

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО» ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

**Кафедра системного програмування і спеціалізованих комп'ютерних
систем**

Лабораторна робота №2

з дисципліни «Основи проектування трансляторів»

Тема: «РОЗРОБКА ГЕНЕРАТОРА КОДУ»

Виконав: студент III курсу

ФПМ групи KB-81

Ядуха Б.В.

Викладач: Марченко О. І.

Київ 2021

Мета лабораторної роботи

Метою лабораторної роботи «Розробка генератора коду» є засвоєння теоретичного матеріалу та набуття практичного досвіду і практичних навичок розробки генераторів коду

Постановка задачі

1. Розробити програму генератора коду (ГК) для підмножини мови програмування SIGNAL, заданої за варіантом.

2. Програма має забезпечувати: • читання дерева розбору та таблиць, створених синтаксичним аналізатором, який було розроблено в розрахунково-графічній роботі; • виявлення семантичних помилок; • генерацію коду та/або побудову внутрішніх таблиць для генерації коду.

3. Входом генератора коду (ГК) мають бути: • дерево розбору; • таблиці ідентифікаторів та констант з повною інформацією, необхідною для генерації коду; • вхідна програма на підмножині мови програмування SIGNAL згідно з варіантом (необхідна для формування лістингу програми). 36

4. Виходом ГК мають бути: • асемблерний код згенерований для вхідної програми та/або внутрішні таблиці для генерації коду; • внутрішні таблиці генератора коду (якщо потрібні).

5. Зкомпонувати повний компілятор, що складається з розроблених раніше лексичного та синтаксичного аналізаторів і генератора коду, який забезпечує наступне: • генерацію коду та/або побудову внутрішніх таблиць для генерації коду; • формування лістингу вхідної програми з повідомленнями про лексичні, синтаксичні та семантичні помилки.

6. Входом компілятора має бути програма на підмножині мови програмування SIGNAL згідно з варіантом;

7. Виходом компілятора мають бути: • асемблерний код згенерований для вхідної програми та/або внутрішні таблиці для генерації коду; • лістинг вхідної програми з повідомленнями про лексичні, синтаксичні та семантичні помилки.

8. Для програмування може бути використана довільна алгоритмічна мова програмування високого рівня. Якщо обрана мова програмування має конструкції або бібліотеки для роботи з регулярними виразами, то використання цих конструкцій та/або бібліотек строго заборонено.

Варіант 19

1. <signal-program> --> <program>

2. <program> --> PROGRAM <procedure-identifier> ;

<block>.

3. <block> --> <variable-declarations> BEGIN
 <statements-list> END

4. <variable-declarations> --> VAR <declarations-list> |
 <empty>

5. <declarations-list> --> <declaration>
 <declarations-list> |
 <empty>

6. <declaration> --> <variable-identifier>:<attribute> ;

7. <attribute> --> INTEGER |
 FLOAT

8. <statements-list> --> <statement> <statementslist> |
 <empty>

9. <statement> --> <condition-statement> ENDIF ;

10. <condition-statement> --> <incompletecondition-statement><alternative-part>

11. <incomplete-condition-statement> --> IF
 <conditional-expression> THEN
 <statements-list>

12. <alternative-part> --> ELSE <statements-list> |
 <empty>

13. <conditional-expression> --> <expression> = <expression>

14. <expression> --> <variable-identifier> |

<unsigned-integer>

15. <variable-identifier> --> <identifier>

16. <procedure-identifier> --> <identifier>

17. <identifier> --> <letter><string>

18. <string> --> <letter><string> |
 <digit><string> |
 <empty>

19. <unsigned-integer> --> <digit><digits-string>

20. <digits-string> --> <digit><digits-string> |
 <empty>

21. <digit> --> 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

22. <letter> --> A | B | C | D | ... | Z

Лістинг програми ГК Test01

Input:

```
PROGRAM PR;  
    VAR V1:INTEGER;  
BEGIN  
    IF V1 = 121 THEN  
    ELSE  
    ENDIF;  
END.
```

Output:

```
.386  
.MODEL MEDIUM  
.DATA  
    V1    DD    ?  
.CODE
```

```

PR PROC
MOV AX, V1
CMP AX, 121
JNE $L0
NOP
JUMP $L1
$L0: NOP
NOP
$L1: NOP
NOP
ret
PR ENDP

```

Test02

Input:

```

PROGRAM PR;
    VAR V1:INTEGER;
        V2:FLOAT;
        V3:INTEGER;
BEGIN
    IF V1 = V3 THEN
        IF 1 = V3 THEN
            ENDIF;
        ENDIF;
    END.

