COMP 2805 — Solutions Assignment 4

Question 1: (25 marks) Construct a Turing machine with one tape that accepts the language

$$\{0^{2n}1^n : n \geq 0\}.$$

Assume that, at the start of the computation, the tape head is on the leftmost symbol of the input string. Explain the meaning of the states that you use.

Solution: The Turing machine will do the following.
Stage 1: Delete the two leftmost symbols (if they are 0's).
Stage 2: Walk to the rightmost symbol and delete it (if it is a 1).
Stage 3: Walk to the leftmost symbol.

Repeat stages 1–3 until the string is empty. If nothing "strange" happens, accept; otherwise, reject.

We use the following states:

- q_0: start state.
- q_1: leftmost symbol was a 0; it has been deleted.
- q_2: second symbol from the left was a 0; it has been deleted; we walk to the rightmost symbol.
- q_3: we are at the rightmost symbol.
- q_4: rightmost symbol was a 1; it has been deleted; we walk back to the leftmost symbol.
- q_{accept}
- q_{reject}

Here are the instructions:

$$\begin{array}{lll} q_0 0 \rightarrow q_1 \Box R & q_1 0 \rightarrow q_2 \Box R & q_2 0 \rightarrow q_2 0 R \\ q_0 1 \rightarrow q_{reject} & q_1 1 \rightarrow q_{reject} & q_2 1 \rightarrow q_2 1 R \\ q_0 \Box \rightarrow q_{accept} & q_1 \Box \rightarrow q_{reject} & q_2 \Box \rightarrow q_3 \Box L \end{array}$$

$$\begin{array}{ll} q_3 0 \rightarrow q_{reject} & q_4 0 \rightarrow q_4 0 L \\ q_3 1 \rightarrow q_4 \Box L & q_4 1 \rightarrow q_4 1 L \\ q_3 \Box \rightarrow q_{reject} & q_4 \Box \rightarrow q_0 \Box R \end{array}$$

Question 2: (25 marks) Construct a Turing machine with one tape, that gets as input an integer $x \geq 1$, and returns as output the integer $x - 1$. Integers are represented in binary.

Start of the computation: The tape contains the binary representation of the input x. The tape head is on the leftmost bit of x, and the Turing machine is in the start state.

End of the computation: The tape contains the binary representation of the number $x-1$. The tape head is on the leftmost bit of $x-1$, and the Turing machine is in the final state.

The Turing machine in this question does not have an accept state or a reject state; instead, it has a final state. As soon as this final state is entered, the Turing machine terminates. At termination, the contents of the tape is the output of the Turing machine.

Solution: The Turing machine will do the following:
Stage 1: Walk to the rightmost bit of the input string.
Stage 2: Walk to the left and replace each 0 by a 1. When the first 1 is reached, replace it by a 0. At that moment, $x-1$ has been computed.
Stage 3: Walk to the leftmost bit.

We use the following states:

- q_0: start state; we are in stage 1.
- q_1: final state.
- q_2: we are in stage 2. Until now, we have encountered only 0's.
- q_3: we are in stage 3; $x-1$ has been computed; we walk to the leftmost bit.

Here are the instructions:

$$\begin{array}{lll} q_0 0 \rightarrow q_0 0 R & q_2 0 \rightarrow q_2 1 L & q_3 0 \rightarrow q_3 0 L \\ q_0 1 \rightarrow q_0 1 R & q_2 1 \rightarrow q_3 0 L & q_3 1 \rightarrow q_3 1 L \\ q_0 \Box \rightarrow q_2 \Box L & & q_3 \Box \rightarrow q_1 \Box R \end{array}$$

Question 3: **(25 marks)** Construct a Turing machine with three tapes that gets as input two non-negative integers x and y, and returns as output the number $x+y$. Integers are represented in binary.

