UNIVERSITE DE NANCY FACULTE DES SCIENCES

CONSTRUCTION D'UN COMPILATEUR ALGOL

POUR I. B. M. 1620

-=-=-=-=-=-

THESE

pour l'obtention du

DOCTORAT de SPECIALITE MATHEMATIQUES (3ème CYCLE)

Soutenue devant le Jury le 27 octobre 1964

par

Michel CUSEY

-=-=-=-=-

Jury : Mr J. LEGRAS Président
Mr C. PAIR Examinateurs
Melle D. HUET

Compilateur algol
Algol , compilateur

UNIVERSITE DE NANCY — FACULTE DES SCIENCES

CONSTRUCTION D'UN COMPILATEUR ALGOL

POUR I. B. M. 1620

-=-=-=-=-

par

Michel CUSEY

UNIVERSITE DE NANCY - FACULTE DES SCIENCES

Doyen : M. AUBRY

Assesseur : M. GAY

Doyens honoraires : MM. CORNUBERT - DELSARTE - URION - ROUBAULT -

Professeurs honoraires : MM. CROZE - RAYBAUD - LAFFITE - LERAY - Y- LAPORTE - EICHHORN - CAPELLE - GODEMENT - DUBREIL - L. SCHWARTZ - UDONNE - De MALLEMANN - LONGCHAMBON - LETORT - DODE - GAUTHIER - JDET - OLMER - CORNUBERT - CHAPELLE - GUERIN - CHEVALLIER - WAHL - RVE -

Maîtres de conférences honoraires : MM. LIENHART - PIERRET -

PROFESSEURS

ON	Chimie biologique	SCHWARTZ	Exploit. minière
LSARTE	Analyse supérieure	GAYET	Physiologie
JBAULT	Géologie	MANGENOT	Phytopathologie
LLET	Biologie animale	MALAPRADE	Chimie
HEVIN	Botanique	HADNI	Physique
RRIOL	Chimie théorique	BONVALET	Mécanique physique
ETTE	Physique	KERN	Minéralogie
LLIEN	Electronique	BASTICK	Chimie
ERT	Chimie physique	DUCHAUFOUR	Pédologie
GRAS	Mécanique rationnelle	NEEL	Chimie ind. orga.
LFA	Minéralogie	GARNIER	Agronomie
LAUSE	Chimie	WEPPE	Minéralogie appli.
VRE	Physique appliquée	BERNARD	Géologie appliquée
BRY	Chimie minérale	CHAMPIER	Physique
VAL	Chimie	REGNIER	Physico-chimie
PPENS	Radiogéologie	GAY	Chimie biologique
HLING	Physique	WERNER	Botanique
NER	Physique expérimentale	CONDE	Zoologie
LY	Géologie	STEPHAN	Zoologie
GOFF	Génie chimique	EYMARD	Cal. Dif. et Int.
PON	Chimie biologique	LEVISALLES	Chimie organique
OLD	Chimie industrielle	N.	Physique
		N.	M. M. P.

MAITRES DE CONFERENCES ET PROFESSEURS SANS CHAIRE

GE	Génie chimique	Mme HERVE	Math. (propédeutique)
CI	Géologie	AUROUZE	Géologie
LAUME	Psychophysiologie	MARI	Chimie I. S. I. N.
N	Mécanique physique	LAFON	Physique I. S. I. N.
BASTICK	Chimie M. P. C. Epinal	FELDEN	Phy. Théor. & Nucl.
EFIN	Physique	FLECHON	M. P.C.
N	Physique	VIGNES	Physique (Mines)
NTZ	Biologie animale	Melle HUET	Math. (S. P. C. N.)
		JANOT	M. P. C. Epinal

SECRETAIRE PRINCIPAL : C. CARON

Je tiens à exprimer ma profonde gratitude à Monsieur le Professeur J. LEGRAS, Directeur du Centre de Calcul, pour l'intérêt qu'il n'a cessé de me temoigner durant ces deux années, et à l'assurer de mon entier dévouement.

J'adresse mes plus vifs remerciements à Monsieur C. PAIR pour l'attention continuelle, l'intérêt constant et l'amicale sollicitude avec lesquels il a dirigé et suivi ce travail.

Je remercie Mademoiselle HUET qui m'honore de sa présence dans le jury.

Je remercie Monsieur PROCYK, Ingénieur I B M, pour la complaisance dont il a toujours fait preuve à mon égard.

Je remercie mon camarade A. FLOC'H pour son appréciable collaboration.

Je remercie enfin Mesdames les Secrétaires du Centre de Calcul pour les nombreux services qu'elles m'ont aimablement rendus.

SOMMAIRE

5. Instruction POUR.
6. Instruction d'affectation.
7. Instruction ALLERA.
8. Instruction - Instruction composée - Bloc.
9. Déclaration de type.
10. Indicateur de fonction.
11. Déclaration et indicateur d'aiguillage.
12. Déclaration de tableau et Variable indicée.
13. Déclaration de procédure.

Annexe : PROGRAMME.

MESSAGES D'ERREURS.

Références .

-=-=-=-=-=-

CHAPITRE I

RESULTATS PRELIMINAIRES

I - SYNTAXE

1. Rappels rapides (d'après [3])

Soit L un langage "context-free" [1] construit sur un vocabulaire terminal T, décrit grâce à un vocabulaire non terminal N. On appelle proposition de L toute suite π de signes terminaux qui dérive d'un élément D_0 quelconque de N ; π est obtenu grâce à une suite $D = (D_0, D_1, D_2, \ldots, D_n)$ où $D_n = \pi$, et pour $0 < i \leq n$, D_i est une réécriture de D_{i-1} : on nomme cette suite une **construction formative de π**. Il lui est associé un arbre de décomposition syntaxique, dont chaque noeud est "valué" par un signe du vocabulaire V = N U T. Plus précisément, toute feuille est valuée par le signe terminal dont elle est occurrence dans π ; un noeud qui n'est pas une feuille, et qu'on appelle un noeud propre, est valué par le signe non terminal dont il est occurrence dans la construction formative. A tout noeud propre on peut associer l'intervalle de π qui en dérive, qui est une proposition, et qu'on appelle sous-proposition de π. L'ensemble des sous-propositions de π, pour la relation d'inclusion, est un arbre. On achéve de définir L en fixant un axiome dans l'ensemble N : les phrases de L sont les propositions qui dérivent de cet axiome. Les sous-propositions d'une phrase seront nommées sous-phrases (bien qu'elles ne soient pas des phrases).