```

Output:

```

.386
.MODEL MEDIUM
.DATA
    V1    DD    ?
    V2    REAL4    ?
    V3    DD    ?
.CODE
PR PROC
MOV AX, V1
CMP AX, V3

```

```

JNE $L0
MOV AX, 1
CMP AX, V3
JNE $L2
NOP
$L2: NOP
NOP
$L0: NOP
NOP
ret
PR ENDP

```

Test03

```

Input:
PROGRAM PR;
BEGIN
END.
Output:
.386
.MODEL MEDIUM
.DATA
.CODE
PR PROC
NOP
ret
PR ENDP

```

Test04

```

Input:
PROGRAM PR;
BEGIN
    IF 100 = 200 THEN
        IF 21 = 12 THEN
            ENDIF;
        ELSE
            IF 21 = 12 THEN
                ENDIF;
            ENDIF;
        END.

```

Output:

.386

.MODEL MEDIUM

.DATA

.CODE

PR PROC

MOV AX, 100

CMP AX, 200

JNE \$L0

MOV AX, 21

CMP AX, 12

JNE \$L2

NOP

\$L2: NOP

NOP

JUMP \$L1

\$L0: NOP

MOV AX, 21

CMP AX, 12

JNE \$L6

NOP

\$L6: NOP

NOP

\$L1: NOP

NOP

ret

PR ENDP

Test05

Input:

PROGRAM PR;

VAR V2:FLOAT;

V4:FLOAT;

BEGIN

IF V2 = V4 THEN

ENDIF;

END.

Output:

.386

.MODEL MEDIUM

.DATA

V2 REAL4 ?

V4 REAL4 ?

.CODE

PR PROC

MOV AX, V2

FCOM AX, V4

JNE \$L0

NOP

\$L0: NOP

NOP

ret

PR ENDP

Test06

Input:

PROGRAM PR;

VAR V1:INTEGER;

PR:INTEGER;

BEGIN

IF 100 = V2 THEN

ENDIF;

END.

Output:

.386

.MODEL MEDIUM

.DATA

V1 DD ?

PR DD ?

.CODE

PR PROC

MOV AX, 100

CMP AX, V2

JNE \$L0

NOP

\$L0: NOP

NOP

ret

PR ENDP

Code Generator : Error(line 3, column 6): Variable: PR is already defined

Code Generator : Error(line 5, column 11): Variable: V2 is not defined

Code Generator : Error(line 5, column 5): Can not compare 100 and V2
different types

Test07

Input:

PROGRAM PR;

VAR V1:INTEGER;

BEGIN

IF PR = V1 THEN

ENDIF;

END.

Output:

.386

.MODEL MEDIUM

.DATA

V1 DD ?

.CODE

PR PROC

MOV AX, PR

CMP AX, V1

JNE \$L0

NOP

\$L0: NOP

NOP

ret

PR ENDP

Code Generator : Error(line 4, column 5): Can not compare PR and V1 different
types

Test08

Input:

PROGRAM PR;

VAR V1:INTEGER;

```

        V2:FLOAT;
BEGIN
    IF V1 = V2 THEN
        ENDIF;
    IF V2 = 121 THEN
        ENDIF;
END.

```

Output:

.386

.MODEL MEDIUM

.DATA

V1 DD ?

V2 REAL4 ?

.CODE

PR PROC

MOV AX, V1

CMP AX, V2

JNE \$L0

NOP

\$L0: NOP

MOV AX, V2

FCOM AX, 121

JNE \$L0

NOP

\$L0: NOP

NOP

ret

PR ENDP

Code Generator : Error(line 5, column 5): Can not compare V1 and V2 different types

Code Generator : Error(line 7, column 5): Can not compare V2 and 121 different types