3P17/4P17 COMPUTABILITY 2014-2015 Lecture Handout 9

Further Reducibility

Theorem 1. *The language*

$$E_{TM} = \{\langle M \rangle : M \text{ is a Turing machine and } L(M) = \emptyset\}$$

is undecidable.

Proof. Recall

$$A_{TM} = \{\langle M, \sigma \rangle : M \text{ is a Turing machine which accepts } \sigma\}.$$

We will reduce the problem of deciding the language A_{TM} to the problem of deciding the language E_{TM} and derive a contradiction.

Suppose that there exists a TM M that decides E_{TM}. Then we can construct a Turing machine M' that decides A_{TM}:

$\underline{M'}$

Check if input string is of form $\langle M_1, \sigma \rangle$ where M_1 is a Turing machine and σ is input to M_1. If not reject. If so, do the following:

(1) Construct string $\langle M_\sigma \rangle$.

(2) Simulate M with input $\langle M_\sigma \rangle$. If M rejects, then accept. If M accepts, then reject.

$\underline{M_\sigma}$

On input string x do the following:

(1) Compare x to σ. If $x \neq \sigma$ the reject. If $x = \sigma$ go to (2)

(2) Simulate M_1 on σ. If M_1 accepts, then accept. If it rejects, then reject.

Note that M_σ merely performs a string comparison then halts or simulates M_1. Thus, the string $\langle M_\sigma \rangle$ can be computed given the string $\langle M_1, \sigma \rangle$.

If $\langle M, \sigma \rangle \in A_{TM}$ then $L(M_\sigma) = \{\sigma\} \neq \emptyset$, so M will reject $\langle M_\sigma \rangle$ and M' will accept $\langle M_1, \sigma \rangle$. If $\langle M, \sigma \rangle \notin A_{TM}$ then $L(M_\sigma) = \emptyset$, so M will accept $\langle M_\sigma \rangle$ and M' will reject $\langle M_1, \sigma \rangle$.

Hence, M' decides A_{TM}. But we know that no such M' exists, so M cannot exist either, and E_{TM} is undecidable. □

Theorem 2. *The language*

$$EQ_{TM} = \{\langle M_1, M_2, \rangle\} : M_1 \text{ and } M_2 \text{ are Turing machines and } L(M_1) = L(M_2)\}$$

is undecidable.

Proof. We will reduce the problem of deciding E_{TM} to the problem of deciding EQ_{TM} and derive a contradiction.

Suppose that there exists a Turing machine M that decides EQ_{TM}. Then we can construct a Turing machine M' that decides E_{TM}:

<u>M'</u>
Check if input string is of form $\langle M_1 \rangle$ where M_1 is a Turing machine. If not reject. If so, do the following:

(1) Construct the string $\langle M_1, M_\emptyset \rangle$ where $M_\emptyset$ is a TM that rejects all input.

(2) Simulate M with input$\langle M_1, M_\emptyset \rangle$. If it accepts, then accept. If it rejects, then reject.

If $\langle M_1 \rangle \in E_{TM}$ then $L(M_1) = \emptyset = L(M_\emptyset)$ and M will accept, so M' will accept. If $\langle M_1 \rangle \notin E_{TM}$ then $L(M_1) \neq \emptyset = L(M_\emptyset)$ and M will reject, so M' will reject.

Hence, M' decides E_{TM}. But we know that no such M' exists, so M cannot exist either, and EQ_{TM} is undecidable. □

The following theorem generalises the above and says that for any non-trivial property P of a language, testing whether or not a Turing machine has that property is undecidable.

Theorem 3 (**Rice's theorem**). *Let $ALL_{TM} = \{\langle M \rangle : M \text{ is a Turing machine}\}$ be the language over an alphabet Σ that contains all Turing machine description (encodings). Suppose P_{TM} is a language over Σ that satisfies the following conditions:*

(i) *P_{TM} is a non-empty, strict subset of ALL_{TM}, that is, P_{TM} is non-trivial.*

(ii) *For any Turing machines M_1, M_2, if $L(M_1) = L(M_2)$, then either $\langle M_1 \rangle \in P_{TM}$ and $\langle M_2 \rangle \in P_{TM}$, or $\langle M_1 \rangle \notin P_{TM}$ and $\langle M_2 \rangle \notin P_{TM}$. That is, a pair of equivalent Turing machines are either both in or both not in P_{TM}.*

Then P_{TM} is undecidable.

Proof. We will reduce the problem of deciding A_{TM} to the problem of deciding P_{TM} and derive a contradiction.

Let $M_\emptyset$ be a Turing machine that always rejects, so $L(M_\emptyset) = \emptyset$. We may assume $\langle M_\emptyset \rangle \notin P_{TM}$, since if it were, and P_{TM} being decidable, we could continue the proof for $\overline{P}$, which must also be decidable and will satisfy conditions **(i)** and **(ii)**.

Since P_{TM} is non-trivial, there is some Turing machine with $\langle M_1 \rangle \in P_{TM}$.

Suppose that there exists a Turing machine M that decides P_{TM}. Then we can construct a Turing machine M' that decides A_{TM}:

<u>M'</u>
Check if input string is of form $\langle M_2, \sigma \rangle$ where M_2 is a Turing machine. If not reject. If so, do the following:

(1) Construct the string $\langle M_\sigma \rangle$.

(2) Simulate M with input $\langle M_\sigma \rangle$. If it accepts, then accept. If it rejects, then reject.

<u>M_σ</u>
On input x do the following:

(1) Simulate M_2 on σ. If it rejects, then reject. If it accepts, then go to (2)

(2) Simulate M_1 on x. If it accepts, then accept. If it rejects, then reject.

Note that because P_{TM} is assumed to be decidable, its members can be enumerated (Examples Sheet 2, Question 3), so a $\langle M_1 \rangle \in P_{TM}$ can be found by M_σ.

Now if $\langle M_2, \sigma \rangle \in A_{TM}$, then $L(M_\sigma) = L(M_1)$ since M_σ will always pass its input to M_1 and give the same verdict.

Therefore, by **(ii)** $\langle M_\sigma \rangle \in P_{AT}$ so M will accept it, meaning M' will accept $\langle M_2, \sigma \rangle$.

On the other hand, if $\langle M_2, \sigma \rangle \notin A_{TM}$, then either M_2 rejects σ or it loops on σ.

If M_2 rejects σ then M_σ rejects whatever input it gets.

If M_2 loops on σ then M_σ loops on whatever input it gets.

In either case, $L(M_s) = \emptyset = L(M_\emptyset)$.

By assumption, $\langle M_\emptyset \rangle \notin P_{TM}$, so by **(ii)** $\langle M_\sigma \rangle \notin P_{TM}$.

Thus, M will reject $\langle M_\sigma \rangle$, meaning M' will reject $\langle M_2, \sigma \rangle$.

Thus, we see that M' decides A_{TM}.

This contradicts our previous theorem, so M' does not exist and we conclude P_{TM} is undecidable. □

Applying Rice's theorem with $P_{TM} = E_{TM}$, we have an alternative proof of Theorem 1: Some Turing machines reject all input, some don't, so condition **(i)** is satisfied. If two Turing machines recognise the same language, then they both either recognise and empty language or both recognise a non-empty one, meaning they are either both in or both not in $P_{TM} = E_{TM}$. Hence, condition **(ii)** is satisfied. We conclude that E_{TM} is undecidable.