Nous utiliserons par la suite, pour écrire les régles du langage, la forme normale de BACKUS [6]

D'après ce qui précéde, toute proposition α peut être construite par une dérivation $(A, \mu, \ldots, \alpha)$ où μ est une chaîne non réduite d'un signe non terminal :

$A ::= \mu$ sera appelée <u>régle origine</u> de la proposition.

Chaque non terminal de la chaine μ a pour dérivation terminale une sous-proposition de α, qu'on appelle une <u>composante</u> de α ;

2. Construction de l'arbre des sous-propositions dans un cas particulier

Nous étudierons le cas où toutes les régles du langage sont du type :

$C ::= C'$

ou $C ::= B_0\ U_1\ B_1 \ldots U_n\ B_n$

avec $C, C' \in N$; $n \geqslant 1$; $U_i \in T$; $B_j \in N$ ou vide.

Monsieur FLOYD [2] définit alors dans le vocabulaire terminal trois relations et donne un algorithme d'analyse syntaxique dans le cas où elles sont incompatibles deux à deux.

Sur le même sujet, Monsieur PAIR [3] donne un algorithme qui permet de construire l'arbre des sous-propositions d'une proposition donnée, grâce à une pile :

La proposition étudiée étant encadrée de deux signes Δ et Φ, les signes terminaux qui la composent entrent dans la pile dans leur ordre d'écriture. Les signes terminaux et les propositions qui sont successeurs dans l'arbre d'une même sous-proposition sortent ensemble de la pile. Chacune de ces sorties détermine ainsi une nouvelle sous-proposition, qui entre aussitôt dans la pile.

Le test d'entrée d'un signe terminal dans la pile se fait grâce à trois relations définies dans V.

<u>Définitions</u> :

1) toute phrase π est construite grâce à une construction formative. Celle-ci peut être effectuée en réécrivant à chaque fois le premier (resp. dernier) signe non terminal ; elle est alors dite <u>normale gauche</u> (resp. <u>droite</u>).

2) $y \in T$ est <u>descendant gauche</u> (resp. <u>droit</u>) de $B \in N$ s'il existe une construction formative normale gauche (resp. droite) $(B, D_1, \ldots D_n)$ et un entier positif i, tel que la première (resp. dernière) occurrence terminale de D_i soit une occurrence de y.

3) Définition de trois relations dans l'ensemble T :

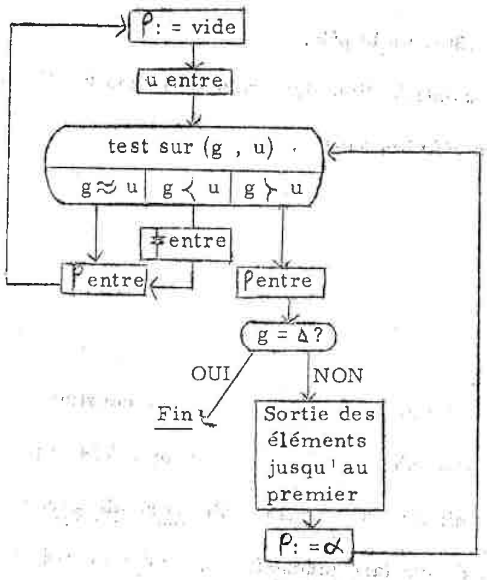
$x \approx y \Longleftrightarrow \exists$ une régle $A ::= \lambda x\ C\ y \mu$

$x \lessdot y \Longleftrightarrow \exists$ une régle $A ::= \lambda x\ B \mu$ telle que y soit descendant gauche de B

$x \gtrdot y \Longleftrightarrow \exists$ une régle $A ::= \lambda B\ y \mu$ telle que x soit descendant droit de B.

Notations : $x \in T$; $y \in T$; $A \in N$; $B \in N$; $C \in N$ ou C vide ; λ et μ sont deux chaînes (éventuellement vides) d'éléments de V.

organigramme :



Notations :

A chaque étape :

g est le premier signe terminal se trouvant au sommet de la pile.

u est le premier signe terminal susceptible d'entrer.

Δ et Φ sont les deux éléments adjoints à T pour encadrer toute phrase et on convient :

$(\forall x \in T)\ (\Delta \lessdot x$ et $x \gtrdot \Phi$ et $\Delta \gtrdot \Phi)$.

ǂ est un séparateur indiquant jusqu'où vider la pile pour avoir les éléments d'une sous-proposition.

ρ est une mémoire auxiliaire où on place α en attendant de savoir s'il doit être précédé d'un séparateur ǂ, c'est-à-dire s'il est la premiére composante d'une autre sous-proposition.

α est la proposition formée avec les sous-propositions et les signes terminaux qui sortent (Remarquons qu'ils sortent dans l'ordre inverse de l'ordre d'écriture).

Exemple Algol : Construction de l'arbre des sous-propositions de la proposition

$\Delta\ a \uparrow 3 \times b\ \Phi$

Etablissons d'abord les priorités que nous allons utiliser ici :

① Comparaison des signes a et $\uparrow$:

Nous avons :

$\langle \text{facteur} \rangle ::= \langle \text{facteur} \rangle \uparrow \langle \text{primaire arith.} \rangle \mid \langle \text{primaire arith.} \rangle$

La régle origine de $a \uparrow 3$ est du type $A ::= \lambda B\ y \mu$, où :

λ vide

$B = \langle \text{facteur} \rangle$

$y = \uparrow$

$\mu = \langle \text{primaire arith.} \rangle$

Par définition de la relation $\gtrdot$, on sait qu'alors :

$\left(\begin{matrix}\text{descendant} \\ \text{droit de B}\end{matrix}\right) \gtrdot y$

réécrivons B , , c'est-à-dire ⟨facteur⟩ :

⟨facteur⟩ :: = ⟨primaire arith.⟩

⟨primaire arith.⟩ :: = ⟨variable⟩

⟨variable⟩ :: = ⟨variable simple⟩

⟨variable simple⟩ :: = ⟨identificateur de variable⟩

⟨identificateur de variable⟩ :: = ⟨identificateur⟩

⟨identificateur⟩ :: = ⟨lettre⟩

⟨lettre⟩ :: = a

Alors :

$\boxed{a \rangle \uparrow}$

② Comparaison des signes ↑ et 3 :

