

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Вятский государственный университет»
Факультет автоматики и вычислительной техники
Кафедра электронных вычислительных машин

Отчет
Лабораторная работа №5 по дисциплине
«Системное программное обеспечение»

Выполнил студент группы ИВТ-32 _____/Рзаев А. Э./
Проверил доцент кафедры ЭВМ _____/Караваева О. В./

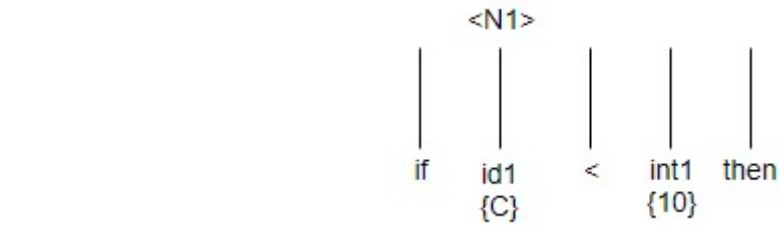
Киров 2017

Построить дерево методом восходящего разбора:

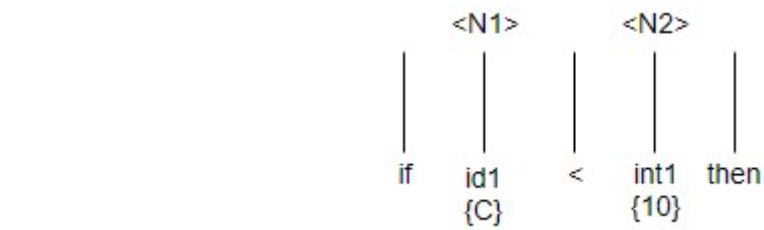
if C < 10 then C := B - 4 else A := (C - 15) + (B + 4);



