

Parker Bixby

5/5/2023

Organized Programming Languages

Section 2

- **A description of the extension with examples**

In my implementation of MyPl I added a constant variable extension. I started working on my project pretty early on once we finished code generation in class so I figured out how to do the constant variables very early and added the tests. I realized once I was done that I wanted more of a challenge so I decided to add constant array objects. I was going to try and add constant structs, but didn't end up having enough time to figure out how I would assign the values of the const structs. Here are some examples of the code that works and how the grammar works:

CONST INT X = 9

CONST ARRAY INT XS = NEW INT {1,2,3,4,5}

- **A description of the parts of the MyPL pipeline that you modified and at a high level what the modifications were**

Even though I have this grammar for constant arrays, I am also able to set regular arrays with the same grammar that we have. After I set up the rules for the grammar, I set up a boolean value called `is_const` that I put into `DataTypes`, that I set in my `Stmt` function in my `ASTParser`. Aswell as the `is_const` I set up a vector of `SimpleRValues` that would contain the values in the array that I could then use during the code generator. Once I set it in my `ASTParser` I had to do the type checking in the semantic checker to

make sure that I am never using a constant variable, or instance of the array's on the left side, or trying to change the values of the constant variables, which I did in AssignStmt. The final part of the extension was making the arrays when I instantiate them with the VarDecl. In my code generator I created an array using ALLOCA, then I just set the index's of the array's to the values of the vector of SimpleRValues I made in my ASTParser.'

• What was successfully completed, what if anything wasn't completed (compared to your specification in Step 2), and what you would do next if you had additional time

I was able to successfully make array's and variables constant, however one of the problems that I had, was when I tried to pass in a instance of an array for example xs[0] into a function, then when I tried to increment it by one I would get a wrong index variant, and I couldn't figure out why even with debugging. I also was not able to make constant structs, however I think that I completed as much as I could have with the time I was given.

•How you tested your extensions including the new tests you developed

Here are some of the types of tests that I ran while going through my project. I did some basic const variable and array creation and access. I did some Parser Tests, Semantic checker tests, and code generation tests as well. A lot of my tests have to do with pass by value vs pass by reference and I am able to pass a constant variable into a function and I can make an adjustment to it, but once I leave the function the variable will be the same as it was before because when I pass it in, I am making a new variable and just adjusting it.

```

pbixby@ada:~/326/project-parkerbixby$ ./const_tests
[=====] Running 18 tests from 1 test suite.
[-----] Global test environment set-up.
[-----] 18 tests from BasicConstVarTest
[ RUN      ] BasicConstVarTest.SimplePrintConstVar
[ OK       ] BasicConstVarTest.SimplePrintConstVar (6 ms)
[ RUN      ] BasicConstVarTest.SimplePrintConstArray
[ OK       ] BasicConstVarTest.SimplePrintConstArray (0 ms)
[ RUN      ] BasicConstVarTest.ReassigningConstArrayValues
[ OK       ] BasicConstVarTest.ReassigningConstArrayValues (0 ms)
[ RUN      ] BasicConstVarTest.TypesDontMatchInVarDecl
[ OK       ] BasicConstVarTest.TypesDontMatchInVarDecl (0 ms)
[ RUN      ] BasicConstVarTest.TypesDontMatchInArrayExpr
[ OK       ] BasicConstVarTest.TypesDontMatchInArrayExpr (0 ms)
[ RUN      ] BasicConstVarTest.TypesBeforeExpr
[ OK       ] BasicConstVarTest.TypesBeforeExpr (0 ms)
[ RUN      ] BasicConstVarTest.NoNewRValue
[ OK       ] BasicConstVarTest.NoNewRValue (0 ms)
[ RUN      ] BasicConstVarTest.ParsingErrorWithoutArray
[ OK       ] BasicConstVarTest.ParsingErrorWithoutArray (0 ms)
[ RUN      ] BasicConstVarTest.AddOneToConstValue
[ OK       ] BasicConstVarTest.AddOneToConstValue (0 ms)
[ RUN      ] BasicConstVarTest.AddOneToConstValueArgument
[ OK       ] BasicConstVarTest.AddOneToConstValueArgument (0 ms)
[ RUN      ] BasicConstVarTest.ChangeConstValueInArray
[ OK       ] BasicConstVarTest.ChangeConstValueInArray (0 ms)
[ RUN      ] BasicConstVarTest.StringChangeInConstArray
[ OK       ] BasicConstVarTest.StringChangeInConstArray (0 ms)
[ RUN      ] BasicConstVarTest.ChangingAConstParamInFunc
[ OK       ] BasicConstVarTest.ChangingAConstParamInFunc (0 ms)
[ RUN      ] BasicConstVarTest.PassingConstArrayInFunc
[ OK       ] BasicConstVarTest.PassingConstArrayInFunc (0 ms)
[ RUN      ] BasicConstVarTest.ReassigningConstStringValues
[ OK       ] BasicConstVarTest.ReassigningConstStringValues (0 ms)
[ RUN      ] BasicConstVarTest.SimpleMultWithConst
[ OK       ] BasicConstVarTest.SimpleMultWithConst (0 ms)
[ RUN      ] BasicConstVarTest.ConcatWithConstStrings
[ OK       ] BasicConstVarTest.ConcatWithConstStrings (0 ms)
[ RUN      ] BasicConstVarTest.ForInnerShadowing
[ OK       ] BasicConstVarTest.ForInnerShadowing (0 ms)
[-----] 18 tests from BasicConstVarTest (11 ms total)

[-----] Global test environment tear-down
[=====] 18 tests from 1 test suite ran. (11 ms total)
[ PASSED ] 18 tests.

```

• Instructions for building and running your extension

In order to run my tests you run `./const_tests` and you can also use the examples file and compile

in Normal mode using

`./mypl examples/const.myp1`

This is the link for my video <https://youtu.be/FTOPFYFSDig>