Partons de la même régle origine considérée cette fois comme étant du type A :: = λ x B μ où | λ = ⟨facteur⟩ ; x = ↑ ; B = ⟨primaire arith.⟩ ; μ = vide

Réécrivons B = ⟨primaire arith.⟩

⟨primaire arith.⟩ :: = ⟨nombre sans signe⟩

⟨nombre sans signe⟩ :: = ⟨nombre décimal⟩

⟨nombre décimal⟩ :: = ⟨entier sans signe⟩

⟨entier sans signe⟩ :: = ⟨chiffre⟩

⟨chiffre⟩ :: = 3

Alors, par définition de la relation ⟨ :

$\boxed{\uparrow \langle 3}$

③ Comparaison des signes ↑ et X ainsi que 3 et X :

Partons de la régle

⟨terme⟩ :: = ⟨terme⟩ X ⟨facteur⟩

c'est une régle du type A :: = λ B y μ où B = ⟨terme⟩ ; nous réécrivons :

⟨terme⟩ :: = ⟨facteur⟩

⟨facteur⟩ :: = ⟨facteur⟩ ↑ ⟨primaire arith.⟩

↑ est descendant droit de ⟨terme⟩ , donc :

$\boxed{\uparrow \rangle X}$

Cherchons le descendant droit suivant, c'est-à-dire réécrivons ⟨primaire arith.⟩ :

⟨primaire arith.⟩ :: = ⟨nombre sans signe⟩

⟨nombre sans signe⟩ :: = ⟨nombre décimal⟩

⟨nombre décimal⟩ :: = ⟨entier sans signe⟩

⟨entier sans signe⟩ :: = ⟨chiffre⟩

⟨chiffre⟩ :: = 3

3 est descendant droit ⟨terme⟩ , donc :

$\boxed{3 \rangle X}$

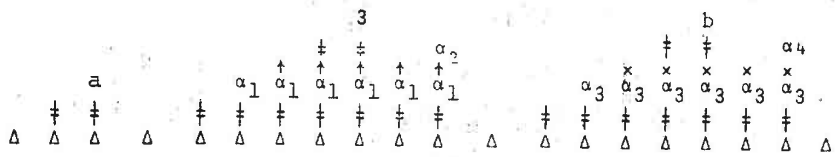
④ Comparaison des signes X et b

En partant de la même régle que dans ③ et en réécrivant ⟨facteur⟩ comme dans ① on trouve facilement : b est descendant gauche de ⟨facteur⟩ , donc :

$\boxed{X \langle b}$

Nous pouvons alors appliquer l'organigramme :

Les différentes étapes de la pile sont :



Notations : $\alpha_1 = a$, sous-phrase composée de a seul

$\alpha_2 = 3$

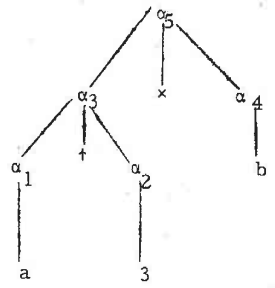
$\alpha_3 = \alpha_1 \uparrow \alpha_2$

$\alpha_4 = b$

$\alpha_5 = \alpha_3 \times \alpha_4$

ordre de sortie de la pile : $a \ddagger 3 \ddagger \alpha_2 \uparrow \alpha_1 \ddagger b \ddagger \alpha_4 \times \alpha_3 \ddagger$

arbre dessous-propositions et signes terminaux



3. Transformations de l'algorithme :

1) On se raménera à des régles du type :

(1)	A :: = A'	avec	A, A' $\in$ N
ou (2)	A :: = B x C		B, C $\in$ N ou vide
ou (3)	A :: = B y C v		x, y, v $\in$ T

Alors $g \approx u$ se présente seulement dans le cas (3) (g = y et u = v) : v entre et sort aussitôt en même temps que y ; on ne le fera pas entrer, c'est-à-dire que y devra alors sortir et v être supprimé de la chaîne d'entrée ; pour cela il faudra convenir $y \succ v$, ce qui entraîne une nouvelle définition de $\succ$:

$y \succ x$, si et seulement si il existe :

- soit une régle du type A :: = B x C où y est descendant droit de B (en ce cas y sort)

- soit une régle du type A :: = B y C x (en ce cas y sort et x est supprimé de la chaîne d'entrée)

2) Le seul rôle de $\ddagger$ est de savoir combien d'éléments doivent sortir de la pile pour construire une sous-phrase α. Cela dépend de la règle originaire A :: = μ. Le nombre d'éléments à sortir est le nombre d'éléments de μ. Mais par divers tests, nous déterminerons μ, de sorte que $\ddagger$ est inutile. Alors la mémoire intermédiaire ρ est aussi inutile. De plus, μ étant connu, donc l'ordre relatif des sous-propositions et terminaux à sortir étant connu, il n'y a pas d'inconvénients à séparer la pile en deux :

- pile 0 des signes terminaux

- pile V des sous-propositions

Notation : on appellera S (P) le sommet de la pile P. Ici g = S (0)

3) Enfin, l'organigramme précédent était fait pour l'analyse d'une phrase dont on était sûr qu'elle appartenait au langage.

Une condition nécessaire pour cela est que pour tout couple (g,u) soit vérifiée l'une des relations

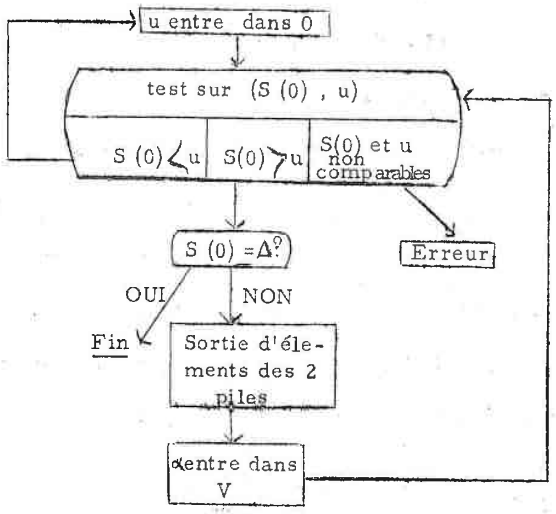
$g \lessdot u$

ou $g \gtrdot u$

ou $g \approx u$ (supprimée ici)

Si donc pour un programme donné à la compilation, aucune de ces relations n'est vérifiée, le programme est syntaxiquement incorrect.

