Universität Zürich
Institut für Informatik

*Student Name:*
*Student Number:*

Autumn 2020
Sven Seuken
Dennis Komm

# Foundations of Computing II
## Assignment 5

### Turing Machines, Undecidability

Distributed: 23.11.2019 – Due Date: 08.12.2019

Upload your solutions to the OLAT system.

### 5.1 Turing Machines

Draw Turing machines (TMs) for the following languages and briefly explain how your TMs work.

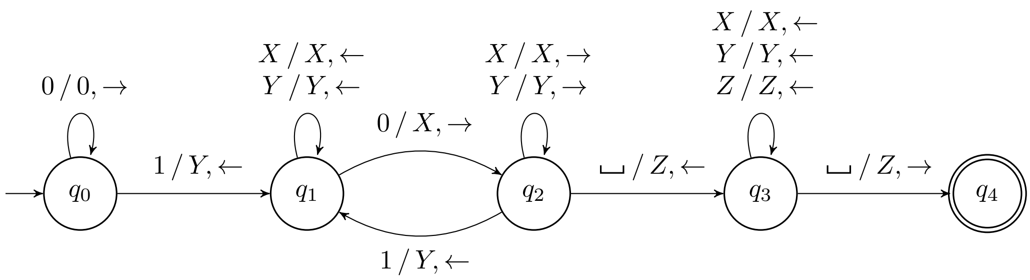
**a)** $L_1 = \{a^k b^k w \mid k \in \mathbb{N}^+ \text{ and } w \in \{a,b\}^*\}$

**b)** $L_2 = \{w0w \mid w \in \{1,2\}^*\}$

*Hint:* It is sufficient to construct TMs with one tape each. Recall that you can assume that the input word only contains letters from the implicitly given alphabet; for instance, in a), there are only letters $a$ and $b$ on the tape at the beginning.

### 5.2 Turing Machines and Configurations

Consider the following TM $M$ with $\text{Lang}(M) \subseteq \{0,1\}^*$.




**a)** Give the computation of $M$ (that is, the unique sequence of configurations) on the following words; indicate when $M$ gets stuck or accepts.

i. 01

ii. 001

iii. 101

iv. 00111

**b)** Describe Lang($M$) in words.

### 5.3 Diagonalization

Consider the following two languages.

$$L_{\text{diag},1} = \{w \in \{0,1\}^* \mid w = w_{2i} \text{ and } M_i \text{ does not accept } w_{2i}\} ,$$
$$L_{\text{diag},2} = \{w \in \{0,1\}^* \mid w = w_i \text{ and } M_{2i} \text{ does not accept } w_i\} .$$

Here, we again assume that $M_i$ is the $i$th TM in a fixed ordering and $w_i$ is the $i$th binary word in a fixed ordering over some given alphabet. We see that both of these languages are constructed in a way that reminds us of the language $L_{\text{diag}}$; the only part that is different is that we do not speak about the main diagonal of the corresponding table, but two different diagonals that are shallower or steeper, respectively.

**a)** For one of the two languages, prove that it is not recursively enumerable.

**b)** For the other language, explain why the same argument as in a) is not valid to prove that this language is also not recursively enumerable.

### 5.4 More Diagonalization

Let $L$ be some infinite language over $\{0,1\}$. Explain how we can identify a subset $L_{\text{diag},L}$ of $L$ such that $L_{\text{diag},L}$ is not recursively enumerable.

### 5.5 Reductions

$L_{\text{diag}}$ was the first language for which we showed that it is not recursively enumerable and thus not recursive. To prove that there are other languages that are not recursively enumerable or not recursive, we use reductions.

In the lecture, we showed that $L_{\text{U}}$ is not recursive and argued as follows. We know that, if $L_{\text{U}}$ were recursive, then also its complement $\overline{L}_{\text{U}}$ would be recursive. Thus, if we succeed in showing that $\overline{L}_{\text{U}}$ is not recursive, then $L_{\text{U}}$ cannot be recursive. We then reduced $L_{\text{diag}}$ to $\overline{L}_{\text{U}}$, that is, the problem of deciding whether a given word is in $L_{\text{diag}}$ to deciding whether some word is in $\overline{L}_{\text{U}}$. If then we would have a TM $\overline{U}^*$ for deciding $\overline{L}_{\text{U}}$ (that is, if this language were recursive), we could use it to decide $L_{\text{diag}}$ (that is, this language would also be recursive).

We want to slightly modify the original proof, but essentially prove the same statement.

**a)** Formally define the language $\overline{L}_{\text{diag}}$, that is, the complement of $L_{\text{diag}}$.

**b)** Prove that $\overline{L}_{\text{diag}}$ is recursively enumerable.

**c)** Reduce $\overline{L}_{\text{diag}}$ to $L_{\text{U}}$ to give an alternative proof that $L_{\text{U}}$ is not recursive.

### 5.6 More Reductions

Consider the two languages

$$L_3 = \{(M, M', w) \mid M \text{ and } M' \text{ are TMs and } w \in \text{Lang}(M) \cap \text{Lang}(M')\} ,$$
$$L_4 = \{(M, M', w) \mid M \text{ and } M' \text{ are TMs and } w \in \text{Lang}(M) \cup \text{Lang}(M')\} .$$

**a)** Show that neither $L_3$ nor $L_4$ is recursive by giving a reduction from $L_{\mathrm{U}}$.

**b)** Give a reduction from $L_3$ to $L_{\mathrm{U}}$. Do so by using that the two TMs $M$ and $M'$ can be simulated sequentially on the same word.

**c)** Point out where we run into problems for a similar reduction as in exercise part b) from $L_4$ to $L_{\mathrm{U}}$. How can we deal with this problem?

### 5.7 Yet Another Reduction

Consider TMs with exactly one accepting state and some fixed way to encode them. Using a reduction, show that the language

$$L_5 = \{(M, w, i) \mid M \text{ is a TM that visits its } i\text{th state at least once when processing } w\} .$$

is not recursive.