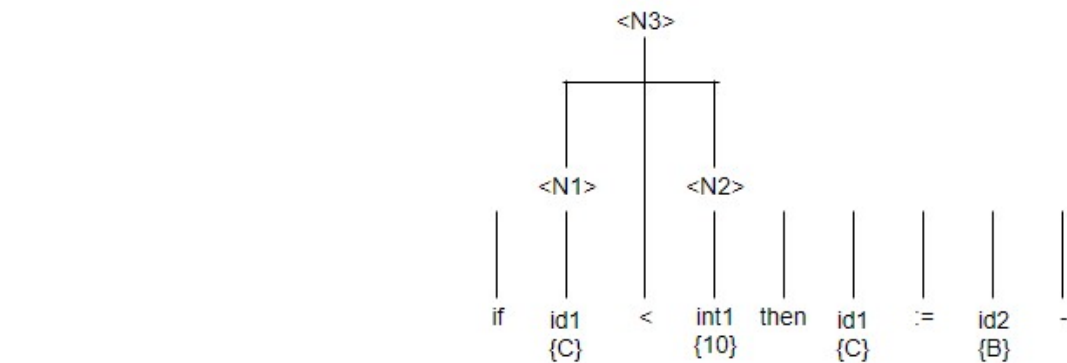
... if id1 <



... if <N1> < int1 then

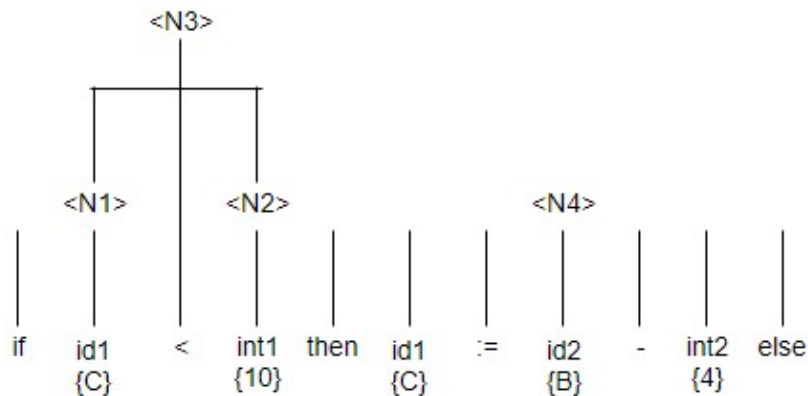


... if <N1> < <N2> then

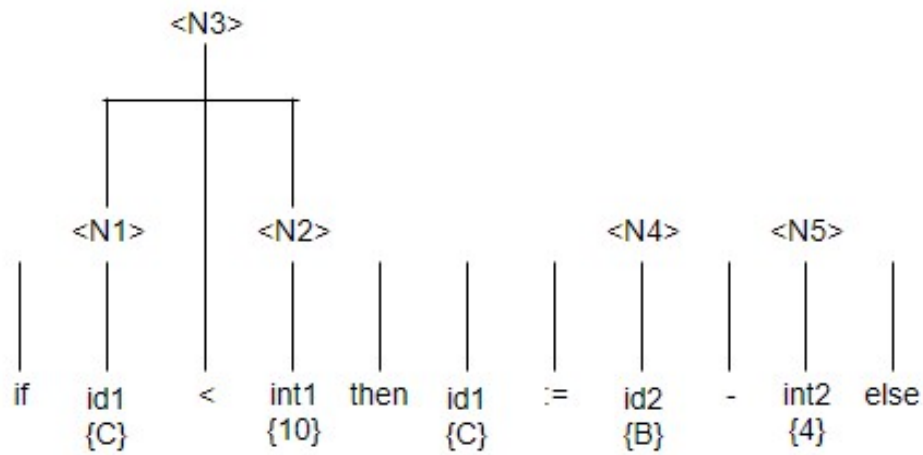


... if <N3> then id1 := id2 -

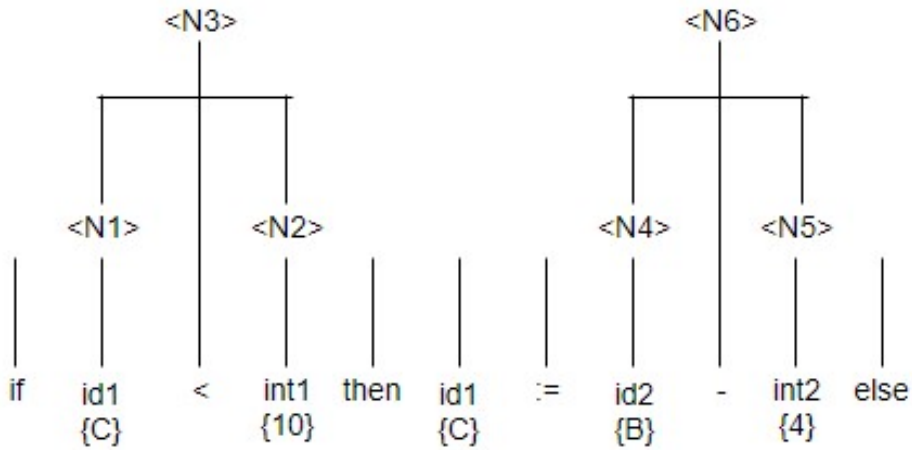
... if <N3> then id1 := <N4> - int2 else



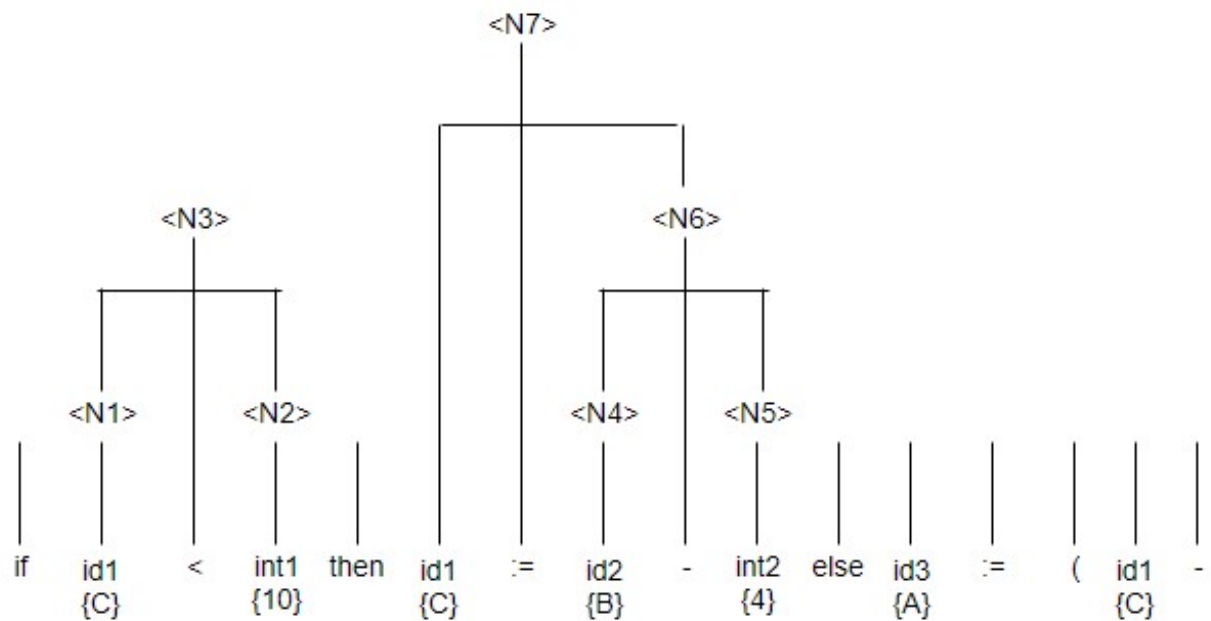
... if <N3> then id1 := <N4> - <N5> else



