

Daedalus Documentation

1 Syntax

Currently, the syntax allows for 5 instructions. Line labels are supported and required for some instructions

1.1 Instructions

The current list of instructions are:

- Accept
- Reject
- If *symbol* Goto *label*
- Write *symbol*
- Goto *label*

1.1.1 Accept

When the program reaches this state, the Turing Machine will halt in the Accepted state.

1.1.2 Reject

When the program reaches this state, the Turing Machine will halt in the Rejected state.

1.1.3 If-Goto

When the program reaches this state, the program will jump to *label* if the symbol currently being read by the Turing Machine is *symbol*.

1.1.4 Write

When the program reaches this state, the Turing Machine will write *symbol* to the current position of the Turing Machine's head on the tape.

1.1.5 Goto

When the program reaches this state, the program will jump to *label* unconditionally.

1.2 Labels

Labels must be identified by being placed on their own line and ended by a colon. They cannot contain spaces, however they can include and begin with numbers, underscores, and any other special symbol. A label in a Daedalus program might look like

```
this_is_a_label:
write 1
move r
goto this_is_a_label
```

2 Daedalus Files

A Daedalus file uses the file extension *.ddls*, its intermediate language uses the extension *.wb1*, and the final, Turing Machine file uses *.tm*. A *Daedalus* file must begin with the symbols used by the Turing Machine, each separated by a space on the same line. All symbols must be a single character. The first symbol designated will be interpreted as the blank symbol.

3 Example Code

The following is an example of a program for binary addition.

```
0 1
start:
if 1 goto found_num
move r
goto start
found_num:
move r
if 0 goto concat
goto found_num
concat:
write 1
move r
if 1 goto seek_end
reject
seek_end:
move r
if 0 goto zero_end
goto seek_end
zero_end:
move l
write 0
accept
```