D' où le nouvel organigramme :



Exemple Algol :

Reprenons la proposition

Δ a ↑3 x b ϕ

Les notations adoptées pour les sous-propositions sont les mêmes que dans le § 2

Les différentes étapes des piles sont :

		a		↑	3 ↑	↑		×	b ×	×	
pile 0 →	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ	Δ
						α_2			α_4		
pile V →			α_1	α_1	α_1	α_1	α_3	α_3	α_3	α_5	

II SEMANTIQUE

Problème de la compilation :

A toute phrase π du langage source associer une phrase t (π) du langage objet ayant même "valeur sémantique". On résoudra un problème plus large : à toute phrase π du langage source associer une phrase t(π) du langage objet et à toute sous-phrase λ de π une sous-phrase t (λ) de t (π) de manière que pour toute sous-phrase λ de π (en particulier pour $\lambda = \pi$), λ et t (λ) aient même valeur sémantique.

Il faudrait donner une définition de la valeur sémantique des sous-phrases d'un langage de programmation. Monsieur C. PAIR [4] a donné une telle définition et montre que lorsque le langage source et le langage objet

vérifient certaines hypothèses, si $\lambda_1, \lambda_2, \ldots, \lambda_n$ sont les composantes d'une sous-phrase λ :

$$t(\lambda) = \Gamma_0 t(\lambda_1) \Gamma_1 t(\lambda_2) \ldots \Gamma_{n-1} t(\lambda_n) \Gamma_n$$

(concaténation)

chaque Γ_i étant une suite d'ordres du langage objet en étant vide ; et les Γ_i dépendant de la règle origine λ et d'un certain nombre de paramètres.

Dans le présent travail, on n'entrera pas dans l'étude des hypothèses faites. On prendra au contraire ce résultat pour hypothèse de travail.

Plus précisement, toute sous-phrase α a ici au plus deux composantes β et γ ; la régle origine de α peut être mise sous la forme A :: = B x C v si on accepte que B, C ou v puissent être absents ; et $\alpha = \beta$ x γ v , où β, γ peuvent être vides et v absent.

On cherchera $t(\alpha)$ sous la forme :

$$t(\alpha) = t(\beta)\, \Gamma'\, t(\gamma)\, \Gamma$$

où Γ et Γ' sont des suites d'ordres (éventuellement vides) du langage machine 1620, dépendant de :

1) la règle origine

2) au plus trois adresses $m(\beta)$, $m(\gamma)$, $m(\alpha)$ associées aux sous-phrases β , γ , α : par exemple si β est un identificateur , γ et α des expressions, $m(\beta)$ sera l'adresse de la zône associée à cet identificateur β pour y ranger sa valeur, $m(\gamma)$ et $m(\alpha)$ seront les adresses où seront rangées les résultats intermédiaires que sont les valeurs des expressions γ et α.

3) au plus deux "indicatifs de type " $i(\beta)$ et $i(\gamma)$.

La régle, $i(\beta)$ et $i(\gamma)$ déterminent $i(\alpha)$.

Autrement dit, on aura à associer à chaque règle une ou deux fonctions d'au plus cinq variables (adresses et indicatifs), dont les valeurs sont des suites d'ordres Γ et Γ' du langage objet.

Lorsque l'analyse syntaxique aura déterminé une nouvelle sous-phrase, on devra connaître :

1) Sa règle origine :

on y parviendra grâce au couple (S (0), u) et aux sous-phrases composantes.

2) $m(\beta)$, $m(\gamma)$, $i(\beta)$, $i(\gamma)$:

Au lieu de ranger des sous-phrases dans la pile V , on y rangera ces renseignements ; c'est-à-dire que pour chaque sous-phrase δ on mettra dans la pile V l'adresse $m(\delta)$ et l'indicatif $i(\delta)$. On connaitra ainsi $m(\beta)$, $m(\gamma)$, $i(\beta)$ et $i(\gamma)$.

3) $m(\alpha)$:

On choisira une mémoire $m(\alpha)$ disponible (cf. chapitre III)

En réalité, Γ ne dépendra que de $m(\beta)$ et $i(\beta)$.

Montrons qu'on peut alors construire $t(\pi)$ par concaténations successives au fur et à mesure de l'analyse syntaxique, plus précisément :

- en enchaînant Γ' à la partie déjà construite lorsque x entre dans 0.
- en enchaînant Γ à la partie déjà construite lorsque x, β et γ sortent des piles (autrement dit à l'entrée de α dans V).

Pour cela montrons par récurrence sur l'ordre d'entrée des sous-phrases dans V que t (α) est construit entre l'entrée dans 0 du premier signe (terminal) de la sous-phrase α , et l'entrée dans V de α ; c'est-à-dire qu si à l'entrée du premier signe de α , la partie construite du programme objet est p, à l'entrée de α elle est p suivie de t (α).

Supposons que cela soit vrai jusqu'à α exclue : c'est vrai pour β et γ qui entren dans V avant α. Donc, si à l'entrée du premier signe de α, qui est aussi le premier signe de β, le programme construit est p, à l'entrée dans V de β , c'est-à-dire juste avant l'entrée de x dans 0, il est p t (β).

Après l'entrée de x, il est par hypothèse p t (β) Γ'

Puis entre le premier signe de γ ; lorsque γ entre dans V, le programme objet écrit est p t (β) Γ' t (γ).

Alors β , x et γ sortent des piles, le programme devient :

