Homework 3

(Deadline: 10th of June 2022, Friday, 23:59)

1 Regulations

1. The homework is due by the 10th of June 2022, Friday, 23:59. **Late submission is not allowed**.
2. This homework consists of two parts: **Graded Questions** and **Self-Study Questions**. As the name implies, Self-Study Questions will not be graded (but you are recommended to solve them). Therefore, please **only** submit answers to graded questions
3. Submissions must be made via **ODTUClass**. Do not send your homework via e-mail, or do not bring any hardcopy.
4. For this homework you are going to use an **online Turing machine visualization website** named https://turingmachine.io for solving the question 1 and question 2.
5. Submissions should be delivered as `.zip` file. In the .zip file;
   - There should be a pdf file for your report named as `report.pdf`.
   - Also, `.yaml document` for first and second question required. The document can be downloaded from the website by clicking the share button.
6. An `example.yaml` document is given for you to study. You can **import** any .yaml document to the website. After you import the document you will understand that shown machine is the same machine from Quiz 3.
7. Name .zip files you will submit as `<yourstudentid_hw3>` (e.g. 2132807_hw1.zip). In case you **violate** the naming convention, you will receive a penalty of 5 points (over 100).
8. Send an e-mail to staha@metu.edu.tr if you need to get in contact.
9. This is an **individual** homework, which means you have to answer the questions on your own. Any contrary case will be considered as cheating and university regulations about cheating will be applied.

2 Graded Questions

2.1 Tool Information

In the question 1 and question 2 you should use an online tool https://turingmachine.io for creating the Turing machines. In the tool, at each step, a Turing machine reads its current state and tape symbol, and looks them up in its transition table for an instruction. Each instruction does 3 things:

- **write** a symbol to the current tape cell
- **move** to the left or right by one cell
- **set** the new state

In this machine you don't need to use **write** and **set** instructions in each transition, but machine needs to **move** in every transition.

You don't need to create reject state, if the machine crashes with unexpected input that means it goes to the reject state.

2.2 Question 1

Design a Turing machine which recognizes the language $L = \{0^N 1^N | N \geq 1\}$

- $\Sigma = \{0, 1, \sqcup\}$, means that you cannot write any other symbol than these symbols.
- Document name: `q1.yaml`.
- Report:
  - The screenshot of the created machine.
  - A clear description of every state used in the machine.
  - Give initial and end state screenshots with a few input samples.
    - 000111, 0000111, 000011111, 0001110, 001101, 100011

2.3 Question 2

Design a Turing machine that computes the function $f(w) = ww^R, \Sigma(w) = \{0, 1\}$

- Example: `1011 -> 10111101`.
- Document name: `q2.yaml`.
- Report:
  - The screenshot of the created machine.
  - A clear description of every state used in the machine.
  - Give initial and end state screenshots with a few input samples.
    - 1011, 1110, 0101, 1010, 1010001, 00111

2.4 Question 3

Formally define a Turing machine with a 2-dimensional tape, its configurations, and its computation. Define what it means for such a machine to decide a language L. Show that t steps of this machine, starting on an input of length n, can be simulated by a standard Turing machine in time that is polynomial in t and n.

- Your answer should be in `report.pdf`.

3 Self-Study Questions

3.1 Exercise 1

Give a three-tape Turing machine which, when started with two binary integers separated by a ';' on its first tape, computes their product.

3.2 Exercise 2

Consider a Turing machine without a "go left" operation. That is, at each point, this machine can only move its head to right, or replace the current symbol under its tape. Show that this Turing machine variant is not equivalent to the standard Turing machine. What class of languages do these machines recognize?

3.3 Exercise 3

(Problem 4.3.4 of Lewis Papadimitriou) The stack of a PDA can be considered as a tape that can be written and erased only at the right end; in this sense a TM is a generalization of the deterministic PDA. In this problem we consider a generalization in another direction, namely the deterministic PDA with two stacks.

1. Define the operation of such a machine informally. Define what it means for such a machine to decide a language.
2. Show that the class of languages decided by such machines is precisely the class of recursive languages.