Assignment 0

Rajnish Ganguli(12551, rajnishg@iitk.ac.in),
Mahendra Meena(12390, mahendrm@iitk.ac.in),
Jitendra Nagar(12325, jnagar@iitk.ac.in)

January 21, 2018

BNF for our source language R is given below

1. *program* : *start*

2. *start* : *conditional_statements* |
   expression_statements |
   functions_statements |
   loop_statements

3. *loop_statements* : *while_loop* *start* |
   while_loop |
   for_loop *start* |
   for_loop

4. *while_loop* : **while** (*if_cond*) *compound_statement*

5. *for_loop* : **for**(**id** **in** **id**) *compound_statement*

6. *conditional_statements* : *if_else_statement* *start* |
   if_else_statement |
   if_statement *start* |
   if_statement

7. *expression_statements* : *expression* *start* |
   expression

8. $functions\_statements : function\_definition \quad start \quad |$
   $function\_definition \quad |$
   $function\_call \quad start \quad |$
   $function\_call$

9. $function\_definition : \mathbf{id} = \mathbf{function}(arg\_list) \quad compound\_statement$

10. $arg\_list : arg\_list, argument \quad |$
    $argument$

11. $argument : \mathbf{id} = any\_type \quad |$
    $\mathbf{id} \quad |$
    ϵ

12. $function\_call : \mathbf{id}(arg\_to\_pass)$

13. $arg\_to\_pass : arg\_to\_pass, arg \quad |$
    arg

14. $arg : rightside$

15. $if\_else\_statement : \mathbf{if} \quad (if\_cond) \quad compound\_statement \quad \mathbf{else} \quad if\_else\_statement \quad |$
    $\mathbf{if} \quad (if\_cond) \quad compound\_statement \quad \mathbf{else} \quad compound\_statement$

16. $if\_statement : \mathbf{if} \quad (if\_cond) \quad compound\_statement$

17. $compound\_statement : statement\_list \quad |$
    $statement$

18. $statement\_list : statement\_list \quad statement \quad |$
    $statement$

19. $statement : expression \quad |$
    $function\_call \quad |$
    $for\_loop \quad |$
    $while\_loop \quad |$
    $if\_statement \quad |$

if_else_statement

20. *expression* : **id** = *rightside* |
jump_statements |
return(*rightside*) |
print(*rightside*)

21. *jump_statements* : **break** |
next

22. *rightside* : **real** *math* *rightside* |
integer *math* *rightside* |
id *math* *rightside* |
(*rightside*) |
any_type *bitop* *rightside* |
any_type *logop* *rightside* |
vector_definition |
id *bitop* *rightside* |
id *logop* *rightside* |
id |
any_type |
function_call |
id[**id**] |
id[**integer**] |
id[**c**(*arg_to_pass*)]

23. *math* : + |
- |
***** |
/ |
^ |
%

24. *if_cond* : *bool* |
comparison_statement |
comparison_statement *logop* *if_cond* |
comparison_statement *bitop* *if_cond* |
! *if_cond*

25. *comparison_statement* : *factor* *compop* *factor* |
any_type == *any_type* |

any_type **!=** *any_type* |
id **==** *any_type* |
id **!=** *any_type* |
(*if_cond*)

26. *compop* : > |
< |
<= |
>=

27. *factor* : **id** |
real |
integer

28. *logop* : **&&** | ||

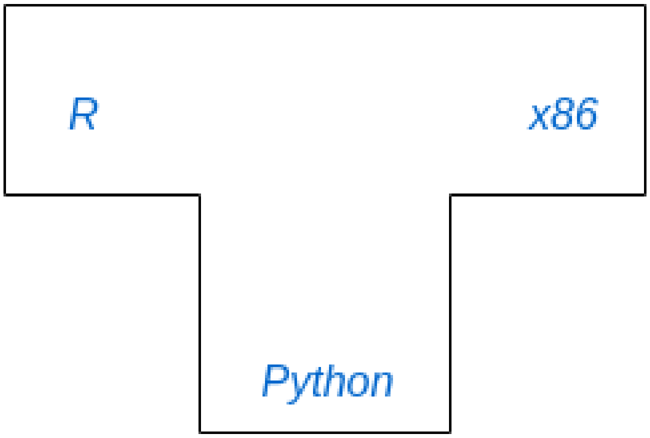
29. *bitop* : **&** | |

30. *any_type* : **real** |
string |
bool |
integer

31. *vector_definition* : **real** : **real** |
vector(, **integer**) |
vector(, **id**) |
c(*arg_to_pass*)

NOTES:

1. All tokens in bold are terminals
2. We will be using PLY.lex and PLY.yacc modules in python
3. The T-diagram for compiler is included below-


R
x86
Python