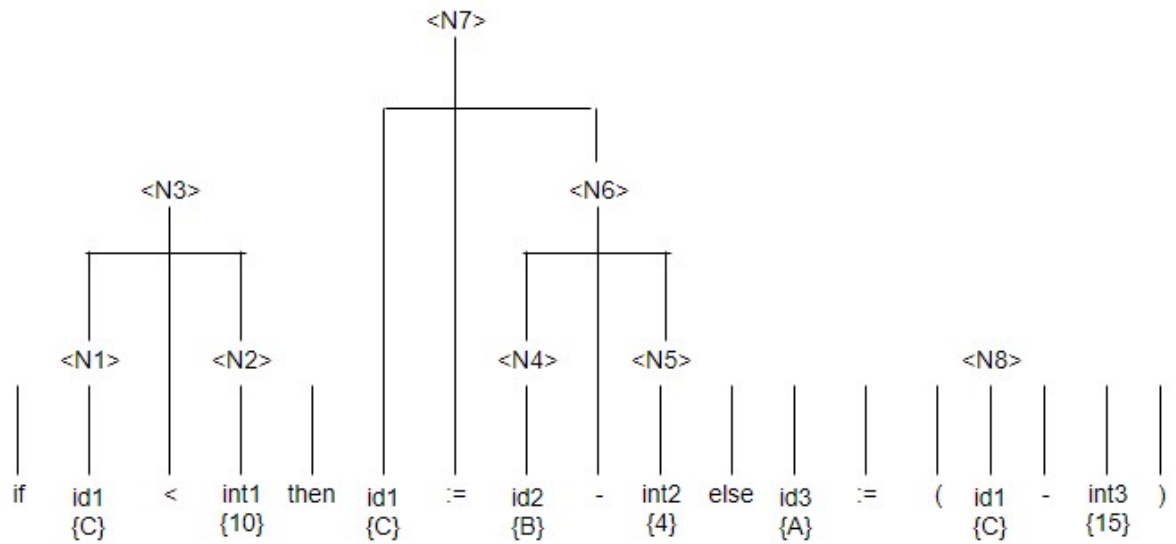
... if <N3> then id1 := <N6> else



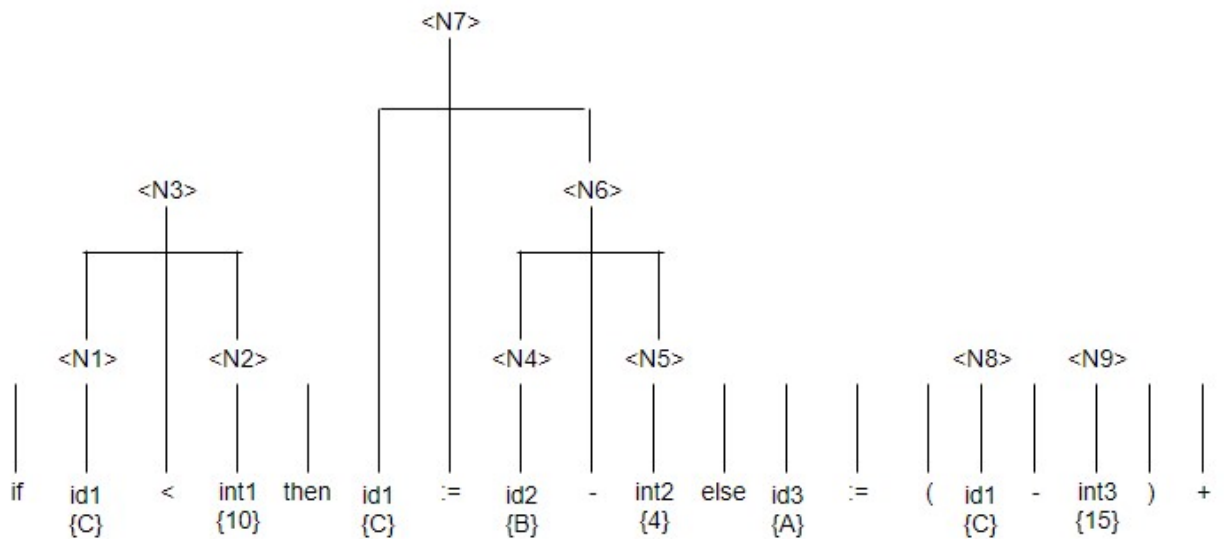
... if <N3> then <N7> else id3 := (id1 -