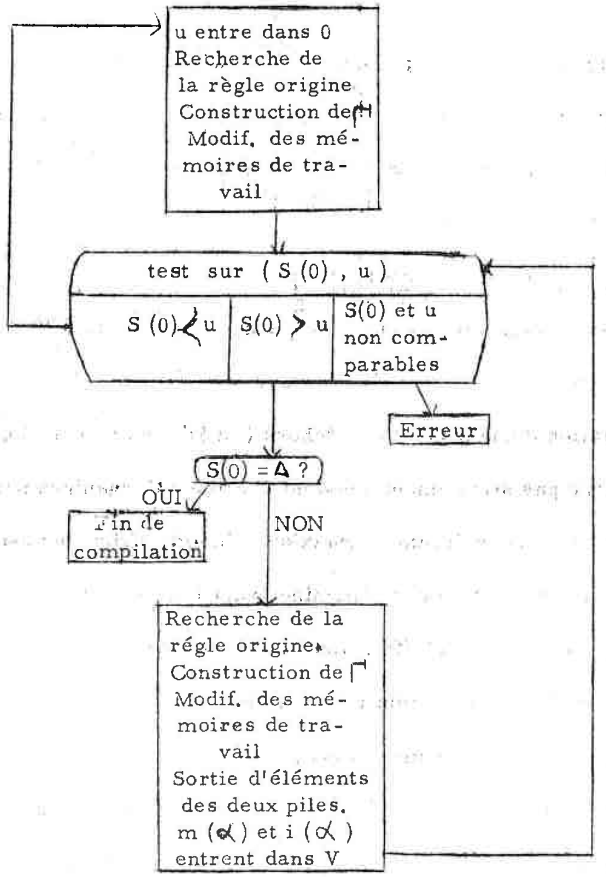
$$p\ t(\beta)\Gamma' t(\gamma)\Gamma = p\ t(\alpha)$$

et α entre dans V.

Donc on obtient bien ainsi t (Π) par concaténations successives comme il a été dit, entre l'entrée du premier signe de Π, et la fin du déroulement de l'algorithme.

Certaines mémoires de travail seront utiles pendant la compilation en dehors des piles 0 et V ; par exemple un compteur de première mémoire disponible pour résultats intermédiaires. On ne modifiera ces mémoires, comme les piles 0 et V, que lors de l'entrée ou de la sortie d'éléments de 0 et V, c'est-à-dire lors de la construction des Γ et Γ'.

D'où l'organigramme :



Il apparaît donc qu'à chaque règle origine sont associés Γ et Γ' et deux suites d'ordres Λ et Λ' permettant la construction de Γ et Γ' et les modifications correspondantes des mémoires de travail.

Rôle des déclarations :

En réalité Algol n'est pas un langage "context-free". Il contient des déclarations : si une sous-phrase α est une déclaration, elle n'a pas nécessai-

rement une traduction t (α) ; son rôle est plutôt de préciser la grammaire utilisée pour construire une partie de la phrase étudiée. Les identificateurs déclarés seront mis dans une table des symboles et on leur affectera une mémoire.

Traitement des identificateurs :

Lorsqu'un identificateur α est reconnu par l'analyse syntaxique, il est donc traité différement selon qu'il se trouve dans sa déclaration (entrée dans table, affectation d'une zône) ou en dehors (m (α) entre dans V).

Il n'est donc pas strictement possible d'utiliser l'organigramme précédent. D'autre part, la reconnaissance syntaxique d'un identificateur est très simple : il commence par une lettre et se termine avant le premier signe suivant qui n'est pas une lettre ou un chiffre. Enfin, les sous-phrases strictement contenues dans un identificateur n'ont pas de valeur sémantique et ne créent donc pas d'ordres dans le programme objet .

La reconnaissance des identificateurs sera donc faite par un sous-programme spécial. On reconnaitra si on est ou non dans une déclaration en testant si S (0) est ou non un déclarateur :

- S (0) est un déclarateur :

On met l'identificateur dans la table et on envoie sur la pile l'adresse où on veut stocker dans la table l'adresse de la zône que l'on affectera à l'identificateur. Cette affectation et le stockage de cette adresse se feront dans la partie A correspondant à la sortie de la sous-phrase "déclaration" en cours.

- S (0) n'est pas un déclarateur :

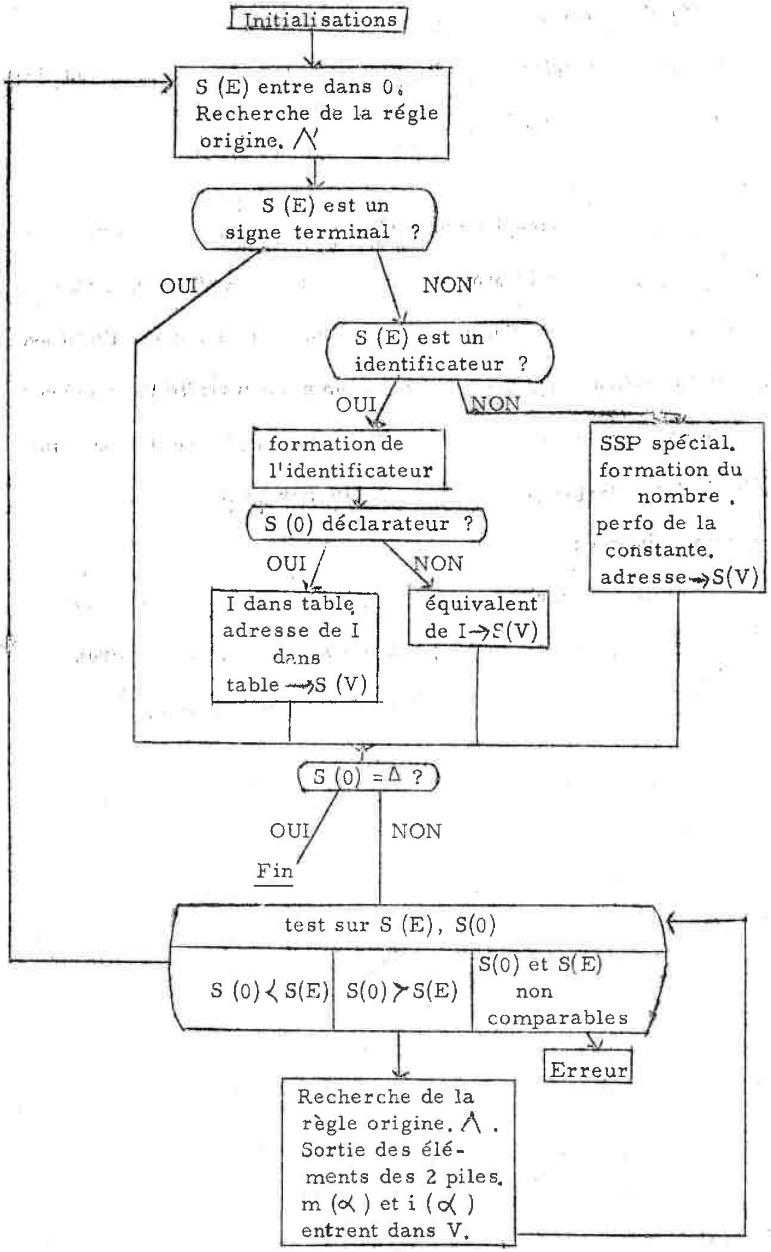
On verra au chapitre III qu'on s'est arrangé pour que l'identificateur soit alors déjà dans la table.

