

Task 1.2

1) Design and write a unanimous BNF of PAM extended allowing complex conditional expressions that are built from comparisons (as in original PAM grammar) by binary logical connectives <=> (equivalence), => (implication), or (disjunction), and and (conjunction), and unary logical connective not (negation).

Extension of BNF meta-notation the alternative symbol | and the regular expressions is allowed.

Specify that the operations increase their binding strength in the following order: <=> (the weakest binding), =>, or, and (the strongest binding).

For instance, **2=3 and 3=4 or 5=5** shall be interpreted the same, as **(2=3 and 3=4) or 5=5** since and binds stronger than or.

The use of parentheses has to be allowed within the expressions to specify any order of evaluation.

Specify that <=>, and and or are left-associative, and => is right-associative.

2) Design and write a corresponding extended grammar of PAM in the ANTLR notation and enable the tool (e.g. IntelliJ IDEA) to recognize the grammar and draw parse trees for programs in extended PAM (left recursion is to be avoided for this purpose).

3) Let the tool draw the parse tree for the program:

x := 5; y :=7; if 2=3 and 3=4 or 5=5 then write x else write y fi

Note: The tool with your grammar has to be able to parse other programs, too.

Submission:

- 1) a .pdf file with the BNF grammar for extended PAM
- 2) a .g4 file that can be loaded into the ANTLR tool environment
- 3) a .pdf file with an ANTLR tool screenshot showing the grammar loaded into the tool and the parse tree of the test program

Grading: Max 30 points