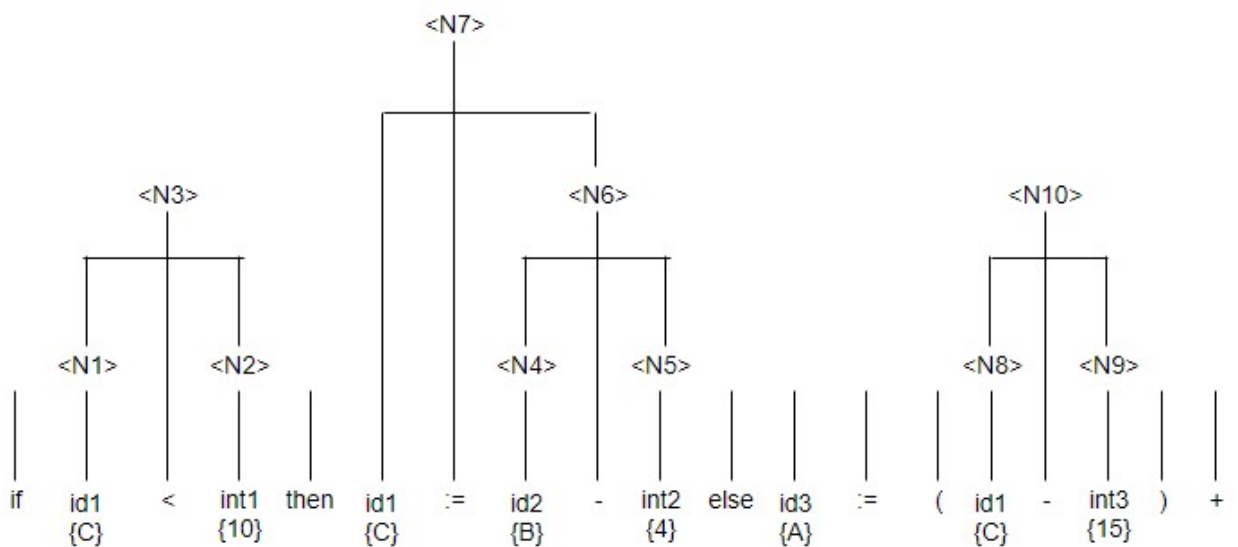
... if <N3> then <N7> else id3 := (<N8> - int3)



... if <N3> then <N7> else id3 := (<N8> - <N9>) +



... if <N3> then <N7> else id3 := (<N10>) +



The diagram shows an Abstract Syntax Tree (AST) for the given expression. The root node is $\langle N7 \rangle$, which branches into three children: $\langle N3 \rangle$, $\langle N6 \rangle$, and $\langle N10 \rangle$.

- $\langle N3 \rangle$ branches into $\langle N1 \rangle$ and $\langle N2 \rangle$.
 - $\langle N1 \rangle$ corresponds to the expression `id1 {C}`.
 - $\langle N2 \rangle$ corresponds to the expression `< int1 {10}`.
- $\langle N6 \rangle$ branches into $\langle N4 \rangle$ and $\langle N5 \rangle$.
 - $\langle N4 \rangle$ corresponds to the expression `id2 {B}`.
 - $\langle N5 \rangle$ corresponds to the expression `- int2 {4}`.
- $\langle N10 \rangle$ branches into $\langle N8 \rangle$ and $\langle N9 \rangle$.
 - $\langle N8 \rangle$ corresponds to the expression `(id1 {C} - int3 {15})`.
 - $\langle N9 \rangle$ corresponds to the expression `+ (id2 {B} + int1 {10})`.

