

**TD N° 4 :
Formes Intermédiaires**

Exercice 1 :

Traduire les expressions suivantes en code postfixé, quadruplés, et arbres abstraits :

- 1- $a := (a + b * c) + ((d + b * c) - f * g) + f * g * d$
- 2- $a := a + b * ((d + a) / (d + b)) * c$

Exercice 2:

Traduire les expressions suivantes en code postfixé, quadruplés et arbres abstraits :

1. Begin Integer Array B[x:x+y];
Repeat J:= J+1;
 $x := B[x+1] * 4$;
Until (J<=20) or (x>10);
End;
2. Begin If (a<b*c)
 Then While a<=b*c
 Do Begin a:=a+1;
 a:=b*c;
 End;
 Else a:=b*c;
End;

Exercice 3:

Traduire l'expression booléenne suivante en quadruplés en utilisant les opérateurs BZ, BNZ, BR et :=
 $V := X \text{ and } Y \text{ or } Z \text{ and } Y \text{ or not } X$

$$TDH^0 = 4$$

Exos:

$$a := (\underbrace{a+b}_{1 \rightarrow 2} * c) + (\underbrace{(d+b}_{4 \rightarrow 3} * c) - \underbrace{f-g}_{5 \rightarrow 6}}_{7 \rightarrow 6} + \underbrace{f}_{8 \rightarrow 9} * g * d$$

1 - postfix:

$$\text{Ex: } \text{fp}(\text{ops} \cdot \text{op}_1 \text{op}_2) = \text{op} \cdot \text{op}_1 \cdot \text{op}_2$$

$$a \quad \underbrace{abc}_{1 \rightarrow 2} * + \underbrace{dbc}_{3 \rightarrow 4} * + \underbrace{fg}_{5 \rightarrow 6} * - + \underbrace{fg}_{7 \rightarrow 8} * d * + :=$$

$$\text{fp}(\text{EXP}) = a * \text{fp}(\text{EXP}_4) \text{fp}(\text{EXP}_5) - + \text{fp}(\text{EXP}_6) + :=$$

$$\equiv a \quad \boxed{abc * +} \quad \boxed{\text{fp}(\text{EXP}_4) \text{fp}(\text{EXP}_5) -} \quad + \quad \boxed{fg * d * +} :=$$

$$\equiv a \quad abc * + \quad dbc * + \quad fg * - + \quad fg * d * + :=$$

(1)

2/ The quadruplet:

~~Exp~~: ~~(op₁~~

Exp: Quad (op₁ op op₂)

= (op op₁ op₂ , tmp)

1) (* , b , c , t₁)

2) (+ a t₁ t₂)

3) (* b c t₃)

4) (+ d t₃ t₄)

5) (* , f , g , t₅)

6) (- t₄ t₅ , t₆)

7) (+ t₂ t₆ t₇)

8) (* , f , g , t₈)

9) (* , t₈ , d , t₉)

10) (+ , t₇ , t₉ , t₁₀)

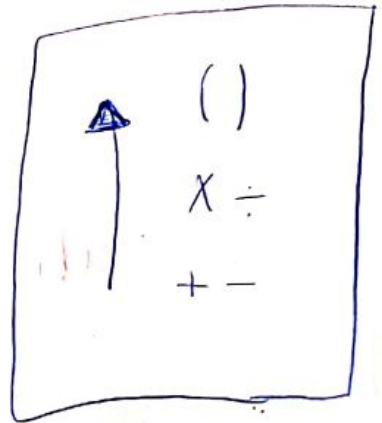
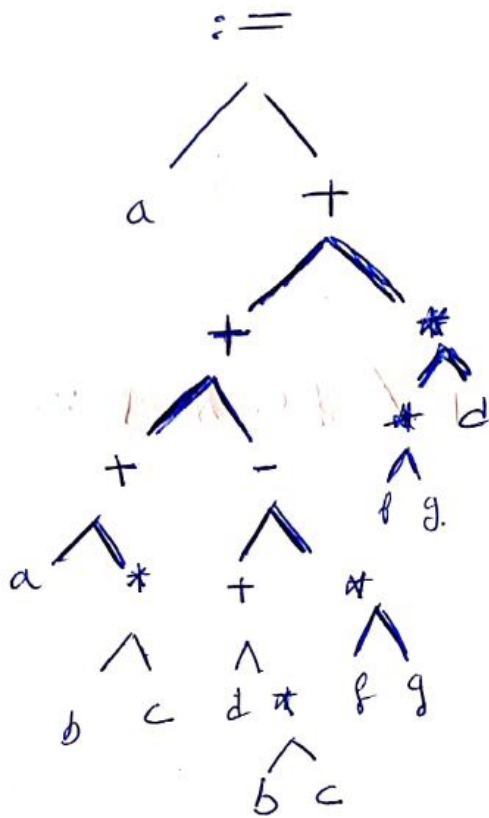
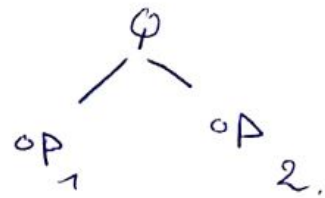
10) (:= t₁₀ , a)