Traitement des nombres :

De même, les sous-phrases strictement contenues dans un nombre ne créent pas d'ordre pour le programme objet : la compilation du nombre produira une constante (dont l'adresse jouera un rôle analogue à l'adresse affectée à un identificateur). C'est pourquoi on a aussi traité les nombres par un sous-programme spécial qui met l'adresse m (α) de la constante sur la pile V. Ce sous-programme est détaillé par [5]

Organigramme général :

On peut considérer la suite des signes terminaux à entrer comme une pile E qui ne ferait que décroître. Pour cette raison nous noterons S (E) le premier élément susceptible d'entrer à chaque instant.



CHAPITRE II

MATRICE DE PRIORITE

Transformation de la grammaire d'Algol :

Pour rester dans le cadre tracé au chapitre précédent, nous sommes amenés à modifier la grammaire d'Algol exposée dans [7] (traduction française [8]). En effet, les régles doivent être du type.

	A :: = A' (1)	avec	A, A' ∈ N
ou	A :: = B x C (2)		B, C ∈ N ou vide
ou	A :: = B y C v (3)		x, y, v ∈ T

Certaines régles de [7] ont été modifiées en ce sens.

Par exemple,

nous avons au paragraphe 3. 3. 1 de [7] :

Régle 7 : ⟨proposition si⟩ ::= SI ⟨Exp. Bool.⟩ ALORS

et Régle 8 : ⟨exp. Arith.⟩ ::= ⟨prop. si⟩ ⟨Exp. Arith. simple⟩ ⟨Exp. Arith.⟩

Cette dernière régle n'est pas d'un type étudié. Nous pouvons nous y ramener en écrivant :

① ⟨prop. Si⟩ ::= SI ⟨Exp. Bool.⟩

② ⟨test Arith.⟩ ::= ⟨prop. Si⟩ ALORS ⟨Exp. Arith. Simple⟩

③ ⟨Exp. Arith.⟩ ::= ⟨test Arith.⟩ SINON ⟨Exp. Arith.⟩

①, ② et ③ sont du type (2).

Toutefois, remarquons dès maintenant que ces nouvelles règles conduis
à deux relations de priorité qui sont compatibles.

En effet, par définition de ces relations,

① et ③ ⟹ SI ⋖ ALORS

① et ② ⟹ SI ⋗ ALORS

On peut éviter cette ambiguité en coupant ALORS en deux symboles AL
de manière que SI et AL jouent le rôle de parenthèses ;

nous écrivons les régles :

⟨prop. Si⟩ ::= SI ⟨Exp. Bool.⟩ AL

⟨test Arith.⟩ ::= ⟨prop. Si⟩ ORS ⟨Exp. Arith. Simple⟩

⟨Exp. Arith.⟩ ::= ⟨test Arith.⟩ SINON ⟨Exp. Arith.⟩

Nous verrons au § 5 qu'en pratique, nous éviterons de scinder ALORS
deux nouveaux symboles.

Pour l'énoncé des régles, nous suivrons l'ordre de [7]. Dans
colonne de droite on affecte un numéro aux régles du type (2) ou (3) : ce
numéro sera utilisé au § 2. Les régles du type (1) non modifiées n'ont p
été reproduites ; elles sont repérées par leur rang dans la paragraphe d
[7] qui les contient.

Paragraphe du Raport Algol	Régles	Numéros
3. 1. 1. (variables)	Règles 1, 2, 3.	
	⟨liste d'indices⟩ ::= ⟨Exp. en indice⟩ | ⟨liste d'indices⟩ , ⟨Exp. en indice⟩	1
	Règle 4.	
	⟨Variable indicée⟩ ::= ⟨ident. de tableau⟩ [⟨liste d'indices⟩]	2
	Règle 5.	
3. 2. 1 (Indicateurs de fonction)	Règles 1, 2.	
	⟨chaîne de lettres⟩ ::= ⟨lettre⟩ | ⟨chaîne de lettres⟩ {A … Z}	
	⟨délimiteur de paramètres⟩ ::= , |) | ⟨chaine de lettres⟩ :((sera remplacé par , dès son identification)	
	⟨liste de paramétres effectifs⟩ ::= paramètre effectif ⟩ |	
	⟨liste de paramètres effectifs⟩ ⟩ ⟨paramètre effectif⟩	3
	⟨indicateur de fonction⟩ ::= ⟨ident. de procédure⟩ |	
	⟨ident. de procédure⟩ (⟨liste de paramètres effectifs⟩)	4
3. 3. 1 (expressions arith.)	Règles 1, 2.	
	⟨primaire arith.⟩ ::= ⟨nombre sans signe⟩ | ⟨variable⟩ |	
	⟨indicateur de fonction⟩ | (⟨exp. arith.⟩)	5

	⟨facteur⟩ ::= ⟨primaire arith.⟩ | ⟨facteur⟩ ↑ ⟨primaire arith.⟩	6
	⟨terme⟩ ::= ⟨facteur⟩ | ⟨terme⟩ X ⟨facteur⟩ |	7
	⟨terme⟩ / ⟨facteur⟩ |	8
	⟨terme⟩ $\frac{o}{o}$ ⟨facteur⟩	9
	⟨exp. arith. simple⟩ ::= ⟨terme⟩ | + ⟨terme⟩ |	10
	- ⟨terme⟩ |	11
	⟨exp. arith. simple⟩ + ⟨terme⟩ |	12
	⟨exp. arith. simple⟩ - ⟨terme⟩	13
	⟨prop. Si⟩ ::= SI ⟨exp. Bool.⟩ AL	14
	⟨test arith.⟩ ::= ⟨prop. Si⟩ ORS ⟨Exp. arith. simple⟩	15
	⟨exp. arith.⟩ ::= ⟨test arith.⟩ SINON ⟨Exp. arith.⟩	16
3. 4. 1. (expressions Boolèenne)	Règle 1	
	⟨relation⟩ ::= ⟨E. A. S.⟩ < ⟨E. A. S.⟩ |	17
	⟨E. A. S.⟩ ⩽ ⟨E. A. S.⟩ |	18
	⟨E. A. S.⟩ = ⟨E. A. S.⟩ |	19
	⟨E. A. S.⟩ ⩾ ⟨E. A. S.⟩ |	20
	⟨E. A. S.⟩ > ⟨E. A. S.⟩ |	21
	⟨E. A. S.⟩ ≠ ⟨E. A. S.⟩	22
	⟨Primaire booléen⟩ ::= ⟨valeur logique⟩ | ⟨variable⟩ |	
	⟨indicateur de fonction⟩ | ⟨relation⟩	
	(⟨exp. booléenne⟩)	23