The final expression is constructed by combining these sub-expressions according to the tree structure: `if id1 {C} < int1 {10} then id1 {C} := id2 {B} - int2 {4} else id3 {A} := (id1 {C} - int3 {15}) + (id2 {B} + int1 {10})`.

```

graph TD
    N7["<N7>"] --- N3["<N3>"]
    N7 --- N6["<N6>"]
    N7 --- id1_1["id1 {C}"]
    N7 --- op1[":="]
    N7 --- id2_1["id2 {B}"]
    N7 --- op2["-"]
    N7 --- int2["int2 {4}"]
    N7 --- else["else"]
    N7 --- id3["id3 {A}"]
    N7 --- op3[":="]

    N3 --- N1["<N1>"]
    N3 --- N2["<N2>"]

    N1 --- id1_2["id1 {C}"]
    N1 --- op4["<"]

    N2 --- int1["int1 {10}"]

    N6 --- N4["<N4>"]
    N6 --- N5["<N5>"]

    N4 --- id2_2["id2 {B}"]

    N5 --- op5["-"]
    N5 --- int3["int3 {15}"]

    N11["<N11>"] --- N10["<N10>"]
    N11 --- N12["<N12>"]
    N11 --- N13["<N13>"]

    N10 --- N8["<N8>"]
    N10 --- N9["<N9>"]

    N8 --- op6["("]
    N8 --- id1_3["id1 {C}"]
    N8 --- op7["-"]
    N8 --- int3_2["int3 {15}"]
    N8 --- op8[")"]

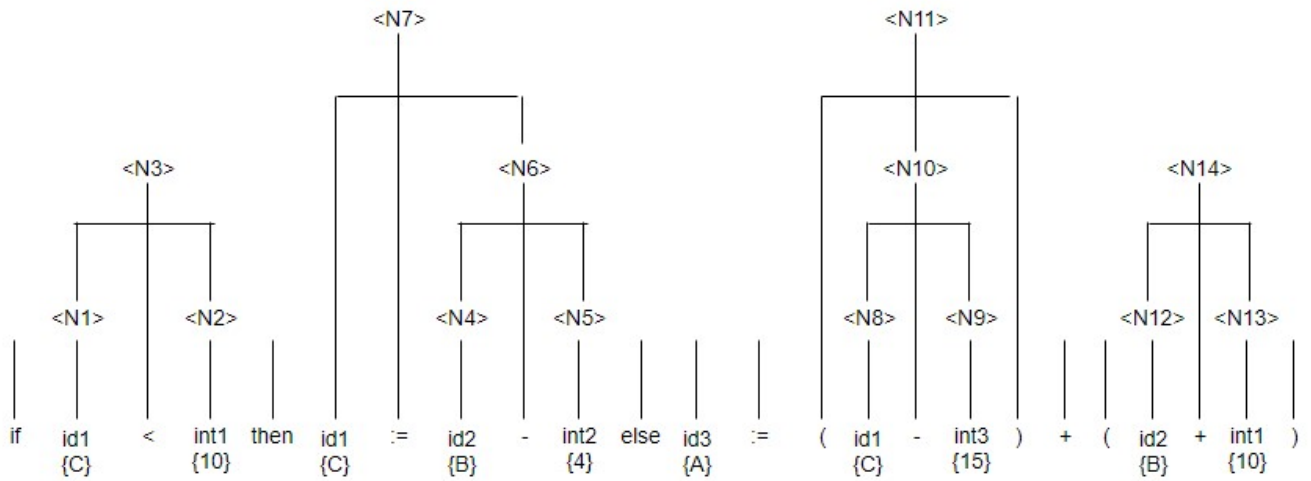
    N9 --- op9["+"]

    N12 --- op10["("]
    N12 --- id2_3["id2 {B}"]
    N12 --- op11["+"]
    N12 --- int1_2["int1 {10}"]
    N12 --- op12[")"]

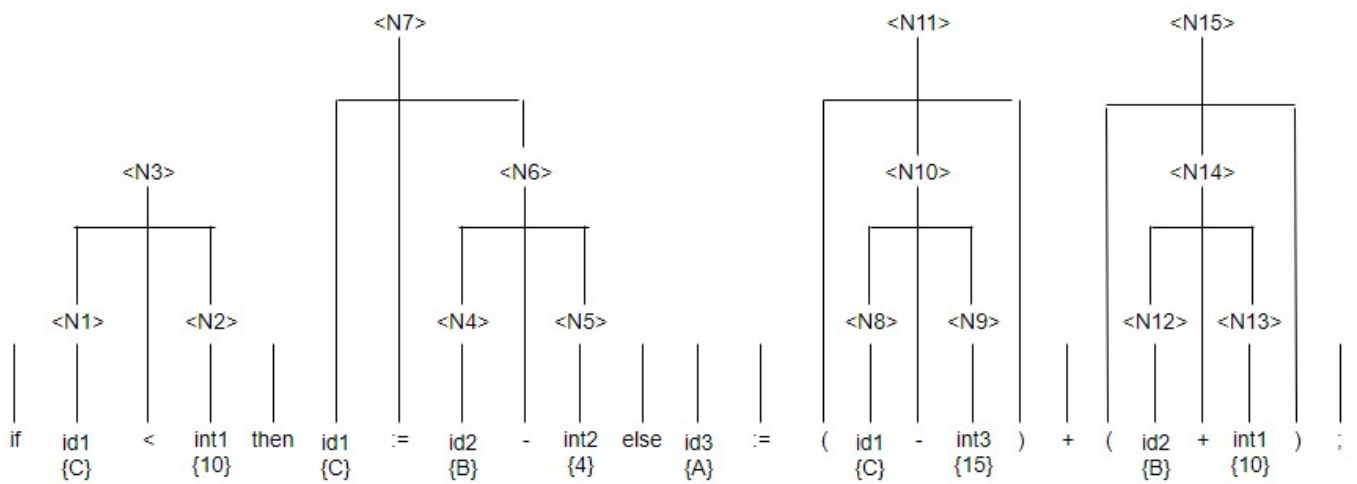
    N13 --- int1_3["<10>"]

    if["if"] --- id1_4["id1 {C}"]
    if --- op13["<"]
    if --- int1_4["int1 {10}"]
    if --- then["then"]
    if --- id1_5["id1 {C}"]
    if --- op14[":="]
    if --- id2_4["id2 {B}"]
    if --- op15["-"]
    if --- int2_2["int2 {4}"]
    if --- else_2["else"]
    if --- id3_2["id3 {A}"]
    if --- op16[":="]
    if --- op6_2["("]
    if --- id1_6["id1 {C}"]
    if --- op17["-"]
    if --- int3_3["int3 {15}"]
    if --- op18[")"]
    if --- op19["+"]
    if --- op10_2["("]
    if --- id2_5["id2 {B}"]
    if --- op20["+"]
    if --- int1_5["int1 {10}"]
    if --- op21[")"]
    if --- int1_6["<10>"]
  
