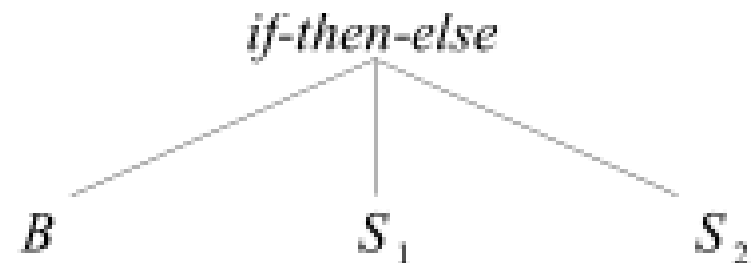
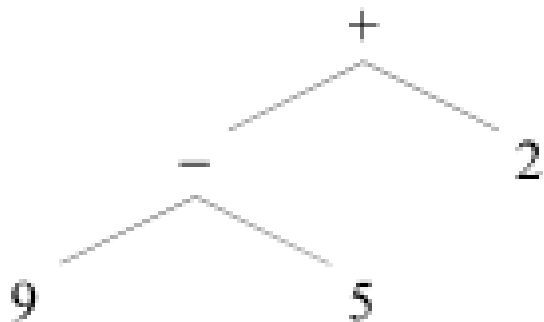


SYNTAX TREE

- Parse tree sering kali disebut sebagai concrete syntax tree. Parse tree tidak efisien karena mengandung informasi yang berlebihan, sebagai gantinya, kalimat dinyatakan sebagai abstract syntax tree atau seringkali disebut sebagai syntax tree saja. Dalam syntax tree setiap operator maupun keyword tidak muncul sebagai leaf. Berikut ini adalah contoh syntax tree untuk kalimat $x: 9 - 5 + 2$ dan $y: \text{if } B \text{ then } S_1 \text{ else } S_2$:



MEMBUAT SYNTAX TREE SEBUAH EKSPRESI

- Pada pasal ini akan dibentuk sebuah syntax tree dimana setiap node-nya dinyatakan sebagai data record. Setiap record paling tidak mengandung sebuah label yang dapat berupa identifier, onstanta, atau operator. Proses pembentukan ini memerlukan tiga buah finction berikut :
 1. Mknode (op,left,right) : membentuk node operator dengan label op dan dua field lainnya berupa pointer ke anak kiri dan anak kanan
 2. Mkleaf (id,entry) : membentuk node identifier dengan label id dan sebuah field berupa pointer ke symbol table. Pelacakan identifier pada symbol table.
 3. Mkleaf (num,val) : membentuk node konstanta dengan label id dan sebuah field yang mengandung nilai bilangan.

Deskripsi instruksi three-address menentukan komponen dari setiap jenis instruksi, tetapi tidak representasi dari instruksi dalam struktur data. Dalam kompilator, petunjuk ini dapat diimplementasikan sebagai objek atau sebagai catatan untuk operator dan operan. Tiga representasi itu disebut “quadruples”, “triples”, and “indirect triples”.

Three-Address Code

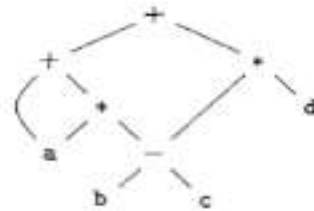
Pada **Three- Address Code**, **operator** pada sisi sebelah kanan akan terbentuk menjadi 3 alamat intruksi, misalkan $x+y*x$ akan menjadi :

$$T_1 = y * z$$

$$T_2 = x + t_1$$

Dimana t_1 dan t_2 adalah nama sementara untuk tempat penyimpanan data.

Contoh :



(a) DAG

```
t1 = b - c
t2 = a * t1
t3 = a + t2
t4 = t1 * d
t5 = t3 + t4
```

(b) Three-address code

Quadruples

Sebuah quadruple (atau hanya “quaff”) memiliki empat bidang, yang kita sebut *op*, *arg1*, *arg2*, dan hasilnya/result.

$a = b * -c + b * -c$ menjadi :

```
t1 = minus c
t2 = b * t1
t3 = minus c
t4 = b * t3
t5 = t2 + t4
a = t5
```

(a) Three-address code

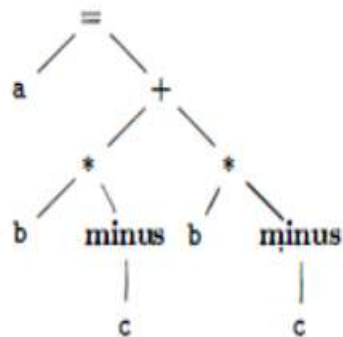
	<i>op</i>	<i>arg₁</i>	<i>arg₂</i>	<i>result</i>
0	minus	c		t ₁
1	*	b	t ₁	t ₂
2	minus	c		t ₃
3	*	b	t ₃	t ₄
4	+	t ₂	t ₄	t ₅
5	=	t ₅		a
		...		

(b) Quadruples

TRIPLES

Sebuah triple hanya memiliki tiga bidang, yang disebut op, arg1, dan arg2

$$a + a * (b - c) + (b - c) * d$$



(a) Syntax tree

	<i>op</i>	<i>arg₁</i>	<i>arg₂</i>
0	minus	c	
1	*	b	(0)
2	minus	c	
3	*	b	(2)
4	+	(1)	(3)
5	=	a	(4)
	...		

(b) Triples

Indirect triples

Indirect triples terdiri dari daftar pointer ke triple, contoh :

<i>instruction</i>	
35	(0)
36	(1)
37	(2)
38	(3)
39	(4)
40	(5)
	...

	<i>op</i>	<i>arg₁</i>	<i>arg₂</i>
0	minus	c	
1	*	b	(0)
2	minus	c	
3	*	b	(2)
4	+	(1)	(3)
5	=	a	(4)
		...	