Département de Mathématiques, Informatique et Gestion
Filière : Sciences Mathématiques et Informatique (SMI/S5)
Module : Compilation
A.U. : 2017/2018

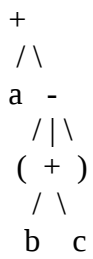
TD : Série 4, Sollutions

Exercice 1 :

Convertir l'expression a+-(b+c) en :

- Un arbre syntaxique.
- Quadruplets.
- Triplets.
- Triplets indirect.

- Arbre abstrait



- Quadruplets : t1=b+c, t2=-t1, t2=-t1 et t3= a+t2

op	arg1	arg2	res
+	b	c	t1
-	t1		t2
+	a	t2	t3

- Triplet

	op	arg1	arg2
0	+	b	c
1	-	0	
2	+	a	1

- Triplet indirect

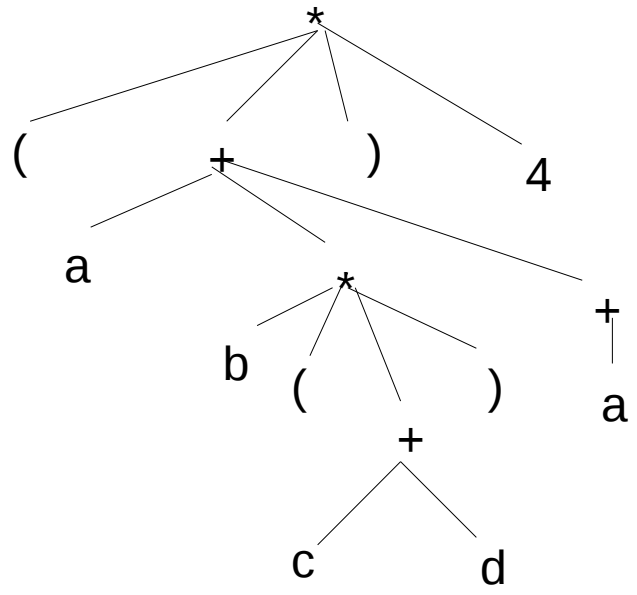
0
1
2

op	arg1	arg2
+	b	c
-	0	
+	a	1

(examen 2014/2015) :

Convertir l'expression (a+b*(c+d)+a)*4 en :

1. Arbre abstrait



2. Triplet

	op	arg1	arg2
0	+	c	d
1	*	b	0
2	+	a	1
3	+	a	2
4	*	4	3

3. Triplet indirect

0
1
2
3
4

op	arg1	arg2
+	c	d
*	b	0
+	a	1
+	a	2
*	4	3

Exercice 2 :

Soit le bloc du code en C suivant :

```
sum=0 ;
i=1 ;
l2 : if(i<=10)
{
     j=0 ;
l1 : if(j<1)
{
sum +=j;
j=j+i;
goto l1;
}
ses = (i+1)*sum
i +=1;
goto l2;
}
```

Donner le code intermédiaire correspondant.

Le code post fixé

```
l2 : i 10 l3 ble
l3 : j 0 =
l1 : j 1 blt
     sum sum  j + =
     j j i + =
l1 br
ses i 1 + sum * =
i i 1 + =
l2 br
l4
```

(examen 2015/2016) :

Donner le code intermédiaire correspondant au bloc du code en C suivant :

```
if (x < 100 || x > 200 && x!=y) x=0 ;
```

Ce code peut être écrit comme

```
     if x < 100 goto L2
     goto L3
L3 : if x > 200 goto L4
     goto L1
L4 : if x!=y goto L2
     goto L1
L2 : x=0
L1 :
```

Le code post fixé

```
x 100 L2 blt
  L3 br
L3 : x 200 L4 bgt
  L1 br
L4 : x y L2 bne
  L1 br
L2 : x 0 =
L1 :
```

Exercice 3 (examen 2014/2015) :

Donner le code intermédiaire correspondant au bloc du code en C suivant :

```
{
     int i; int j; float[100] a; float v; float x;

     while ( true ) {
          do i = i+1; while ( a[i] < v );
          do j = j-1; while ( a[j] > v );
          if ( i >= j ) break;
          x = a[i]; a[i] = a[j]; a[j] = x;
     }
}
```

Le code peut être réécrit de la façon suivante	Le code post fixé

L1: L3: i = i + 1
if(a[i] < v) goto L3
L4: j = j – 1
if(a[j] > v) goto L4
if(i>=j) goto L5
goto L2
L5: x= a[i]
a[i] = a[j]
a[j]=x
goto L1
L2:
	L1: L3: i i 1 + =
	a[i] v L3 blt
	L4: j j 1 - =
	a[j] v L4 bgt
	i j L5 bge
	L2 br
	L5: x a[i] =
	a[i] a[j] =
	a[j] x =
	L1 br
	L2: