COMP 3803 — Assignment 4

Due: Wednesday December 7, 23:59.

Assignment Policy:

- Your assignment must be submitted as one single PDF file through Brightspace.

> Use the following format to name your file:
>
> `LastName_StudentId_a4.pdf`

- **Late assignments will not be accepted. I will not reply to emails of the type "my internet connection broke down at 23:57" or "my scanner stopped working at 23:58", or "my dog ate my laptop charger".**
- You are encouraged to collaborate on assignments, but at the level of discussion only. When writing your solutions, you must do so in your 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.
- When writing your solutions, you must follow the guidelines below.
  - You must justify your answers.
  - The answers should be concise, clear and neat.
  - When presenting proofs, every step should be justified.

Question 1: Write your name and student number.

Question 2: Construct a Turing machine with one tape that shifts a non-empty string over the alphabet $\{a, b\}^*$ one cell to the right.

Start of the computation: The tape contains the string

$$*w_1w_2 \dots w_n$$

where $n \geq 1$. The tape head is at the special symbol $*$ and the Turing machine is in the start state.

End of the computation: The tape contains the string

$$*\Box w_1w_2 \dots w_n$$

The tape head is at the special symbol $*$ 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.

Start by explaining your algorithm in plain English, then mention the states that you are going to use, then explain the meaning of these states, and finally give the list of instructions.

Question 3: Construct a Turing machine with two tapes that "doubles" a non-empty string over the alphabet $\{a\}$.

Start of the computation: Tape 1 contains a string of the form a^n, for some $n \geq 1$, and its head is on the **leftmost** a. Tape 2 is empty (that is, it contains only $\Box$'s) and its head is at an arbitrary position. At the start, the Turing machine is in the start state.

End of the computation: Tape 1 contains the same string a^n and its head is at the **leftmost** a. Tape 2 contains the string a^{2n} and its head is at the **leftmost** a. At the end, 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.

Start by explaining your algorithm in plain English, then mention the states that you are going to use, then explain the meaning of these states, and finally give the list of instructions.

Question 4: Consider the language

$$tztt = \{\langle M \rangle : \quad M \text{ is a Turing machine, for every input string } w \in \{0,1\}^*, \text{ the computation of } M \text{ on input } w \text{ terminates within 2022 steps}\}.$$

Professor Justin Bieber claims that *tztt* is undecidable.
Is Professor Bieber's claim correct? As always, justify your answer.

Question 5: Consider the language

$$Weird = \{\langle M \rangle : M \text{ is a Turing machine that does not accept the string } \langle M \rangle\}.$$

Use a proof by contradiction to show that *Weird* is undecidable.

Bonus Question: Consider the following algorithm:

Algorithm *Collatz*(n):
comment: $n \geq 1$ is an integer
if $n = 1$
then terminate
else if n is even
  then *Collatz*($n/2$)
  else *Collatz*($3n + 1$)
  endif
endif

A famous conjecture in mathematics (that nobody has been able to prove) is the following:

Collatz Conjecture: For every integer $n \geq 1$, algorithm *Collatz*(n) terminates.

In class, we have seen the language

$$Halt = \{\langle P, w \rangle : P \text{ is a Java program that terminates on the binary input string } w\}.$$

Prove the following claim: If the language *Halt* is decidable, then we can prove whether or not the Collatz Conjecture is true.