	⟨Secondaire booléen⟩ ::= ⟨primaire booléen⟩ | ¬ ⟨primaire booléen⟩	24
	⟨Facteur booléen⟩ ::= ⟨secondaire booléen⟩ | ⟨facteur booléen⟩ ∧ ⟨secondaire booléen⟩	25
	⟨Terme booléen⟩ ::= ⟨facteur booléen⟩ | ⟨terme booléen⟩ ∨ ⟨facteur booléen⟩	26
	⟨Implication⟩ ::= ⟨terme booléen⟩ | ⟨Implication⟩ ⊃ ⟨terme booléen⟩	27
	⟨Exp. booléenne simple⟩ ::= ⟨Implication⟩ | ⟨Exp. bool. simple⟩ ≡ ⟨Implication⟩	28
	⟨test booléen⟩ ::= ⟨Prop. Si⟩ ORS ⟨Exp. bool. simple⟩	29
	⟨Exp. booléenne⟩ ::= ⟨test booléen⟩ SINON ⟨Exp. bool.⟩	30
3. 5. 1 (expression de désignation)	Régles 1, 2,	
	⟨Indicateur d'aiguillage⟩ ::= ⟨identificateur d'aiguillage⟩ [⟨exp. en indice⟩]	31
	⟨Exp. de désignation simple⟩ ::= ⟨étiquette⟩ | ⟨indic. d'aiguillage⟩ | (⟨exp. de désignation⟩)	32
	⟨test de désignation⟩ ::= ⟨prop. Si⟩ ORS ⟨exp. de désignation simple⟩	33

4. 1. 1. Instructions composées et blocs

⟨exp. de désignation⟩ ::= ⟨test de désignation⟩ SINON ⟨exp. de désignation⟩ 34

Règle 1.

⟨Inst. de base⟩ ::= ⟨Inst. non étiquetée de base⟩ |
⟨étiquette⟩ : ⟨Inst. de base⟩ |
⟨étiquette⟩ :

Nous verrons au chapitre III que les étiquettes subissent un traitement spécial ; c'est pourquoi nous ne nous soucions pas des priorités de (:) suivant ⟨étiquette⟩

Règles 3, 4

⟨partie d'inst. composée⟩ ::= ⟨instruction⟩ |
⟨partie d'instruction compo.⟩ ; ⟨inst.⟩ 35

⟨partie déclaration⟩ ::= ⟨déclaration⟩ |
⟨partie déclaration⟩ ; ⟨déclaration⟩ 36

⟨Inst. composée non étiquetée⟩ ::=
DEBUT ⟨partie d'inst. composée⟩ FIN | 37
DEBUT FIN 38

⟨bloc ouvert⟩ ::= ⟨partie déclaration⟩ ; | 39
⟨partie déclaration⟩ ; ⟨partie d'inst. composée⟩ 40

⟨bloc non étiqueté⟩ ::= DEBUT ⟨bloc ouvert⟩ FIN 41

Règles 9, 10, 11.

4. 2. 1. Instructions d'affectation

⟨partie gauche⟩ ::= ⟨variable⟩ | ⟨ident. de procédure⟩

⟨partie droite⟩ ::= ⟨Exp. arith.⟩ | ⟨exp. booléenne⟩ |
⟨Inst. d'affectation⟩

⟨Inst. d'affectation⟩ ::= ⟨partie gauche⟩ : = ⟨partie droite⟩ 42

4. 3. 1. Instruction aller à

⟨Instruction aller a⟩ ::= ALLER A ⟨exp. de désignation⟩ 43

4. 4. 1 Instruction vide

Règle supprimée parce qu'incompatible avec la définition des langages du type 2 ("context-free") de [1.]
En revanche, on ajoute des règles à 4. 1. 1. , 4. 5. 1. et 4. 6. 1.

4. 5. 1. Instruction conditionnelle

⟨Inst. Si⟩ ::= ⟨prop. Si⟩ ORS ⟨inst. inconditionnelle⟩ 44
⟨prop. Si⟩ ORS 45

⟨Inst. conditionnelle⟩ ::= ⟨Inst. Si⟩ |
⟨Inst. Si⟩ SINON ⟨Instruction⟩ 46
⟨Inst. Si⟩ SINON | 47
⟨prop. Si⟩ ORS ⟨inst. pour⟩ 48
⟨étiquette⟩ : ⟨Inst. conditionnelle⟩
(cf 4. 1. 1.)

4. 6. 1. Instruction pour

⟨ progression ⟩ ::= ⟨ Exp. arith. ⟩ PAS ⟨ Exp. arith. ⟩ 49

⟨ élément d'une liste de Pour de type 1 ⟩ ::= ⟨ Exp. arith. ⟩

⟨ élément d'une liste de Pour de type 2 ⟩ ::= ⟨ progression ⟩ JUSQUA ⟨ Exp. arith. ⟩ 50

⟨ élément d'une liste de Pour de type 3 ⟩ ::= ⟨ Exp. arith. ⟩ TANTQUE ⟨ Exp. booléenne ⟩ 51

