

Task 1. For the second part of the project, your task is to reuse the grammar defined for the language in order to build the actual ALU language, using Rascal. Specifically you will have to:

1. Define syntax rules for ALU, with the syntax highlighting.
2. Build the abstract syntax for ALU (the AST).
3. Define the Parser for the language
4. Generate the code for the ALU definition from the syntax rules and the abstract syntax, so that you can run ALU programs. The input for the program generation should be a text file (`.alu`) with the contents of the program. You should show the result of the program in the console.

In the definition of the language you will have to fix any mistakes that may have appear during the grammar definition. To fix the problems from the first part of the project you may follow the instructions from the TA's or start from the project's solution provided by staff. Together with the Rascal project of your working solution, you should hand in a small document explaining the changes done to the grammar definition (if any) from the first project.

If you do not fix the problems from the first project (1) your language will not work properly, (2) you will be deducted points again for the same mistakes.