COMPX361-23B Logic and Computation

LABORATORIES

- Included here are the four labs you are asked to complete. These labs run starting Week 5 only. Each lab counts 2.5% of your final mark.
- Each lab can either be submitted via *Moodle*, or can be verified in *your* lab class, and *will only be verified up to a certain week, depending on the lab* (see below for details). If getting your lab verified in this way, bring along this leaflet each week for marks to be recorded and the relevant page filled in.
- You *must attend the lab class you signed up for, not the other one.*
- You will need to use the software *Tuatara Turing Machine*. This software was designed by previous class members and lab demonstrators. You should prepare your Turing machine prior to the lab in which it is to be verified. The software is freely available online (via *GitHub*).
- For each lab, there will be a range of weeks in which it can be verified:

Lab no.	To be verified by end of:
1	Week 7 (ends 8 September)
2	Week 8 (ends 15 September)
3	Week 10 (ends 29 September)
4	Week 12 (ends 13 October)

  In labs, priority will be given to students needing their lab work to be verified that week.

- **For this part of the paper, we will always assume 0 is a natural number, so $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$.**

Laboratory 1

Bring this to the lab class to receive your marks.

1. Design a Turing machine which computes the successor function *scsr* defined by

   $$scsr(n) = n + 1$$

   where the input is a unary representation of the nonnegative integer n.

   (NOTE: the unary representation of $n \in \mathbb{N}$ is the string 1^{n+1}, i.e., a string of $(n + 1)$ "1"s. So the number zero is represented as 1, and so on!)

   As in all cases, the machine should eventually halt with the read/write head at the left end of the tape (parked).

2. Design a Turing machine with input alphabet $\{0, 1\}$ that shifts a non-empty string one cell to the right, and writes a "0" in the leftmost cell of the tape.

3. Finally, design a Turing machine to compute the predecessor function defined by

   $$pred(n) = n - 1$$

   where the input is a unary representation of the integer $n > 0$.

NAME :

MARKS (each out of 2):

1.

2.

3.

Laboratory 2

Bring this to the lab class to receive your marks.

1. Design a Turing machine that duplicates a non-empty string over $\{0, 1\}$; i.e. if the input string is w, then the output string is ww, with its first symbol in the leftmost cell of the tape. Your tape alphabet can contain symbols other than 0, 1 and blank.

2. Design a Turing machine which, when the input is the non-empty binary string w, gives as output the string ww^r in its leftmost cells. (*Note*: w^r is the string w in reverse order; so if $w = x_1x_2 \cdots x_n$, with all $x_i \in \{0, 1\}$, then $w^r = x_nx_{n-1} \cdots x_1$.)

NAME :

MARKS (each out of 3):

1.

2.

Laboratory 3

Bring this to the lab class to receive your marks.

1. Design a *binary* Turing machine that compares the length of two strings of 1s, and outputs 11 if the strings are of equal length or outputs the string 1 if the strings are of unequal length.

2. Design a *binary* Turing machine to compute the function **cutoff** : $\mathbb{N}^2 \to \mathbb{N}$ defined by

$$\mathbf{cutoff}(m, n) = \begin{cases} m - n, & \text{if } m \geq n \\ 0 & \text{otherwise.} \end{cases}$$

   The input is to be in unary form, with m, n separated by a 0.

   Recall that for current purposes, $0 \in \mathbb{N}$.

NAME :

MARKS (each out of 3):

1.

2.

Laboratory 4

Bring this to the lab class to receive your marks.

1. Design a *binary* Turing machine to calculate the partial function **THIRD**: $\mathbb{N} \to \mathbb{N}$ defined by

$$\mathbf{THIRD}(n) = \begin{cases} n/3, & \text{if } n \text{ is divisible by 3, and} \\ \text{undefined} & \text{otherwise.} \end{cases}$$

   Recall that for current purposes, $0 \in \mathbb{N}$. Ensure that your TM at least stops (does not carry on generating new configurations forever) regardless of the input string. If the function is undefined, your Turing machine should terminate somehow but without completing a computation (so there is no output). The input and output (if it exists) should be in unary form.

2. For each of the following input strings (what numbers do they represent?), note the state and the contents of the tape when the machine stops. (If submitting via *Moodle*, include a copy of the completed table below in your submission.)

Input	State	Output String
1		
111		
1111		
1111111111		

NAME :

MARKS (Question 1 out of 4, Question 2 out of 2):

1.

2.