

Assignment 3

Design a parser on top of the baseline compiler infrastructure provided to you.

Right now the parser only supports the declaration of variables, and arithmetic operations. In this assignment you will update the baseline parser located in

`src/parser.yy` to add

- Variable assignment statement
- Ternary operator (`a = <expr> ? <expr> : <expr>`)

Define the appropriate grammar for the parser to take in the generated tokens by the lexer and generate the corresponding Abstract Syntax Tree nodes.

The code should parse statements like:

```
let a = 5;  
a = b + c;  
a = a ? b : c;
```

Tasks

1. You will have to add rules to the parser so that assignment and ternary operators are parsed.
 - a. When parsing for an assignment statement you should check if you are assigning to a variable that has not been declared. Print the appropriate error message.
 - b. You will have to declare new tokens for `?` and `:` and edit `src/lexer.lex` to add support to these new tokens, and also set the type of the assignment and ternary operator rules. Look at lines 27-39 in `src/parser.yy` to see how to do this.
 - c. The ternary operator works just like it does in C with respect to precedence. It should have lesser precedence than the arithmetic operators.
2. The new rules must create the appropriate AST nodes. You will have to define the new nodes in `include/ast.h` and `src/ast.cc`. Each new AST node that you will

define will inherit the base `Node` struct, and extends it with extra information according to the type of node. You will also have to implement two abstract methods, `to_string` and `llvm_codegen`.

a. Edit the `to_string` function following this specification:

i. `to_string` : The assignment `a = b` must return the string “`(assign a b)`”.

ii. `to_string` for the ternary operator : `a ? b : c` must return the string “`(?: a b c)`”

b. As we are not testing codegen in this assignment, we don't expect you to implement the `llvm_codegen` methods. We will leave them “blank” for now. In `src/llvmcodegen.cc`, implement the `llvm_codegen` methods for the assignment and the ternary operator nodes and have them return `nullptr`.

3. As the codegen for the new constructs you have added have not been implemented, we cannot run the entire pipeline. To test your parser run `./bin/base test.be -p`