TDT 4205
Problem Set 1

The deadline for this problem set is September 15^{th}. The theory questions (1.1 through 2.1) may be turned in on paper before the lectures - label them clearly. (This is for the convenience of those who prefer drawing with pen and paper; it is of course possible to submit everything electronically.)

Submissions for the programming exercise (2.2), as well as any other electronic submission, will be accepted via 'it's learning', or by email to `janchris@idi.ntnu.no`.

1 Regular Expressions

1.1 20%

Draw a deterministic finite automaton (DFA) which recognizes complex numbers in (finite) decimal notation, e.g. $0, 127.5, 2.7182 - 3.1425i, 0 + 3i$, etc. (Assume that any pure imaginary number will still be written with the real part, i.e. with prefix $0+$ or $0-$)

1.2 20%

Write a regular expression for the same language, using character classes, positive closure and zero-or-one-instance notation (Dragon 3.3.5).

2 A Simple Language for Drawing Lines

These exercises will be concerned with a minimal toy language to create simple line drawings on a page. The context is that of an imaginary pencil point initially located near the top left corner of the page. It is controlled by the three commands `turn`, `draw`, and `move`:

- `Draw` draws a line of fixed length at a given angle (initially, straight towards the right), leaving the pencil point at its end.
- `Turn` alters the angle of the next step by 30° clockwise.
- `Move` shifts the point by one step without drawing the line in between.

These operations are already implemented as the functions `turn()`, `draw()` and `move()`; what remains to be implemented is an automaton which recognizes the corresponding keywords from a text stream, and calls the appropriate function.

The three keywords are to be case-insensitive (i.e. turn, TuRn and TURN are all correct). Characters not part of a command are to be ignored, such that text, whitespace and commands may be freely mixed in the input.

2.1 20%

Create a deterministic finite automaton which recognizes the three specified commands and consumes/ignores other input until a word has been recognized.

2.2 40%

The provided archive (`pencil.tgz`) contains code to translate the specified language into a postscript graphics file, except that it is missing a scanner. Implement your DFA from the previous task in `scanner.c`, to complete the program.