Test 3

1. Let $\Sigma = \{\texttt{a},\texttt{b}\}$. The language L consists of the strings with equal number of `a` and `b` (in any order, so for example, $\texttt{ab} \in L$, $\texttt{ba} \in L$ and $\texttt{bbabaaab} \in L$ and also the empty string belongs to L). This is part of the transition function of a Turing machine with 3 tapes for L.

	reading			writing			
state	tape 1	tape 2	tape 3	tape 1	tape 2	tape 3	new state
q_0	a	$*$	$*$	a R	X R	$*$ S	q_0
q_0	b	$*$	$*$	b R	$*$ S	X R	q_0
q_0	$*$	$*$	$*$	$*$ S	$*$ L	$*$ L	q_1
q_1	$*$	X	X	$*$ S	X L	X L	q_1
q_1	$*$	$*$	$*$	$*$ S	$*$ S	$*$ S	q_2

(a) What are the contents of the tapes and where are the heads when this TM reaches state q_1?

The content of tape 1 is not changed, its head is on the first * after the input. Tape 2 has as many X as the number of a in the input, tape 3 has as many X as the number of b. Head 2 and 3 are on the last X on their tapes.

(b) Continue the table so that the resulting TM accepts the described language. Do it with the restriction that the head on the first tape is not allowed to move to Left, but the other heads can make left moves. (You do not have to use all the empty rows of the table and you can add some rows if you need.) What is the set of accept states? Also, explain in English, how your TM works,

In q_0 the right amount of X are written on tape 2 and 3. In q_1 nothing happens on the input tape, on the other two we simultaneously move backward. State q_2 is reached if these two tapes had the same number of X, so let q_2 be the only accept state.
This works also when the input is empty: we just change from q_0 to q_1 without writing any X, and then to q_2, that is accepting.

Neptun: **Name: SOLUTIONS**

2. (a) Define the diagonal language (L_d).

$L_d = \{w : w$ is a description of a TM and $w \notin L(M_w)\}$

(b) When do we say that a language is recognizable (i.e. is in the class RE)?

L is recognizable iff there is a Turing machine M that accepts L, i.e. $L(M) = L$.

(c) Give a proof that the diagonal language is not recognizable.

Assume that L_d is recognizable and let M be a TM, such that $L(M) = L_d$. Let w denote the description of M, so $M_w = M$.

If $w \in L_d$ then, by the definition of the diagonal language, $w \notin L(M_w)$. Since $M_w = M$, this means that $w \notin L(M) = L_d$ which contradicts to $w \in L_d$.

On the other hand, if $w \notin L_d$ then, by the definition of the diagonal language, because w is a description of a TM, $w \in L(M_w)$. Since $M_w = M$, this means that $w \in L(M) = L_d$ which contradicts to $w \notin L_d$..

3. Let L be the language of all Turing machine descriptions $w \in \Sigma^*$ for which $L(M_w)$ has exactly one string of length 1, exactly one string of length 2, exactly one string of length 3, etc.
Prove that this language is undecidable.

Notice that the assumption " has exactly one string of length 1, exactly one string of length 2, exactly one string of length 3, etc." is a language property. Let this property be denoted by P. Then $L = L_P = \{w : w \text{ is a description of a TM and } L(M_w) \in P\}$

By the theorem of Rice, if P is a nontrivial property of languages then L_P is undecidable, so we are done.

This property is not trivial:
There is a language $L_1 \in \mathrm{RE}$ that has it, for example $L_1 = \{\mathsf{a}^n : n \geq 1\}$. Clearly this language has one word for each length, and because it is a regular language (it has a FA), it also has TM.

There is a language $L_2 \in \mathrm{RE}$ that does not have it, for example $L_2 = \{\mathsf{a}, \mathsf{b}\}^*$ has 2 strings of length 1, 4 of length 2, etc. and we have already seen that this is in RE.

4. Consider the language L generated by the grammar

$$\begin{aligned} S &\longrightarrow \mathtt{a}S\mathtt{b}|BA \\ A &\longrightarrow \mathtt{a}A|\mathtt{a} \\ B &\longrightarrow \mathtt{b}B|\mathtt{b} \end{aligned}$$

Is it true that:

(a) L is a context-free language?

(b) $L \in R$?

(c) $L \in TIME(n)$?

Justify your answers.

(a) Yes: Since the grammar is a context-free grammar, the generated language is context-free.

(b) Yes: Context-free languages are decidable languages.

(c) Yes: For this we have to show a TM that on inputs of length n takes not more than $c \cdot n$ steps.

First we need the language. It is $L = \{\mathtt{a}^k\mathtt{b}^p\mathtt{a}^r\mathtt{b}^k : k \geq 0, p \geq 1, r \geq 1\}$.

Our TM has 2 tapes. It copies the first block of a's to tape 2. With the first b moves to a different state where it reads the b's wihout changing anything on tape 2. For the first a it moves to a new state, reads these a's without changing tape 2. Then, when it reaches the second block of b's, again in a new state, starts to compare the number of b's on tape 1 and the number of a's on tape 2 (stepping to the left on tape 2). If the two numbers are the same, i.e., it reaches the first empty character at the same time on the two tapes then it accepts, otherwise halts and rejects.

The steps here correspond to reading the input, so their number is the length of the input, we have a TM with time bound $t(n) = n$.