

## Лекція №15

### Генерація коду для простого оператора присвоювання

Розглянемо семантику і генерацію коду для простого оператора присвоювання, описаного такою граматикою:

1.  $\langle \text{оператор} \rangle \rightarrow \langle \text{змінна} \rangle := \langle \text{вираз} \rangle$
2.  $\langle \text{вираз} \rangle \rightarrow \langle \text{змінна} \rangle$
3.  $\langle \text{вираз} \rangle \rightarrow \langle \text{вираз} \rangle \langle \text{зод} \rangle \langle \text{змінна} \rangle$
4.  $\langle \text{змінна} \rangle \rightarrow a1$
5.  $\langle \text{змінна} \rangle \rightarrow a2$
6.  $\langle \text{змінна} \rangle \rightarrow a3$
7.  $\langle \text{змінна} \rangle \rightarrow a4$
8.  $\langle \text{змінна} \rangle \rightarrow a5$
9.  $\langle \text{зод} \rangle \rightarrow +$
10.  $\langle \text{зод} \rangle \rightarrow -$

### Неформальна семантика

Нехай операція присвоювання визначена тільки для цілих чисел і визначені тільки операції додавання і віднімання.

Наведемо семантичні відповідності між компонентами оператора присвоювання і командами асемблера:

Операції	Команди
$:=$	MOV $\langle \text{змінна} \rangle$ ,AX
+	ADD AX, $\langle \text{змінна} \rangle$
-	SUB AX, $\langle \text{змінна} \rangle$
Отримання значення змінної	MOV AX, $\langle \text{змінна} \rangle$

### Структури даних для генератора коду

Вхідні: RAS : array [1..RAS\_Length] of integer;

Identifiers : array [4..8] of string[2]; {вектор ідентифікаторів  $a_i$ }

Вихідні: WPR : array [1..WPR\_Length] of

record {команда асемблера}

LAB : string[5]; {поле мітки}

COP : string [5]; {поле коду операції}

OPR : string [10]; {поле операндів}

end;

Допоміжні:

BF : string[2]; {буфер для зберігання ідентифікатора змінної}

j : word; {поточний індекс вектора WPR}

Zn : char; {знак}

## Семантична процедура SPR оператора присвоювання

```

Procedure SPR (u : integer);
begin
    if RAS[u] < 0 then k := - RAS[u]
        else k := u;
    i := RAS[k];
    k1 := k+1; k2 := k+2; k3 := k+3;
    case i of
    1: (* <оператор> → <змінна> := <вираз> *)
        begin
            SPR (k1);      (*результат записується в AX*)
            SPR (k2);      (*запис імені в BF*)
            (*генерація команди*)
            WPR[j].LAB := '____';
            WPR[j].COP := 'MOV____';
            WPR[j].OPR := BF + ',AX';
            j := j +1;
        end;
    2: (* <вираз> → < змінна> *)
        begin
            SPR(k1);      (*значення змінної в BF*)
            (* генерація команди завантаження *)
            WPR[j].LAB := '____';
            WPR[j].COP := 'MOV____';
            WPR[j].OPR := 'AX,' + BF;
            j := j +1;
        end;
    3: (* <вираз> → <вираз> <зод> < змінна> *)
        begin
            SPR (k3);      (*результат записується в AX*)
            SPR (k2);      (*знак записується в Zn *)
            SPR (k1);      (*значення імені записується в BF *)

            (* генерація команди *)
            WPR[j].LAB := '____';
            If Zn = '+' then WPR[j].COP := 'ADD____'
                else WPR[j].COP := 'SUB____';
            WPR[j].OPR := 'AX,' + BF;
            j := j + 1;
        end;
    4...8: (* < змінна>→ a1|...|a5 *)
        BF := Identifiers[i];
    9: (* <зод> → + *)
        Zn := '+';
    10: (* <зод> → - *)
        Zn := '-';
    end; {case}
end; {SPR}

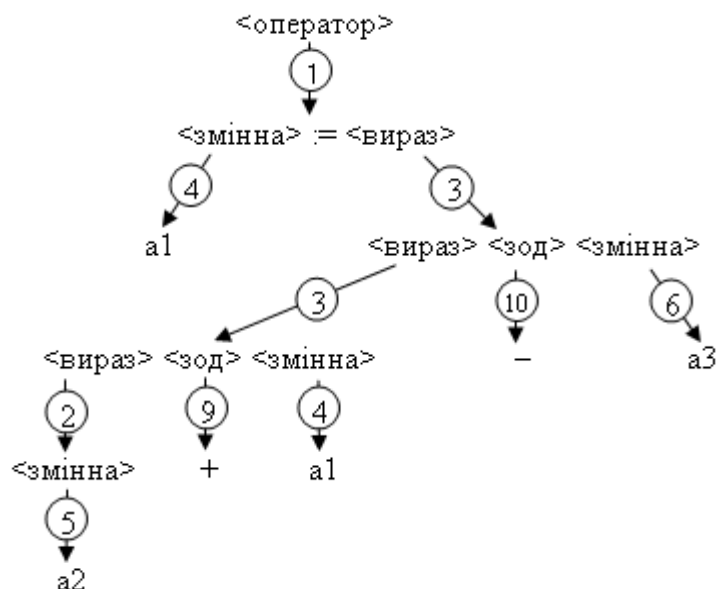
```

## Програма генератора коду

```
Program SP;  
  (* опис даних *)  
  (* опис процедури SPR*)  
begin  
  j := 1;  
    (* Введення значення LRAS – довжини вектора RAS *)  
    (* Введення вектора RAS *)  
    SPR (LRAS);  
    (* Вивід результуючої програми *)  
end;
```

## Перевірка роботи генератора коду

Контрольний приклад  $a1 := a2 + a1 - a3$ .



Нарисуем вектор RAS

															LRAS	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
9	4	10	6	3	-4	-3	3	-2	-1	2	5	1	-5	4	-13	

Вихід генератора коду (код результуючої асемблерної програми):

```
MOV AX,a2  
ADD AX,a1  
SUB AX,a3  
MOV a1,AX
```