Start of the computation: Tape 1 contains the binary representation of x, its head is on the **rightmost** bit of x. Tape 2 contains the binary representation of y, its head is on the **rightmost** bit of y. Tape 3 is empty (that is, it contains only $\Box$'s), its head is at an arbitrary position. At the start, the Turing machine is in the start state.

End of the computation: Tapes 1 and 2 are empty, and tape 3 contains the binary representation of the number $x+y$. The head of tape 3 is on the **rightmost** bit of $x+y$. The Turing machine is in the final state.

The Turing machine in this question does not have an accept state or a reject state; instead, it has a final state. As soon as this final state is entered, the Turing machine terminates. At termination, the contents of the tape is the output of the Turing machine.

Solution: The Turing machine will do the following.
Stage 1: Walk (simultaneously on all tapes) from right to left, and perform the addition. While doing this, write $x + y$ on tape 3, and delete all bits from tapes 1 and 2.
Stage 2: In this stage, the sum $x + y$ has been computed, and tapes 1 and 2 are already empty. In this stage, head 3 moves to the rightmost bit on its tape.

We use the following states:

- q_0: start state; we are in stage 1; there is no carry.
- q_1: we are in stage 1; there is a carry.
- q_2: we are in stage 2.
- q_3: final state.

Here are the instructions:

$$\begin{array}{l}
q_0 00\square \rightarrow q_0 \square\square 0LLL \\
q_0 01\square \rightarrow q_0 \square\square 1LLL \\
q_0 10\square \rightarrow q_0 \square\square 1LLL \\
q_0 11\square \rightarrow q_1 \square\square 0LLL \\
q_0 \square 0\square \rightarrow q_0 \square\square 0LLL \\
q_0 \square 1\square \rightarrow q_0 \square\square 1LLL \\
q_0 0\square\square \rightarrow q_0 \square\square 0LLL \\
q_0 1\square\square \rightarrow q_0 \square\square 1LLL \\
q_0 \square\square\square \rightarrow q_2 \square\square\square RRR \\
q_1 00\square \rightarrow q_0 \square\square 1LLL \\
q_1 01\square \rightarrow q_1 \square\square 0LLL \\
q_1 10\square \rightarrow q_1 \square\square 0LLL \\
q_1 11\square \rightarrow q_1 \square\square 1LLL \\
q_1 \square 0\square \rightarrow q_0 \square\square 1LLL \\
q_1 \square 1\square \rightarrow q_1 \square\square 0LLL \\
q_1 0\square\square \rightarrow q_0 \square\square 1LLL \\
q_1 1\square\square \rightarrow q_1 \square\square 0LLL \\
q_1 \square\square\square \rightarrow q_2 \square\square 1NNN \\
q_2 \square\square 0 \rightarrow q_2 \square\square 0RRR \\
q_2 \square\square 1 \rightarrow q_2 \square\square 1RRR \\
q_2 \square\square\square \rightarrow q_3 \square\square\square LLL
\end{array}$$

Question 4: (10+10+5marks) In class, we have seen that the language

$$A_{TM} = \{\langle M, w\rangle : \ M \text{ is a Turing machine that accepts } w\}$$

is not decidable. Consider the language

$$REG_{TM} = \{\langle M\rangle : \ M \text{ is a Turing machine whose language } L(M) \text{ is regular}\}.$$

The questions below will lead you through a proof of the claim that the language REG_{TM} is not decidable.

(4.1) Consider a fixed Turing machine M and a fixed binary string w.

We construct a new Turing machine T_{Mw} which takes as input an arbitrary binary string x. On such an input x, the Turing machine T_{Mw} does the following:

> **if** $x = 0^n 1^n$ for some $n \geq 0$
> **then** terminate in the accept state
> **else** run M on the input w;
>   **if** M terminates in the accept state
>   **then** terminate in the accept state
>   **else if** M terminates in the reject state
>   **then** terminate in the reject state
>   **endif**
>   **endif**
> **endif**

Answer the following two questions:

- Assume that M accepts the string w. What is the language $L(T_{Mw})$ of the new Turing machine T_{Mw}?