Andres abstracts:

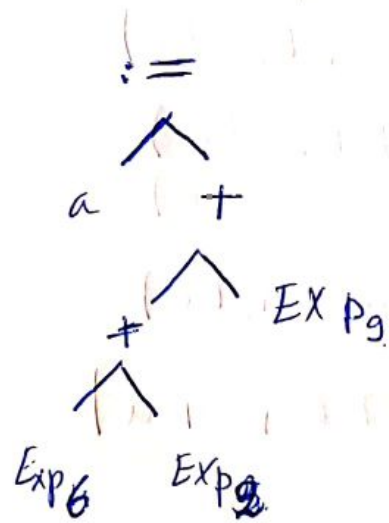
3)

~~4~~

op_1 op op_2



sinon ob fait

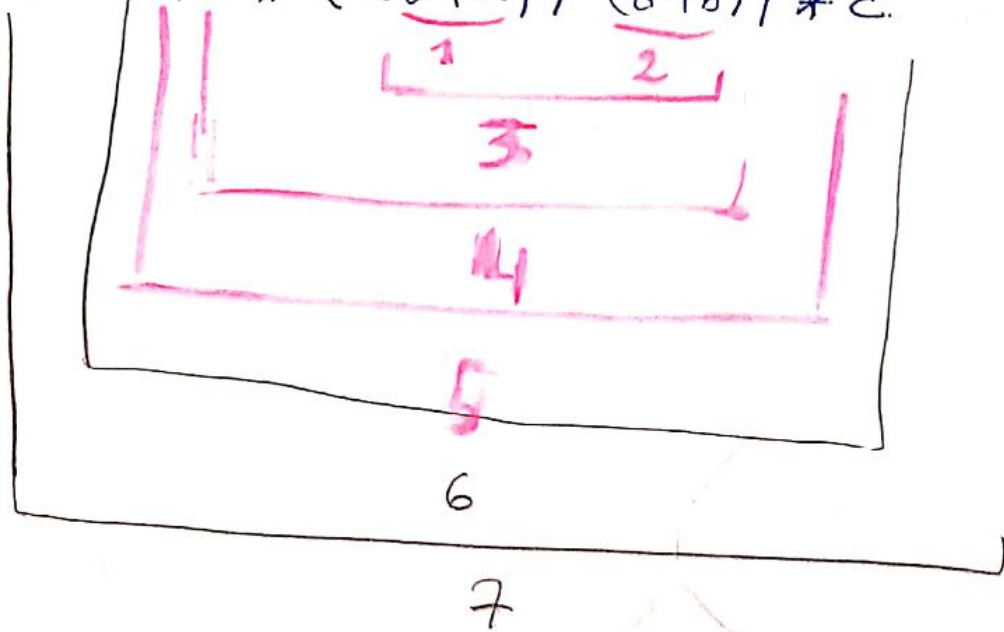


②

2

$$a := a + b * ((d + a) / (d + b)) * c.$$

postfixe:



1) a ab da + db + / * c * + :=

2) Q vandumpbt:

1) (+ d a t₁)

2) (+ d b t₂)

~~3)~~

3) (/ t₁, t₂, t₃)

