Theory of Computation

Spring 2025

Homework 1

HW due: March 24, 2025, 13:30

Write the programs that solve the problems on the next page for one of the following Turing Machine (TM) simulators:

1. `turingmachine.io` (single-tape, diagrams)
2. `turingmachinesimulator.com` (multi-tape)
3. `francoisschwarzentruber.github.io/turingmachinesimulator` (NTM)

We used simulator 1 in the 1st lab and simulators 2 and 3 in the 2nd lab. Use simulator 1 or 2 for problems 1 and 2 (single-tape), simulator 2 for problems 3 and 4 (multi-tape), and simulator 3 for problems 5 and 6 (non-deterministic, NTM).

In order to receive full points, you are required to:

- Write a **report** of your solution for the problems including the diagram of the problem (the format is up to you). The expected length per problem is a half page to a full page. The report must describe the basic ideas of your approach. Do not include the source codes or snapshots of the execution of the programs in the report. The report must be in PDF format.
- Provide the **source code** of all of the programs in the form of raw text files in TXT format (such that it is easy for us to copy them into a simulator). Alternatively, you can provide a share link to the program in the case of simulators that support sharing (i.e., simulators 1 and 2).
- Explicitly define when an input is **accepted** or **rejected** by your programs.
- Provide at least 3 **inputs** that your program **accepts** and **rejects**, respectively, and that are different from those that appear in the assignment of the problem. The output on the tape is not important.
- The solution should not be overly complicated and inefficient. All branches of the program executions must always terminate (e.g., it must avoid looping).

Problems must be solved individually (recall the rules from the 1st lecture), plagiarism is not allowed. We will check the source codes for similarities with others.

Score. Each particular problem is scored up to a given number of points, denoted in brackets. The maximum total number of points is 100.

Problems

1. (5 points)
   Provide a TM with the binary alphabet {0,1} that accepts the input string iff the amount of 0's is equal to the amount of 1's.

   Examples of inputs the TM must accept:

   ␣ (empty string)

   01

   001110

   111011010000

   Examples of inputs the TM must reject:

   0

   01110

2. (15 points)
   Provide a single-tape TM with the input alphabet {a, . . . , f, #, |} that decides if a given non-empty string of letters is present in an array of non-empty letter strings separated by the symbol #. The string to search for is located at the beginning of the tape, separated from the array by the symbol |. The TM must accept if the given string is present in the array and must reject otherwise. The TM also rejects inputs in an invalid format. You can assume that the input does not contain multiple characters # in a row.

   Examples of inputs the TM must accept:

   ace|deb#a#ace#cad

   dace|cafe#deaf#dace#cafe#deaf#dace

   f|f

   Examples of inputs the TM must reject:

   ace|deb#a#ade#cad

   af|fad#caf#afe#fae

   abc|

   |abc

   abc|abc#

3. (15 points)
Provide a multi-tape TM that accepts the language $\{a\texttt{-}b\texttt{=}c \mid a, b, c \in \{\texttt{0}, \texttt{1}\}^+ \wedge c = a - b\}$, that is, correct equations of a *subtraction* where all numbers are non-negative. The notation $\{\texttt{0}, \texttt{1}\}^+$ is same as $\{\texttt{0}, \texttt{1}\}^*$ except that empty strings are excluded. Note that preceding irrelevant zeros at the beginning of the binary strings are allowed (e.g. `001`).

Examples of inputs the TM must accept:

```
00-0=000
1000-1=0111
0100-11=1
10-010=0
```

Examples of inputs the TM must reject:

```
-1
0-=0
0-0=1
01-10=-1
```

4. (25 points)
Provide a multi-tape TM that accepts the language $\{a\texttt{/}b\texttt{=}c \mid a, b, c \in \{\texttt{0}, \texttt{1}\}^+ \wedge c = \lfloor a/b \rfloor\}$, that is, correct equations of an integer *division*, similar to the previous problem.

Example of inputs the TM must accept:

```
1/10=0
1000/101=1
01000/10=100
```

Example of inputs the TM must reject:

```
0/=
/101=0
1000/10=11
```

5. (15 points)
Provide a non-deterministic TM that accepts the input if it does *not* contain any identical binary strings. The particular substrings of the input are non-empty and are separated by `|`. (Hence, the input is also rejected if it starts or ends with `|`.) You can assume that the input does not contain multiple `|` characters in a row.

Examples of inputs the TM must accept:

- $_$ (empty string)
- `101`
- `11|1111`
- `11|10101|1101`
- `1010|1111|1|101|1011`

Examples of inputs the TM must reject:

- `|`
- `1000|1000`
- `11|10101|11|1101`

6. (25 points)
Provide an NTM that accepts if there is *no* path between vertices A and B in a given *undirected* graph with vertices $V = \{A, B, 1, 2, 3, 4\}$ and edges specified in the input string. Each edge is a pair of vertices with no separator. Particular edges are separated by symbol `;`. Self-edges and repetitive edges in the graph are allowed. The TM also rejects all inputs in an invalid format.

Examples of inputs the TM must accept:

- $_$ (empty string)
- `A1`
- `12`
- `AA;A1;12`
- `A1;12;2A;B3;34;4B`

Examples of inputs the TM must reject:

- `AB`
- `A1;1B`

```
2A;2B
A1;12;23;34;4B
A1;12;34;43;B2
A
;
AB;
A1;;1B
A1;1B;
```