Turing’s Machine Realization Using Iterator.

Ⅰ Part.

Realization of simplified Turing’s Machine.

Formulated and created by Kosolapov Nikolai. 11.06.2021.



Table of contents:

* What is simplified version of Turing’s machine?
* Structure and description.
* Using realization for solving problems.
* Sources

What is simplified version of Turing’s machine ?

This implementation based on iterator. Why I choose this conception to implement Turing’s Machine? Let’s create table of similarities and differences:

|  |  |
| --- | --- |
| Similarities | Differences |
| Turing’s machine reading head moving around the tape like iterator. | Turing’s Machine has a wider functionality. |
| Used for working with data during program execution. | The goals and applications vary. |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a0 | L | } | a0 | a0 | a0 | a0 | a0 | a0 | a0 | a0 |

Iterator

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a0 | L | } | a0 | a0 | a0 | a0 | a0 | a0 | a0 | a0 |

Turing’s Machine

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| a | q1 | q2 | q3 | … |
| L | q1}R | … | … | … |
| } | q2a0L | q0a0N | … | … |
| a0 | q1a0R | … | … | … |

This scheme shows and explains fact, that Turing’s Machine -modernized iterator. So, we need to increase functionality. In 1st part I tell you about simplified implementation. There is no conditions system in that implementation.

Structure and description.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a0 | L | } | a0 | a0 | a0 | a0 | a0 | a0 | a0 | a0 |

Turing’s Machine

In simplified model all processes set by consecutive commands without using conditions.

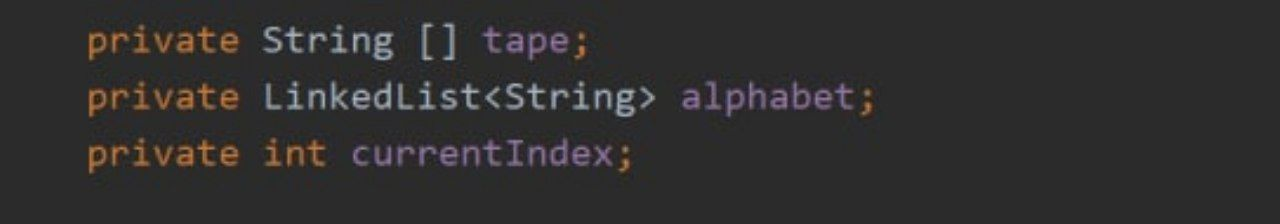
1) a0 > a0R

2) L > }R

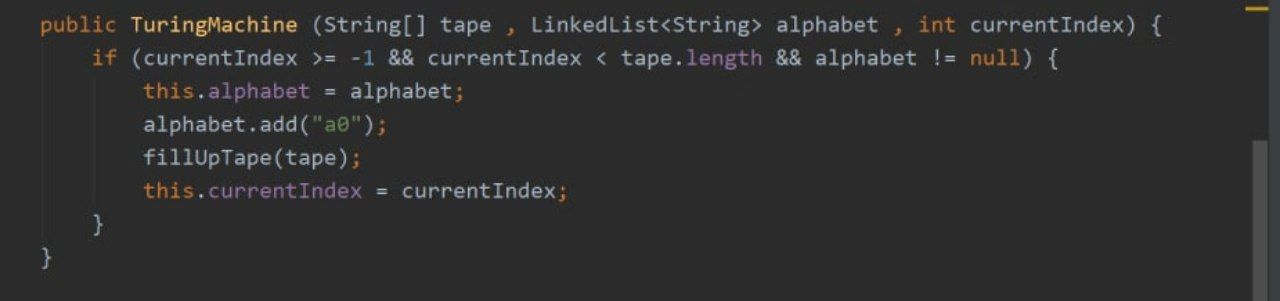
3) } > a0L

Let’s take a closer look at the structure.

1. Tape of machine (Use array , not infinite tape).
2. Alphabet of values (Use list).
3. Current position of reading head. (an int type variable).



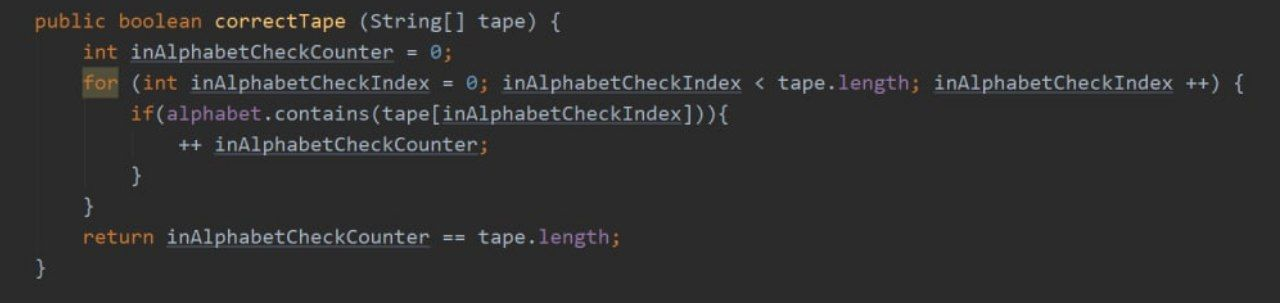
Constructor of our class:



Adding a0 as an empty value in alphabet.

Removing null values and replacing them with a0.





Tape is correct if all values in tape recorded in symbols list (alphabet).

Consider the control method of the reading head.