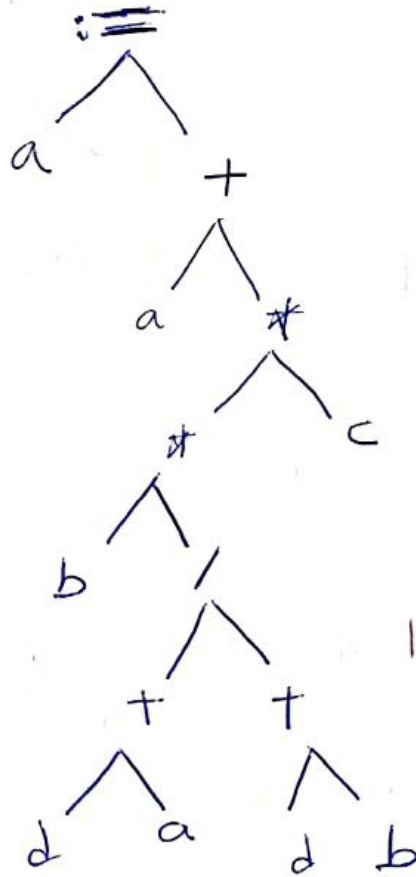
4) (* b t₃ t₄)

5) (* t₄ c t₅)

6) (+ a t₅ t₆)

7) (:= t₆, a)





Exo $h^0 = 2$:

1/

Begin	
Integer Array $B[X : X + Y]$	partie. Declaration.
Repeat	
$j = j + 1;$ $x = B[x + 1] * 4;$	Inst
Until $(j \leq 20) \text{ on } (x > 10)$	Condition.
End.	

* $fp(dec)$ — dec inst
 * $fp(Inst)$
 * $fp(Condition)$
 * Verifier si (cond) (3)

* Tester la chose la + loin.

* si (cond) fausse : alors aller vers deb inst

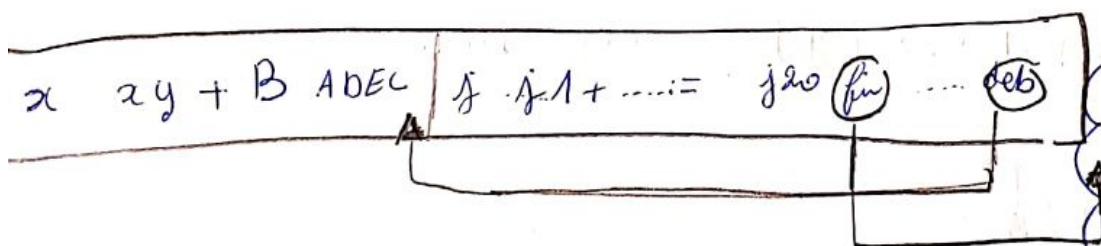
- fp(dec) = $x \quad x y + B \quad ADEC.$
 1^{ère} 2^{ème} ↑
 borne borne Non
 du
 tableau.

- fp(Inst) = $\boxed{j \quad j+1 \quad \dots} \quad x \quad x 1 + B \quad \text{SUBS} \quad 4 * i =$
 ↑
 indice

- fp(cond)

$j \geq 20 \rightarrow \text{fin} \quad B H Z \quad x 10 - \text{deb} \quad B H Z$

or:
on sort si
toutes les 2 sont
fausses



$BZ = 0$
 $BNZ = \neq 0$
 $BHZ = \leq 0$
 $BH = <$
 $BP = > 0$
 $BPZ = \geq 0$

Branchements
 Conditionnels
 (forme postfixée)

* Quaduplet:

- ① $(:=, x, , t_1)$
- ② $(+, x, y, t_2)$
- ③ $(\text{BOUNDS}, t_1, t_2,)$
- ④ $(\text{ADEC}, B, ,)$
- ⑤ $(+, j, 1, t_3)$
- ⑥ $(:=, t_3, , j)$
- ⑦ $(+, x, 1, t_4)$
- ⑧ $(*, B[t_4], 4, t_6)$
- ⑨ $(:=, t_6, , x)$
- ⑩ $(\text{BLE}, \text{fir}, j, 20)$
- ⑪ $(\text{BLE}, \text{deb}, x, 10)$

L'arbre abstrait:

