Solution Sheet n°3

Solution of exercise 1:

1.

   (a) This TM accepts the language $\{(01)^n 0 \mid n \in \omega\}$. It actually computes the function $(01)^n 0 \mapsto (10)^n 1$.

   (b) This TM accepts the language $\{01^n \mid n \in \omega\}$. It actually computes the function $01^n \mapsto 1^{n+1}$.

2. We can for example define δ by

$$\delta(q_0, 0) = (q_{acc}, 1, R);\ \delta(q_0, 1) = (q_0, 0, R);\ \delta(q_0, \sqcup) = (q_{acc}, 1, R).$$

Solution of exercise 2:

Let $M = (Q, \Sigma, \Gamma, \delta, q_0, q_{acc}, q_{rej})$ be a bi-infinite TM. We define a one sided infinite tape TM $\overline{M} = (\overline{Q}, \Sigma, \overline{\Gamma}, \overline{\delta}, q_{init}, q_{acc}, q_{rej})$ which recognises the same language. We duplicate the set of states and we add a state for each symbol in Σ and a new initial state q_{init}

$$\overline{Q} = Q \cup \{\overline{q} \mid q \in Q\} \cup \{q^a \mid a \in \Sigma\} \cup \{q_{init}\}.$$

The idea is to "fold" the tape of M by setting $\overline{\Gamma} = \Sigma \cup \{\binom{a}{b} \mid a, b \in \Gamma\} \cup \{\rhd\}$. We also added a new symbol $\rhd$ in order to make the left-hand end of the tape explicit. We define the transition function as follows. We first modify the input and come back to the left-hand end of the tape:

$$\begin{aligned}
\overline{\delta}(q_{init}, a) =& (q^a, \rhd, R) \quad \text{for all } a \in \Sigma \\
\overline{\delta}(q^a, b) =& (q^b, \tbinom{\sqcup}{a}, R) \quad \text{for all } a, b \in \Sigma \\
\overline{\delta}(q^a, \sqcup) =& (q_{init}, \tbinom{\sqcup}{a}, L) \quad \text{for all } a \in \Sigma \\
\overline{\delta}(q_{init}, \tbinom{\sqcup}{a}) =& (q_{init}, \tbinom{\sqcup}{a}, L) \quad \text{for all } a \in \Sigma \\
\overline{\delta}(q_{init}, \rhd) =& (q_0, \rhd, R) \quad \text{for all } a \in \Sigma
\end{aligned}$$

On the input $aba \sqcup \sqcup \cdots$ this first step would lead to $\rhd \binom{\sqcup}{a} \binom{\sqcup}{b} \binom{\sqcup}{a} \sqcup \sqcup \cdots$.

We then simulate M on the "folded" tape:

$$\begin{aligned}
\overline{\delta}(q, \tbinom{a}{b}) =& (r, \tbinom{a}{c}, \epsilon) \quad \text{if } \delta(q, b) = (r, c, \epsilon) \\
\overline{\delta}(q, \sqcup) =& (r, \tbinom{\sqcup}{c}, \epsilon) \quad \text{if } \delta(q, \sqcup) = (r, c, \epsilon) \\
\overline{\delta}(q, \rhd) =& (\overline{q}, \rhd, R) \\
\overline{\delta}(\overline{q}, \tbinom{a}{b}) =& (\overline{r}, \tbinom{c}{b}, \epsilon^*) \quad \text{if } \delta(q, a) = (r, c, \epsilon) \\
\overline{\delta}(\overline{q}, \sqcup) =& (\overline{r}, \tbinom{c}{\sqcup}, \epsilon^*) \quad \text{if } \delta(q, \sqcup) = (r, c, \epsilon) \\
\overline{\delta}(\overline{q}, \rhd) =& (q, \rhd, R),
\end{aligned}$$

where $\epsilon \in \{L, R\}$ and $R^* := L, L^* := R$.

The idea here is that every time M passes through the "middle" of the bi-infinite tape, $\overline{M}$ switches its focus from the bottom row to the top row of the $\binom{a}{b}$s or vice versa. The Turing machine $\overline{M}$ recognises the same language as M.

Solution of exercise 3:

1. The transition for $(A, 0)$, namely reading 0 in the starting state A, can be defined by one of the following eight possibilities:

$$A0R,\ A1R,\ A0L,\ A1L,\ H0R,\ H0L,\ H1R,\ H1L,$$

where for instance $A0R$ stands for write 0 and go Right in state A. The transitions $A0R$, $A1R$, $A0L$, $A1L$ eventually lead to an infinite computation (loop) on the empty word. For instance, in the case of $0RA$ the consecutive configurations are

$$A0 \to 0A0 \to 00A0 \to 000A0 \to \dots$$

Therefore, there essentially is only four one-state machines which halt on the empty word. Since we technically also have to define the transition for $(A, 1)$, reading 1 in state A, for which there are also 8 possibilities, there are $4 \cdot 8 = 32$ one-state machines which halt on the empty word.

2. It follows from the previous answer that the answer is one, as realised by the machines whose transitions for $(A, 0)$ are either $H1R$ or $H1L$.

3. The two-state machine whose transition function is given by the following table does the work.

	A	B
0	$B1R$	$A1L$
1	$B1L$	$H1R$

Its computation on the empty word is the following:

$$A0 \to 1B0 \to A11 \to B011 \to A0111 \to 1B111 \to 11H11.$$

Solution of exercise 4:

1. Let L_1 and L_2 be Turing decidable languages. There are by definition Turing machines M_1 and M_2 which respectively decide L_1 and L_2.

   (a) A decider for $L_1^{\complement}$, the complement of L_1, is a Turing machine which is identical to M_1, but whose accepting and rejecting states are switched.

   (b) A decider for $L_1 \cup L_2$ is the 2-tape Turing machine M described by the following instructions:

   > Beginning on input w on tape 1: copy w on tape 2; return both heads to the leftmost position [1]. Then simulate M_1 on tape 1 and M_2 on tape 2. If at any point M_1 (resp. M_2) enters a rejecting state, enter a dummy computation on the first (resp. second) tape and continue the computation of M_2 (resp. M_1).

[1] this requires $o(|\Sigma|)$ additional states and a new symbol.

- If at some point M_1 or M_2 enter an accepting state, then **accept**.
- If at some point both M_1 and M_2 have entered a rejecting state, then **reject**.

(c) A decider for $L_1 \cap L_2$ is the 2-tape Turing machine M described by the following instructions:

> Beginning on input w on tape 1: copy w on tape 2; return both heads to the leftmost position[1]. Then simulate M_1 on tape 1 and M_2 on tape 2. If at any point M_1 (resp. M_2) enters an accepting state, enter a dummy computation on the first (resp. second) tape and continue the computation of M_2 (resp. M_1).
>
> - If at some point M_1 or M_2 enter a rejecting state, then **reject**.
> - If at some point both M_1 and M_2 have entered an accepting state, then **accept**.

2. The ideas are analogous to the first point of the exercise. Notice that in this case the computation on one or both of the tapes could fail to halt on some input w.