Turing Machine Language

Pete Gautam

University of Glasgow

March 22, 2023

Introduction

- Turing Machines (TM) are a model of computation.
- They are typically taught theoretically, i.e. pen-and-paper.
- Students find it hard to learn the content.
- Perhaps it is easier to learn as a programming language (PL)?
- We present a PL for TMs which abstracts TM operations but keeps tape operations concrete.
- We investigate whether students find this method easier than the traditional approach.

Turing Machines

- Can be thought of as boolean functions on string
- Can be represented as a directed graph (called FSM)
- Has states and transitions

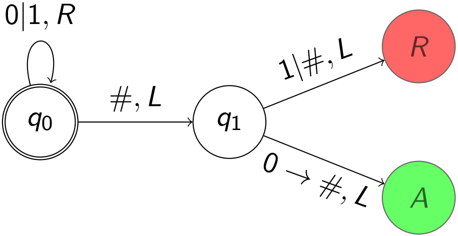

Figure: A FSM representation of TMs.

Project Overview

1. Creating the Turing Machine Language (TML).
2. Constructing a parser for TML.
3. Constructing the product (website) to showcase the parser.

Language

```
alphabet = {0, 1}
module isDiv2 {
    // move to the end
    while 0, 1 {
        move right
    } if blank {
        move left
        // check last letter is 0
        if 0 {
            changeto blank
            accept
        } if 1, blank {
            reject
        }
    }
}
```

Parser

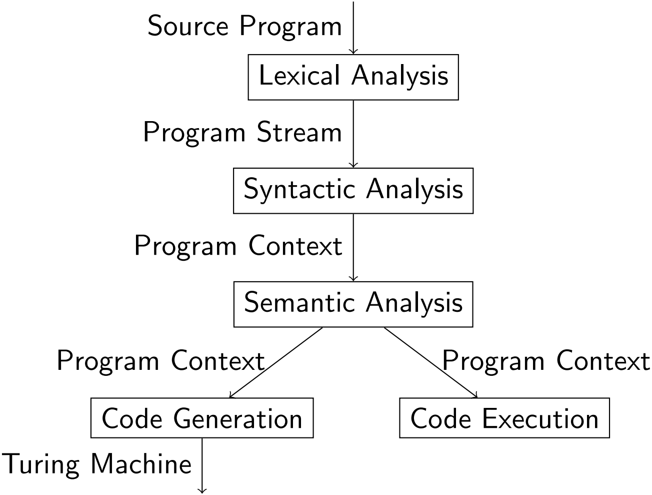

Figure: The parsing process.

Product

1. Homepage
   1.1 Editor
   1.2 Convert to FSM and definition
   1.3 Execute on tape
2. Documentation Pages
3. Error Pages

Evaluation

- Unit testing throughout
  - Parser
  - Website- less successful due to mocking
- User evaluation- worksheet on TM and TML with second year CS students
  - TML was easy to use
  - Students would consider using TML when constructing TMs
  - Students believe TML is a good language to learn about TMs while abstracting TM operations
  - Website is well-designed

Future Work

- Improve the language with more features:
  - `move end`;
  - parametrisation.
- Make the website more accessible:
  - Add a play button;
  - Allow the FSM to be zoomed in.