

INF3110 - Mandatory assignment 2

Mikael Olausson (mikaello)

November 11, 2012

Implementation

Symbols used in grid

I've used five different symbols inside the grid:

- default, this means empty space
- < means -x
- > means x
- ∨ means -y
- ∧ means y

When pen is down, all steps are recorded with the correct direction (depending on which way the robot was moving while recording).

Example of code and grid

```
down
start(2,3, x)
forward(3)
left(4)
backward(1)
right(2)
stop
```

```
· · · · · ^ · · · · ·
· · < < v · · · · ·
· · · · · ^ · · · · ·
· · · · · ^ · · · · ·
· > > > > · · · · ·
· · · · · · · · · · ·
· · · · · · · · · · ·
```

Coordinates

I start coordinates from 1 and up to the specified size. This means that if you specify a grid: `size(5,5)`, coordinate $x = 1, y = 5$ is valid, but not $x = 0, y = 4$.

Pen-up/Pen-down

When command down is used (*Pen-Down*), all the steps the robot takes is stored in the first parameter of «state», this variable is updated with these steps.

IfThenElse

IfThenElse works without a else-statement, just pass a empty list.

Attempts to move robot outside the grid

When someone tries to move the robot outside the grid, a error-message is written and an exception is raised (which makes the program quit because it isn't handled).

«Pretty print»

I have made a function for pretty printing, this is called with all the statements before execution of statements.

TestPrograms

I've added test-programs for the example given above, «Testing code 1» and «Testing code 4» (enabled as default)

How to run the program

You run the program by writing this in the terminal:

```
$ sml oblig2.sml
```

Get the code from GitHub

This program is out on GitHub, you can get it with this command

```
$ git clone  
https://github.com/mikaello/inf3110_oblig2.git
```