

Compiler Construction

Programming Assignment 2

Syntactic and Semantic Definitions for μGo

NCKU
National Cheng Kung University



What to do in this Assignment?

- Write an LALR(1) parser for μGO using Lex and Yacc.
- The parser supports print I/O, arithmetic operations, and some programming language basic concepts.
- The spec of μGO is available for your reference.
- You need to design grammar for your own parser by following the given spec.
- You also need to check semantic correctness by implementing *symbol table*.



Assignment Requirements

- Each test case is 10pt and the total score is 110pt.
- You can judge your code locally with the judger.

```
~ judge
local-judge: v2.7.1
====================+=============================================================
            Sample  | Accept
====================+=============================================================
    in01_arithmetic | ✔
====================+=============================================================
    in02_precedence | ✔
====================+=============================================================
         in03_scope | ✔
====================+=============================================================
    in04_assignment | ✔
====================+=============================================================
    in05_conversion | ✔
====================+=============================================================
           in06_if  | ✔
====================+=============================================================
          in07_for  | ✔
====================+=============================================================
    in08_type_error | ✔
====================+=============================================================
in09_variable_error | ✔
====================+=============================================================
      in10_function | ✔
====================+=============================================================
        in11_switch | ✔
====================+=============================================================
Correct/Total problems: 11/11
Obtained/Total scores:  110/110
```

```
// "Hard Coding" will get 0pt.
main() {
    result = read(answer_file);
    print(result);
}
```

https://github.com/aben20807/local-judge/tree/master/examples/student



Scoping

- What will this program print with proper scoping?

```
var x int32 = 10
{
  var x int32 = 5
  x++
  println(x)
}
println(x)
{
  x++
  println(x)
}
println(x)
```

- Output

```
6
10
11
11
```



Scoping (cont.)

```
var x int32 = 10
{
  var x int32 = 5
  x++
  println(x)
}
println(x)
{
  x++
  println(x)
}
println(x)
```

Scope Block A

Scope Block B

Scope Block A

Scope Block C

Scope Block A

Scope B's x

Access scope A's x



Scoping (cont.)

- A scope block is a set of statements enclosed within left and right braces ({ and }).
- A variable declared in a block is accessible in the block and all the inner blocks of that block, but not accessible outside the block.
- Different inner scope block in same scope block can't see each other.
- You can declare variable with same name in different scope.



Symbol table functions

- **create_symbol**: Create a symbol table when entering a new scope.
- **insert_symbol**: Insert entries for variables declarations.
- **lookup_symbol**: Look up entries in the symbol table.
- **dump_symbol**: Dump all contents in the symbol table of current scope and its entries when exiting a scope.

Note:

- Function's names, return type, and parameters can be properly defined by yourself.



Symbol table

```
1 package main
2
3 func main() {
4   var x int32 = 10
5   var y float32
6
7   {
8     var x float32 = 3.14
9   }
10 }
```

Insert main into symbol table (scope level: 0)

Insert x into symbol table (scope level: 1)

Insert y into symbol table (scope level: 1)

Insert x into symbol table (scope level: 2)



Symbol table (cont.)

Index: unique in each symbol table
Addr: unique in whole program

```
1  package main
2
3  func main() {
4    var x int32 = 10
5    var y float32
6
7    {
8      var x float32 = 3.14
9    }
10 }
```

Dump scope level 2's symbol table:

Index	Name	Type	Addr	Lineno	Func_sig
0	x	float32	2	8	-

Dump scope level 1's symbol table:

Index	Name	Type	Addr	Lineno	Func_sig
0	x	int32	0	4	-
1	y	float32	1	5	-

Dump scope level 0's symbol table:

Index	Name	Type	Addr	Lineno	Func_sig
0	main	func	-1	3	()V



Handle semantic error

```
1 var x int32
2
3 var z float32
4 var x int32
5 y = 8
```

error:4: x redeclared in this block. previous declaration at line 1

error:5: undefined: y

Submission

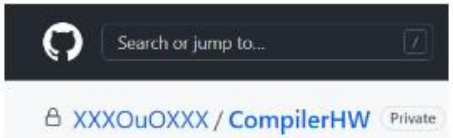
- Upload your homework to Moodle.
- The expected arrangement of your codes:
  - Only **.zip** types of compression are allowed.
  - The directory should be organized as:

```
Compiler_StudentID_HW2.zip/
└── Compiler_StudentID_HW2/
    ├── compiler_hw2.l
    ├── compiler_hw2.y
    ├── compiler_hw_common.h
    └── Makefile
```

  - You will lose 10pt if your programs were uploaded in incorrect format!!!

