Project: Turing's Assembler
Theory of Computation (CSCI 3500)

An assembly language is a low-level programming language for programming computational devices. A low-level programming language is one that has a tight correspondence with the hardware of the machine. In this project we develop an assembly language for Turing Machines. The following is an example TM assembler program:

```
-- Initialization:
{states: Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7,A,R}
{start: Q0}
{accept: A}
{reject: R}
{alpha: 0,1,#}
{tape-alpha: 0,1,#,x}


-- Main Algorithm:
-- 0:
rwRt Q0 0 x Q1; -- Read, Write, Right, Transition.
rRl Q1 0; -- Read, Right, Loop.
rRl Q1 1;
rRt Q1 # Q3; -- Read, Right, transition.
rRl Q3 x;
rwLt Q3 0 x Q5; -- Read, Write, Left, Transition.


-- 1:
rwRt Q0 1 x Q2;
rRl Q2 0; -- Read, Right, Loop.
rRl Q2 1;
rRt Q2 # Q4; -- Read, Right, transition.
rRl Q4 x;
rwLt Q4 1 x Q5; -- Read, Write, Left, Transition.
-- Find #:
rLl Q5 x; -- Read, Left, Loop.
rLt Q5 # Q6; -- Read, Left, Transition.


-- Reset:
```

```
rLl Q6 0;
rLl Q6 1;
rRt Q6 x Q0;


-- Accept:
rRt Q0 # Q7;
rRl Q7 x;
rRt Q7 _ A;
```

Comments start with a -- and can appear anywhere. The first part of each TM source file is the initialization of the hardware which is a list of configurations. Then the rest of the file is a series of commands that defines the algorithm of the TM. The previous example is the definition of the TM that accepts the language $L = \{w\#w \mid w \in \{0,1\}^*\}$.

The main part of this project is to define the compiler for TM source files. It is to prompt the user for the path to the TM source file, and for a word that is to be tested for acceptance with respect to the TM defined in the TM source file. Then output a complete trace of the machine. For example:

```
Please enter a path to the source file: copy.TM
Please enter a word: 01#01
[Q0]01#01
x[Q1]1#01
 x1[Q1]#01
x1#[Q3]01
x1[Q5]#x1
x[Q6]1#x1
[Q6]x1#x1
x[Q0]1#x1
xx[Q2]#x1
xx#[Q4]x1
xx#x[Q4]1
xx#[Q5]xx
xx[Q5]#xx
x[Q6]x#xx
xx[Q0]#xx
xx#[Q7]xx
xx#x[Q7]x
xx#xx[Q7]
Accept: xx#xx[A]
```

1. TM file syntax

Every TM source file must contain two parts: the TM configuration and the algorithm for the transition function. The file must contain the configuration first. Comments can appear on any line and start with -- (including before the configuration).

The configuration of the TM consists of the following lines (they must appear in this order at the beginning of the file):

- States: {states: q0,...,qn} where each qi can be any string of alpha or numerical symbols.
- Start state: {start: q} where q is a state from the list of states given in the states configuration.
- Accept state: {accept: A} where A is a state from the list of states given in the states configuration.
- Reject state: {reject: R} where R is a state from the list of states given in the states configuration, and is distinct from the specified start state,
- Input alphabet: {alpha: a1,...,am} where each ai is any character at all.
- Tape alphabet: {tape-alpha: t1,...,tm,_} where each ti is any character at all, and the underscore (_) represents the blank symbol, and each input character in the input alphabet is an element of the tape alphabet.

The remainder of the file contains a list of commands describing the transitions of the Turing machine. These may occur in any order within the file. Each command ends with a ;. The following lists the possible commands and their intended semantics:

- rwRt q t t' q'; : in state q where the head is reading t off of the tape, write t', transition to state q' and move the head right.
- rwLt q t t' q'; : in state q where the head is reading t off of the tape, write t', transition to state q' and move the head left.
- rRl q t; : in state q where the head is reading t off of the tape, write t, transition to state q and move the head right.
- rLl q t; : in state q where the head is reading t off of the tape, write t, transition to state q and move the head left.
- rRt q t q'; : in state q where the head is reading t off of the tape, write t, transition to state q' and move the head right.
- rLt q t q'; : in state q where the head is reading t off of the tape, write t, transition to state q' and move the head left.