```

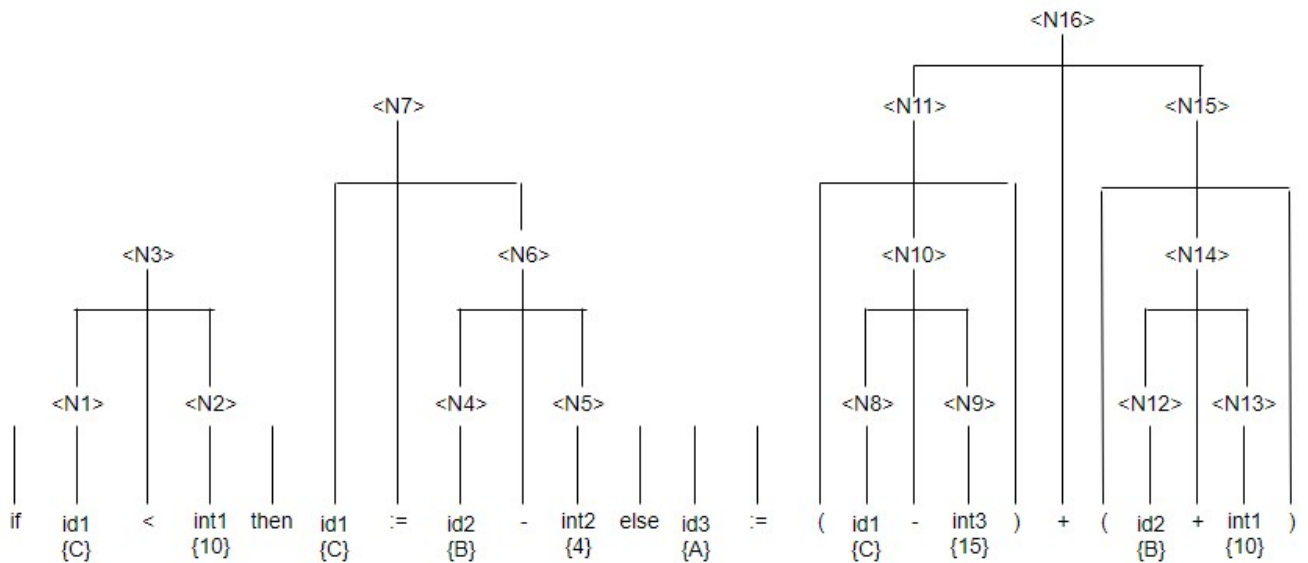
... if <N3> then <N7> else id3 := <N11> + (<N14>)



