

CS352 Compilers: Principle and Practice

Spring 2005

PSO Session on Monday, February 21, 2005

By Mummoorthy M

NOTE: Project 2 due this week

Topics:

1. Testing
2. Testing Script

1. Testing

Remember always to test you program.

The best proven way of writing bug-free software is ‘bottom-up’ approach.

- Write method f().
- Test it thoroughly.
- Write a bigger method g() which uses f().
- Thus g() is almost proven bug-free as it uses f().
- Add one functionality at a time; test it thoroughly before adding more.
- Like building a sky-scraper: Build each floor at a time

Always remember:

“Your test case should make the program run each line at least ONCE”

Let’s take the straight line grammar:

$Stm \rightarrow Stm ; Stm$ (CompoundStm)
 $Stm \rightarrow id := Exp$ (AssignStm)
 $Stm \rightarrow print (ExpList)$ (PrintStm)
 $Exp \rightarrow id$ (IdExp)
 $Exp \rightarrow num$ (NumExp)
 $Exp \rightarrow Exp Binop Exp$ (OpExp)
 $Exp \rightarrow (Stm , Exp)$ (EseqExp)
 $ExpList \rightarrow Exp , ExpList$ (PairExpList)
 $ExpList \rightarrow Exp$ (LastExpList)
 $Binop \rightarrow +$ (Plus)
 $Binop \rightarrow -$ (Minus)
 $Binop \rightarrow \times$ (Times)
 $Binop \rightarrow /$ (Div)

You need to create a test case i.e write a program that makes your interpreter run each and every line at least ONCE

Let's build from start:

Input program	Execute what productions
i:=0	Stm: id:=Exp
Note: For other number formats use them here. Like i:=0x1, etc.	Exp is a numexp
Print(1)	Stm: print(Exp) Note: This does not cover Explist
i:=0 ; print(1)	Stm: Stm ; Stm You've tested Compound statement
At this point, you can be fairly certain that compound statement production is correct.	

In a similar way, keep on adding more test cases to test each production rules.

Don't throw away any test cases. Keep a repository of test cases and write a script (I'll explain now) to test your recent version against all test cases.