

CLP: Regular expressions (WS 2018)

This exercise sheet will be discussed in the exercise sessions on November 7.

Exercise 1 Regular expressions

Consider the following regular expressions, which are supposed to accept multiline comments starting with `{-` and ending with `-}`.

```
1 \{-[^-]*?-\  
2 "{-" {CommentContent} "-"+ "  
3 \{-(.|{LineTerminator})*-\  
4 \{- (. * {LineTerminator} * ) * -\  
5 "{-~"-}"  
6 (\{~(-\  
7 "{-" ({Letter}|{Number}|{WhiteSpace}|{SpecialC})+ "-}"  
8  
9 // macro definitions:  
10 CommentContent = ( [^-] | "-"+ [^]* ) *  
11 Letter = [a-zA-Z_]*  
12 Number = 0 | [1-9][0-9]*  
13 LineTerminator = \r|\n|\r\n  
14 WhiteSpace = {LineTerminator} | [ \t\f]  
15 SpecialC = ['+', '-', '*', '/', '<']
```

Hints:

- `~a` (upto)
matches everything up to (and including) the first occurrence of a text matched by `a`. The expression `~a` is equivalent to `!([~]* a [~]*) a`.
- `!a` (negation)
matches everything but the strings matched by `a`. To construct the automaton for `!a`, first construct the DFA for `a` and then invert the final states.

- Construct a DFA for each of the given regular expressions.
- Minimize the resulting automata if possible.
- Which regular expressions detect comments correctly?
- What is the worst-case running time of a scanner generated by JFlex?

Hint: Consider the input `{- {- {- {- {- {- {- {- {- {- ... <<EOF>>`