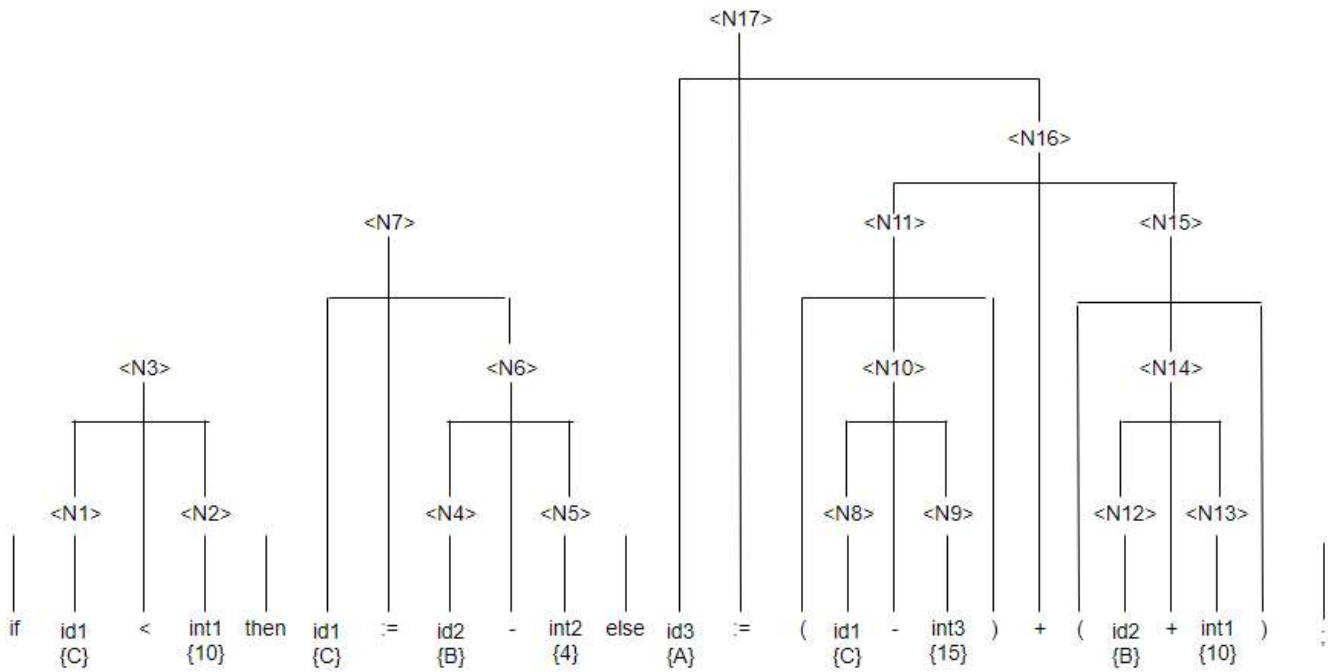
... if <N3> then <N7> else id3 := <N11> + <N15> ;