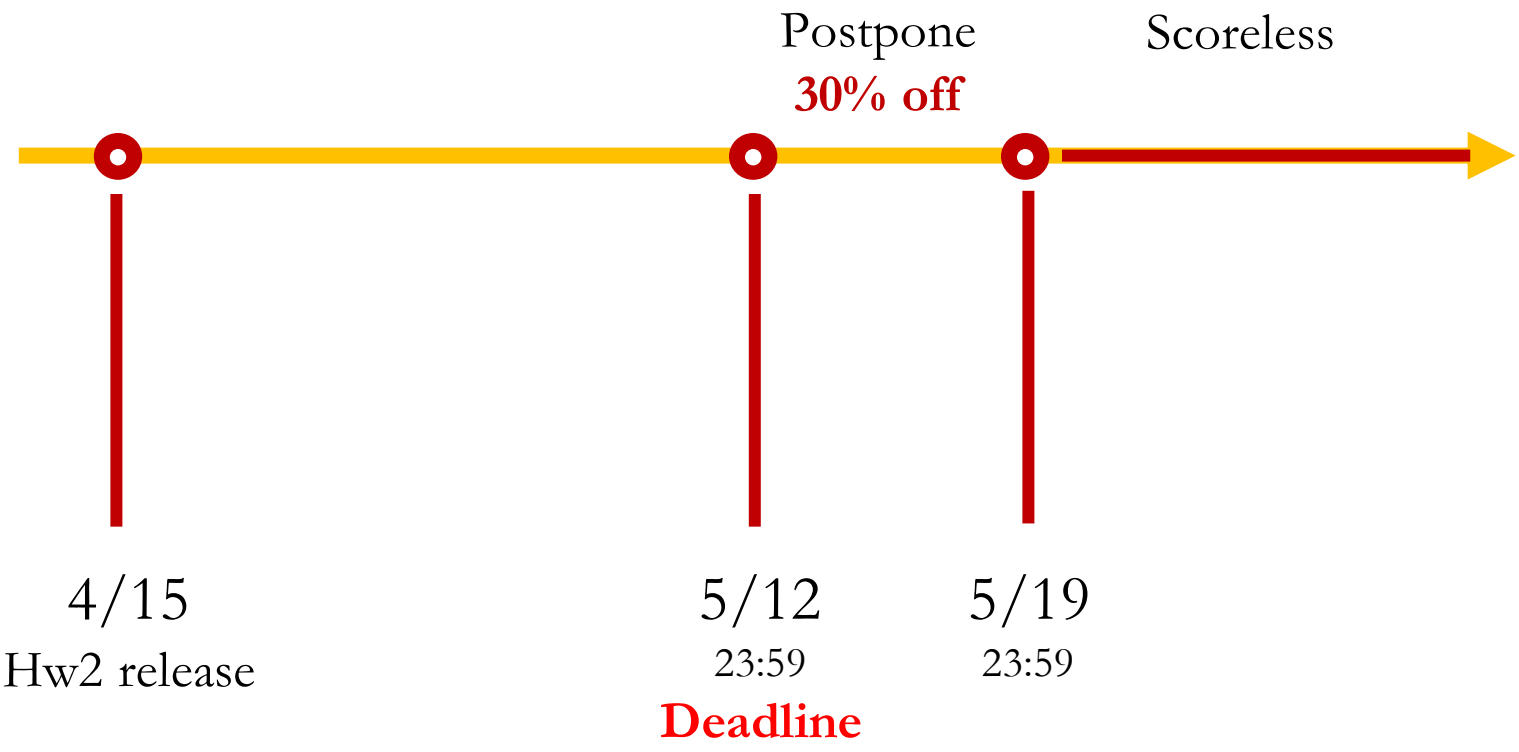


Please use “Private” repo





Deadline


Postpone
30% off
Scoreless
4/15
Hw2 release
5/12
23:59
Deadline
5/19
23:59



How to Mail TAs

- Send mail to asrlab@csie.ncku.edu.tw, not any TA's mail!!
- Email subject starts with "[Compiler2022]"



QUESTIONS ?