

a - grammaire syntaxique

$$\left\{ \begin{aligned} \langle \text{uit} - \text{adds} \rangle &\rightarrow \text{id} = \text{if} ((\text{add}(\text{list} - \text{end} - \text{exp}) \\ &\quad \text{ensembles; condition})) \\ \langle \text{list} - \text{end} - \text{exp} \rangle &\rightarrow (\langle \text{condi} \rangle, \langle \text{expi} \rangle) / (\langle \text{condi} \rangle, \langle \text{expi} \rangle) \\ &\quad (\langle \text{list} - \text{end} - \text{exp} \rangle) \end{aligned} \right.$$

b - Analyse descendante

$$\left\{ \begin{aligned} \langle \text{uit} - \text{adds} \rangle &\rightarrow \text{id}_1 = \langle A \rangle \text{if} ((\text{add}(\text{list} - \text{end} - \text{exp}) \\ &\quad \langle B \rangle \text{ensembles} \langle C \rangle, \text{condition} \langle D \rangle) \\ \langle \text{list} - \text{end} - \text{exp} \rangle &\rightarrow \langle \text{condi} \rangle, \langle \text{expi} \rangle \\ &\quad \langle \text{condi} \rangle, \langle \text{expi} \rangle, \langle \text{list} - \text{end} - \text{exp} \rangle \end{aligned} \right.$$

$\langle A \rangle, \langle B \rangle, \langle C \rangle, \langle D \rangle, \langle E \rangle, \langle F \rangle, \langle G \rangle \rightarrow \epsilon$

Code g n ral

1 - (=, 0, , id)

2 - (BR, des - condition, ,)

3 - { quadruplet de $\langle \text{condi} \rangle \rightarrow T.\text{cond1}$

4 - (BZ, des, , T.cond1)

5 - (+, id, T.exp1, id) $\rightarrow T.\text{exp1}$

6 - { quadruplet de $\langle \text{cond2} \rangle \rightarrow T.\text{cond2}$

7 - (BZ, des, , T.cond2)

8 - (+, id, T.exp2, id) $\rightarrow T.\text{exp2}$

9 - { quadruplet de $\langle \text{condn} \rangle \rightarrow T.\text{condn}$

10 - (BZ, fin, , T.condn)

11 - (+, id, T.expn, T.condn) $\rightarrow T.\text{expn}$

12 - (BR, fin, ,)

Routine C $\rightarrow E$

/ * récupération de la Constante
et mise à jour du BR. des Gndtr *
begin

Quod (QC) := (1 =, Constante, Gndtr)
QC++

Quod (Sav. BR. des Gndtr, 2) := QC
end;

0,5

Routine D $\rightarrow E$

begin / * Test de la Gndtr, Si ne
aller à des Gnd, Si
affecter Constante à cd *
end;

Quod (QC) := (BNZ, Sav. BR. Gndtr+2,)
QC++

Quod (QC) := (1 =, Constante, cd)
QC++

- Quod (Sav. BR. fin, 2) := QC

end; ~~Quod (Sav. BR. fin, 2) := QC~~

1

(3)

Routine G $\rightarrow E$

begin

Quod (QC) := (BNZ, fin, T. Gndtr)
Sav. BR. fin := QC
QC++

0,5

end;

Routine B $\rightarrow E$

begin / * demi exp * > |

Quod (QC) := (+ cd, T. exp, id)
QC++

Quod (QC) := (BR, fin, ls 02 types)
QC++

end; ~~Sav. BR. fin := QC~~
Sav. BR. fin := QC

~~B~~

1

foncteur $\langle E \rangle \rightarrow E$

begin /* test de condition, ferie la suite,
Somme si vraie */

Quand (cc) := (BZ, ^{cond}_{sm}, T. cond);
Som. cond. sur i = cc

end; cc++

~~Quand (cc) := (id~~

~~0.5~~

foncteur $\langle F \rangle \rightarrow E$

~~1~~

begin /* Dans le cas où cond est
vrai, ferie somme expr. *
if id.type \neq hyp. T. expr then errSe
else Quand (cc) := (+, id, T. expr, id)
cc++

Quand (~~Same~~ cond. sur, 2) := cc