  Solution: We have to find out which strings x are accepted by T_{Mw}. Consider an arbitrary binary string x. We go with x through the pseudocode for T_{Mw} and see what happens:

  1. If $x = 0^n 1^n$ for some $n \geq 0$, then T_{Mw} accepts x.
  2. Otherwise, we run M on the string w. Since M accepts w, T_{Mw} accepts x.

  We conclude that T_{Mw} accepts all binary strings x. It follows that

  $$L(T_{Mw}) = \{0, 1\}^*.$$

- Assume that M does not accept the string w. What is the language $L(T_{Mw})$ of the new Turing machine T_{Mw}?

  Solution: We have to find out which strings x are accepted by T_{Mw}. Consider an arbitrary binary string x. We go with x through the pseudocode for T_{Mw} and see what happens:

  1. If $x = 0^n 1^n$ for some $n \geq 0$, then T_{Mw} accepts x.

2. Otherwise, we run M on the string w. We are given that M does not accept w. This means that either M terminates in its reject state or M does not terminate. It follows that either T_{Mw} terminates in its reject state or T_{Mw} does not terminate. Thus, T_{Mw} does not accept the string x.

We conclude that T_{Mw} accepts exactly all binary strings x of the form $x = 0^n 1^n$ for some $n \geq 0$. It follows that

$$L(T_{Mw}) = \{0^n 1^n : n \geq 0\}.$$

(4.2) The goal is to prove that the language REG_{TM} is not decidable. We will prove this by contradiction. Thus, we assume that R is a Turing machine that decides REG_{TM}. Recall what this means:

- If M is a Turing machine whose language is regular, then R, when given $\langle M \rangle$ as input, will terminate in the accept state.
- If M is a Turing machine whose language is not regular, then R, when given $\langle M \rangle$ as input, will terminate in the reject state.

We construct a new Turing machine R' which takes as input an arbitrary Turing machine M and an arbitrary binary string w. On such an input $\langle M, w \rangle$, the Turing machine R' does the following:

construct the Turing machine T_{Mw} described above;
run R on the input $\langle T_{Mw} \rangle$;
if R terminates in the accept state
then terminate in the accept state
else if R terminates in the reject state
    then terminate in the reject state
    endif
endif

Prove that M accepts w if and only if R' (when given $\langle M, w \rangle$ as input), terminates in the accept state.

Solution: We first assume that M accepts w.

- As we have seen above, $L(T_{Mw}) = \{0, 1\}^*$, which is a regular language.
- Thus, when running R on the input $\langle T_{Mw} \rangle$, R terminates in its accept state.
- Then it follows from the pseudocode for R' that, on input $\langle M, w \rangle$, R' terminates in its accept state.

This proves one direction.

For the other direction, we assume that, on input $\langle M, w \rangle$, R' terminates in its accept state.

- It follows from the pseudocode for R', that R, on input $\langle T_{Mw} \rangle$, terminates in the accept state.
- This means that the language $L(T_{Mw})$ of T_{Mw} is regular.
- As we have seen above,
  - if M accepts w, then $L(T_{Mw}) = \{0, 1\}^*$, which is regular.
  - if M does not accept w, then $L(T_{Mw}) = \{0^n 1^n : n \geq 0\}$, which is not regular.
- Thus, since $L(T_{Mw})$ is regular, it follows that M accepts w.

(4.3) Now finish the proof by arguing that the language REG_{TM} is not decidable.

Solution: Above, we have assumed that REG_{TM} is decidable. Based on this assumption, we have constructed a Turing machine R' that has the following property:

- R' accepts the input string $\langle M, w \rangle$ if and only if M accepts the input string w.
- This means: R' accepts $\langle M, w \rangle$ if and only if $\langle M, w \rangle \in A_{TM}$.
- But, by definition, this means that the language A_{TM} is decidable.
- However, A_{TM} is not decidable. Therefore, REG_{TM} is not decidable.