COMP 2805 — Assignment 4

Due: Thursday, April 2.

Where to submit: Due to renovations in Herzberg 4135, you have to submit the assignment in the main office of the School of Computer Science (Herzberg 5302). The office will be open until 4:30pm.

Assignment Policy: Late assignments will not be accepted. Students are encouraged to collaborate on assignments, but at the level of discussion only. When writing the solutions, they should do so in their own words. Past experience has shown conclusively that those who do not put adequate effort into the assignments do not learn the material and have a probability near 1 of doing poorly on the exams.

Question 1: 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.

Question 2: 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.

Question 3: 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.

Question 4: 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}?
- Assume that M does not accept the string w. What is the language $L(T_{Mw})$ of the new Turing machine T_{Mw}?

(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.

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