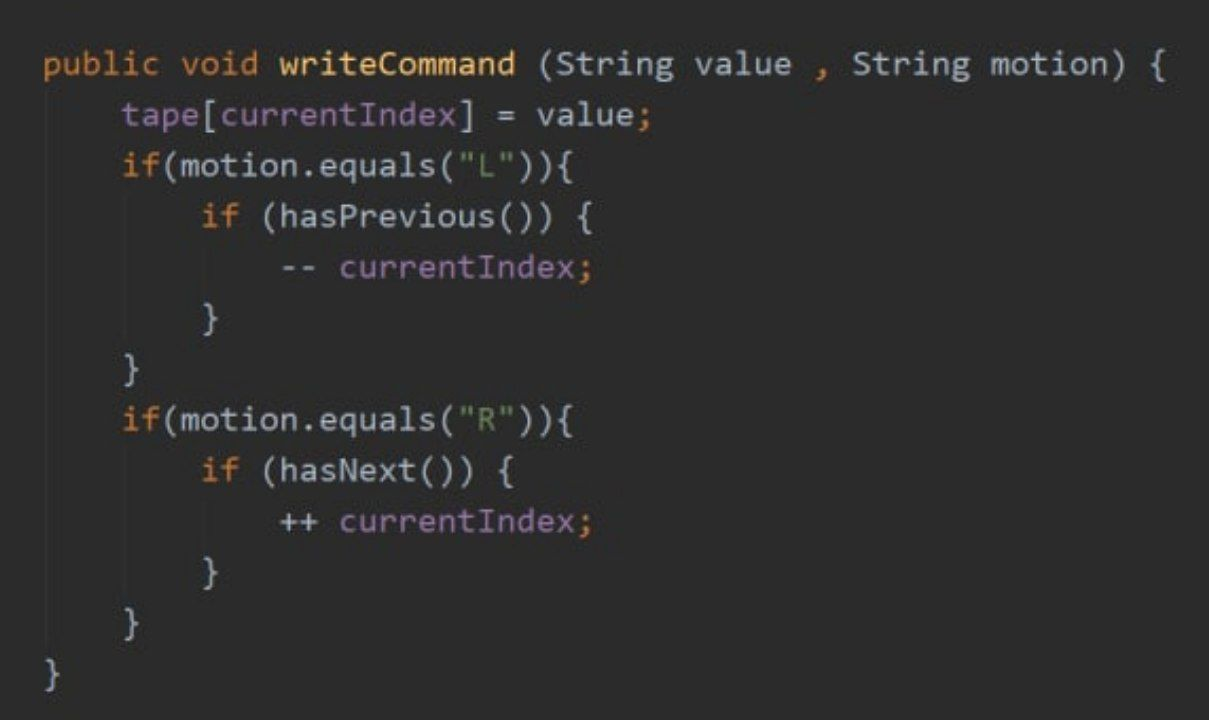
Example: Simplified:

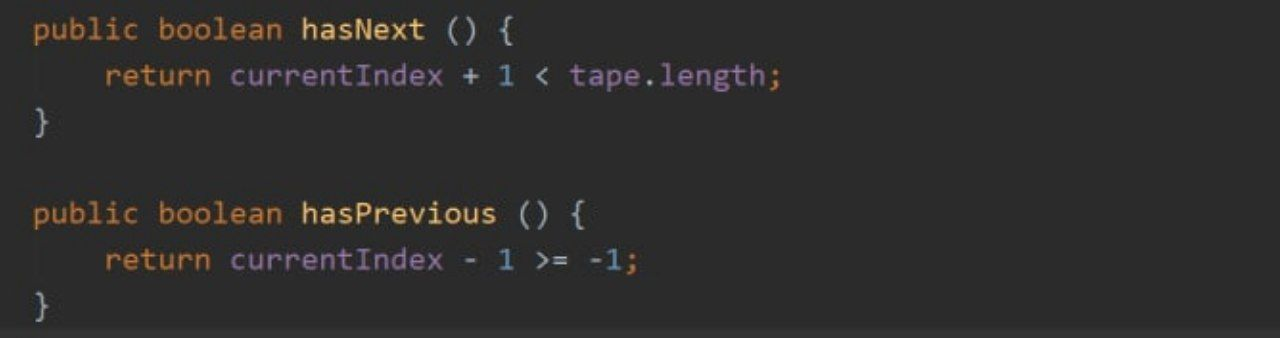
Input parameters

Current parameters

q1V > q1a0N V > a0N

q1X > q2a0R X > a0R





hasNext() and hasPrevious() – methods from iterator’s implementation.

I used them to check whether the current cell is the extreme one (whether the next/previous one exists).

Using realization for solving problems.

Problem:

Reverse “XYZ”. at the end of the program execution, the reading head should point to Z. (Position 1).

Input: “XYZ”

Output: “ZYX”

Alphabet: A: {a0 , X , Y , Z}

Current index: 0

Tape t:

0 1 2 3 4 5 6 7 8 9

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a0 | X | Y | Z | a0 | a0 | a0 | a0 | a0 | a0 |

In constructor:

(t , A , 0)

1. a0 > a0R
2. X > ZR
3. Y > YR
4. Z > XL
5. Y > YL
6. X > XN

Sources.

1. My Iterator’s implementation: <https://github.com/KosolapovNikolai05/ResearchesArchive/tree/master/Improved%20Array's%20Iterator%20Implementation>