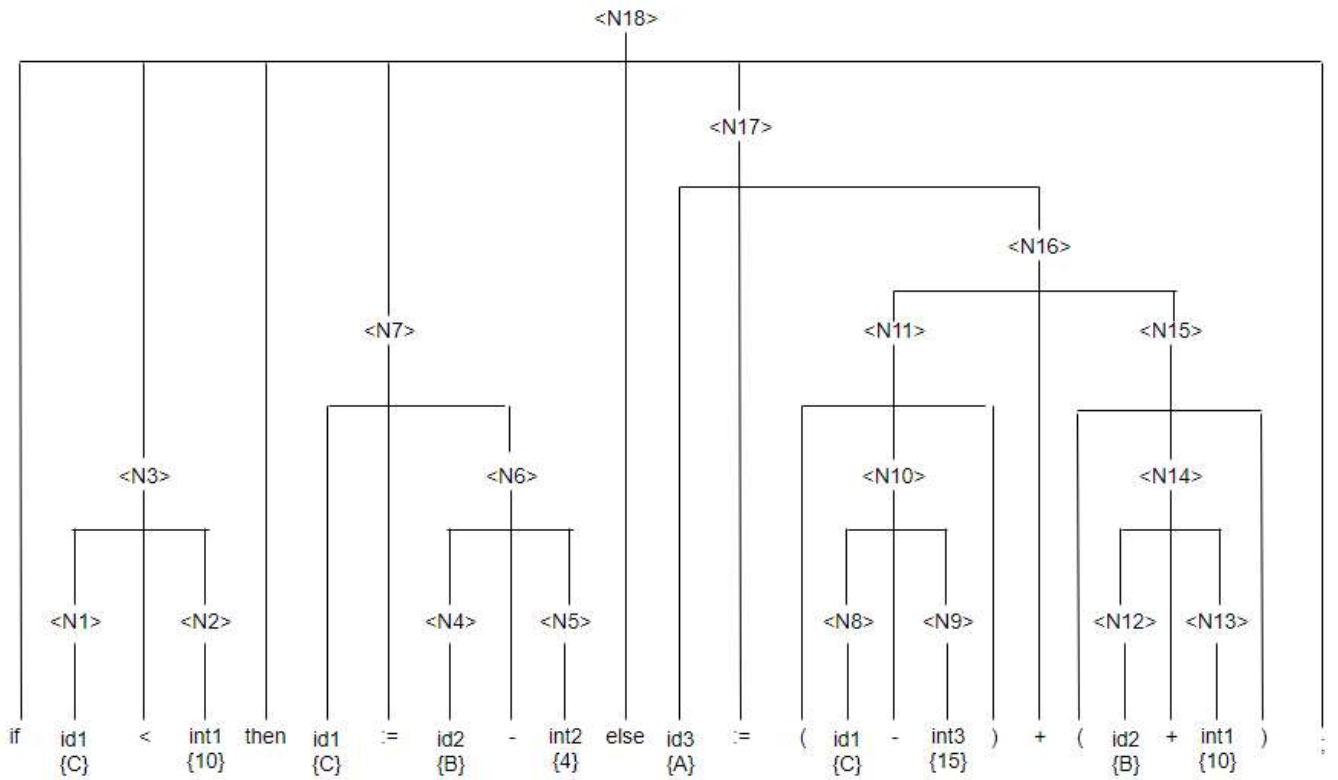
... if <N3> then <N7> else id3 := <N16> ;



... if <N3> then <N7> else <N17> ;



... <N18> ;



Оптимизация

$$A1 = (A2 + A4) * (A3 - A5)$$

$$A2 = (A2 + A4) / 17$$

$$A4 = A1 * A2 - A5$$

$$A5 = (A2 + A4) / (A3 - A5)$$

Исходные триады	Зависимости переменных					dep(i)	Полученные триады
	A1	A2	A3	A4	A5		
(1) + A2 A4	0	0	0	0	0	1	(1) + A2 A4
(2) – A3 A5	0	0	0	0	0	1	(2) – A3 A5
(3) * (1) (2)	0	0	0	0	0	1	(3) * (1) (2)
(4) = A1 (3)	0	0	0	0	0	1	(4) = A1 (3)
(5) + A2 A4	4	0	0	0	0	1	(5) Same (1)
(6) / (5) 17	4	0	0	0	0	2	(6) / (5) 17
(7) = A2 (6)	4	0	0	0	0	3	(7) = A2 (6)
(8) * A1 A2	4	7	0	0	0	8	(8) * A1 A2
(9) – (8) A5	4	7	0	0	0	9	(9) – (8) A5
(10) = A4 (9)	4	7	0	0	0	10	(10) = A4 (9)
(11) + A2 A4	4	7	0	10	0	11	(11) + A2 A4
(12) – A3 A5	4	7	0	10	0	1	(12) Same (2)
(13) / (11) (12)	4	7	0	10	0	12	(13) / (11) (12)
(14) = A5 (13)	4	7	0	10	0	13	(14) = A5 (13)