M'$ halts starting from a blank tape and has $k + c$ states, by definition of $BB$, we have $BB(2k) \leq BB(k + c)$. But this is a contradiction choosing $k = c + 1$, since $BB$ is a strictly increasing function. So $BB$ cannot be a computable function.