# Computability and Complexity: Exercise Session 3 (2017-10-18)

Aurélien Ooms*

October 21, 2017

## Contents



# 1 Exercise 4.30[1]

Let $A$ be a Turing-recognizable language consisting of descriptions of Turing machines, $\{\langle M_1\rangle, \langle M_2\rangle, \ldots\}$, where every $M_i$ is a decider. Prove that some decidable language $Q$ is not decided by any decider $M_i$ whose description appears in $A$. (Hint: You may find it helpful to consider an enumerator for $A$.)

## 1.1 Solution

The solution is in two steps. First we construct a new language $Q$ that is not decided by any of the $M_i$ using the technique of diagonalization. Second we show how to build a decider for $Q$ using the fact that $A$ is Turing-recognizable.

Table 1 shows a diagonalization that is similar to the ones we saw during a previous lecture. We build a new language $Q$ that disagrees with each language decided by a decider $M_i$, that is, $Q$ disagrees with $L(M_i)$ on $w_i$. Hence, none of the $M_i$ decides $Q$.

We assume, without loss of generality, that the order of the $w_i$ is the standard string order[2], that is, first ordered by length, then lexicographically. For a given finite alphabet $\Sigma$, there are finitely many words of finite length, hence there is an algorithm $W$ that enumerates all the words of $\Sigma^*$ in the standard string order.

Since $A$ is Turing-recognizable, there exists an enumerator for $A$, as we have shown in a previous lecture.

To prove that $Q$ is decidable, we build the following TM:

$D =$ on input $w_i$:

**1.** Determine the value of $i$ using the enumerator $W$, that is, enumerate all words in standard string order while maintaining a count of how many words have been enumerated, and stop when encountering $w_i$.

*aureooms@ulb.ac.be

[1] Exercises from the reference book: Sipser M., *Introduction to the Theory of Computation*, 3rd edition (2013). In the second edition of the book, this exercise is Exercise 4.28.

[2] This string order is called shortlex or quasi-lexicographic order but is refered to as "lexicographic order" in the textbook. See https://en.wikipedia.org/wiki/Lexicographical_order and https://en.wikipedia.org/wiki/Shortlex_order.

Table 1: Building the language $Q$. Entry $(M, w)$ is 1 if $w \in L(M)$ and 0 otherwise. $w_i \in Q$ iff $w_i \notin L(M_i)$.

| | $w_1$ | $w_2$ | $w_3$ | $w_4$ | $\dots$ | $w_i$ | $\dots$ |
|---|---|---|---|---|---|---|---|
| $Q$ | 0 | 1 | 1 | 0 | $\dots$ | 0 | $\dots$ |
| $M_1$ | 1 | 1 | 1 | 1 | $\dots$ | 1 | $\dots$ |
| $M_2$ | 0 | 0 | 0 | 0 | $\dots$ | 0 | $\dots$ |
| $M_3$ | 1 | 0 | 0 | 0 | $\dots$ | 0 | $\dots$ |
| $M_4$ | 0 | 1 | 0 | 1 | $\dots$ | 0 | $\dots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ |
| $M_i$ | 1 | 0 | 1 | 0 | $\dots$ | 1 | $\dots$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ | $\ddots$ |

**2.** Enumerate the $M_k$ of $A$ using the enumerator for $A$. Stop once we have enumerated $i$ different deciders, that is, the last decider yielded by the enumerator is $M_i$.

**3.** Simulate $M_i$ on $w_i$. If $M_i$ accepts, reject. If $M_i$ rejects, accept.

$D$ is a decider since:

1. Steps 1 and 2 execute a finite number of steps since $i$ is finite.
2. Step 3 stops after a finite number of steps since $M_i$ is a decider.

$D$ accepts $w_i$ iff $M_i$ rejects $w_i$. Since $D$ and $M_i$ are deciders, this is equivalent to say that $w_i \in L(D)$ iff $w_i \notin L(M_i)$, hence $L(D) = Q$. $D$ is thus a decider for $Q$ and hence $Q$ is decidable.