⟨ liste de Pour de type 1 ⟩ ::= ⟨ élément d'une liste de Pour de type 1 ⟩ |

⟨ élément d'une liste de Pour de type 1 ⟩ , ⟨ liste de Pour de type 1 ⟩ | 52

⟨ élément d'une liste de Pour de type 2 ⟩ , ⟨ liste de Pour de type 1 ⟩ | 53

⟨ élément d'une liste de Pour de type 3 ⟩ , ⟨ liste de Pour de type 1 ⟩ 54

⟨ liste de Pour de type 2 ⟩ ::= ⟨ élément d'une liste de Pour de type 2 ⟩ |

⟨ élément d'une liste de Pour de type 1 ⟩ , ⟨ liste de Pour de type 2 ⟩ | 55

⟨ élément d'une liste de Pour de type 2 ⟩ , ⟨ liste de Pour de type 2 ⟩ | 56

⟨ élément d'une liste de Pour de type 3 ⟩ , ⟨ liste de Pour de type 2 ⟩ 57

⟨ liste de Pour de type 3 ⟩ ::= ⟨ élément d'une liste de Pour de type 3 ⟩ |

⟨ élément d'une liste de Pour de type 1 ⟩ , ⟨ liste de Pour de type 3 ⟩ | 58

⟨ élément d'une liste de Pour de type 2 ⟩ , ⟨ liste de Pour de type 3 ⟩ | 59

⟨ élément d'une liste de Pour de type 3 ⟩ , ⟨ liste de Pour de type 3 ⟩ 60

⟨ variable contrôlée ⟩ ::= POUR ⟨ variable ⟩ 61

⟨ prop. pour ⟩ ::= ⟨ var. contrôlée ⟩ : = ⟨ liste de Pour de type 1 ⟩ | 62

⟨ var. contrôlée ⟩ : = ⟨ liste de Pour de type 2 ⟩ | 63

⟨ var. contrôlée ⟩ : = ⟨ liste de Pour de type 3 ⟩ 64

⟨ Inst. Pour ⟩ ::= ⟨ prop. Pour ⟩ FAIRE ⟨ Instruction ⟩ | 65

⟨ prop. Pour ⟩ FAIRE | 66

⟨ étiquette ⟩ : ⟨ inst. Pour ⟩ (cf 4. 1. 1.)

4. 7. 1. Instruction procèdure

Règles 1, 2.

Règle 3. (cf 3. 2. 1.)

⟨ liste de paramètres effectifs ⟩ ::= ⟨ paramétre effectif ⟩ |

⟨ liste de paramètres effectifs ⟩ , ⟨ paramètre effectif ⟩ 67

⟨ Inst. procèdure ⟩ ::= ⟨ identificateur de procédure ⟩ |

⟨ ident. de procédure ⟩ (⟨ liste de paramètres effectifs ⟩) 68

⟨élément d'une liste de Pour de type 1⟩ , ⟨liste de Pour de type 3⟩ | 58

⟨élément d'une liste de Pour de type 2⟩ , ⟨liste de Pour de type 3⟩ | 59

⟨élément d'une liste de Pour de type 3⟩ , ⟨liste de Pour de type 3⟩ 60

⟨variable contrôlée⟩ ::= POUR ⟨variable⟩ 61

⟨prop. pour⟩ ::= ⟨var. contrôlée⟩ : = ⟨liste de Pour de type 1⟩ | 62

⟨var. contrôlée⟩ : = ⟨liste de Pour de type 2⟩ | 63

⟨var. contrôlée⟩ : = ⟨liste de Pour de type 3⟩ 64

⟨Inst. Pour⟩ ::= ⟨prop. Pour⟩ FAIRE ⟨Instruction⟩ | 65

⟨prop. Pour⟩ FAIRE | 66

⟨étiquette⟩ : ⟨inst. Pour⟩ (cf 4. 1. 1.)

4. 7. 1. Instruction procédure

Règles 1, 2.

Règle 3. (cf 3. 2. 1.)

⟨liste de paramètres effectifs⟩ ::= ⟨paramétre effectif⟩ |

⟨liste de paramètres effectifs⟩ , ⟨paramètre effectif⟩ 67

⟨Inst. procèdure⟩ ::= ⟨identificateur de procédure⟩ |

⟨ident. de procédure⟩ (⟨liste de paramètres effectifs⟩) 68

5. 1. 1. Déclarations de type

⟨liste de type⟩ ::= ⟨variable simple⟩ |

⟨liste de type⟩ , ⟨variable simple⟩ 69

⟨déclaration de type Réel⟩ ::= REEL ⟨liste de type⟩ 70

⟨déclaration de type entier⟩ ::= ENTIER ⟨liste de type⟩ 71

⟨déclaration de type Booléen⟩ ::= BOOLEEN ⟨liste de type⟩ 72

⟨déclaration locale⟩ ::= ⟨déclaration de type Réel⟩ |

⟨déclaration de type entier⟩ |

⟨déclaration de type Booléen⟩

⟨déclaration Rémanente⟩ ::= REMANENT ⟨déclaration locale⟩ 73

⟨déclaration de type⟩ ::= ⟨déclaration locale⟩ | ⟨déclaration Rémanente⟩

5. 2. 1. Déclarations de tableau

Règles 1, 2

⟨paire de bornes⟩ ::= ⟨borne inf.⟩ : ⟨borne sup.⟩ 74

⟨liste de paires de bornes⟩ ::= ⟨paire de bornes⟩ |

⟨liste de paires de bornes⟩ , ⟨paire de bornes⟩ 75

⟨Section de tableaux⟩ ::= ⟨ident. de tableau⟩ [⟨liste de paires de bornes⟩] | 76

⟨ident. de tableau⟩ , ⟨section de tableau⟩ 77

⟨liste de tableaux⟩ ::= ⟨section de tableaux⟩ |

⟨liste de tableaux⟩ , ⟨section de tableaux⟩ 78

⟨déclaration de tableau sans type⟩ ::= TABLEAU ⟨liste de tableaux⟩ 79

5.1.1. Déclarations de type

⟨liste de type⟩ ::= ⟨variable simple⟩ | ⟨liste de type⟩ , ⟨variable simple⟩ 69

⟨déclaration de type Réel⟩ ::= REEL ⟨liste de type⟩ 70

⟨déclaration de type entier⟩ ::= ENTIER ⟨liste de type⟩ 71

⟨déclaration de type Booléen⟩ ::= BOOLEEN ⟨liste de type⟩ 72

⟨déclaration locale⟩ ::= ⟨déclaration de type Réel⟩ | ⟨déclaration de type entier⟩ | ⟨déclaration de type Booléen⟩

⟨déclaration Rémanente⟩ ::= REMANENT ⟨déclaration locale⟩ 73

⟨déclaration de type⟩ ::= ⟨déclaration locale⟩ | ⟨déclaration Rémanente⟩

5.2.1. Déclarations de tableau

Règles 1, 2

⟨paire de bornes⟩ ::= ⟨borne inf.⟩ : ⟨borne sup.⟩ 74

⟨liste de paires de bornes⟩ ::= ⟨paire de bornes⟩ | ⟨liste de paires de bornes⟩ , ⟨paire de bornes⟩ 75

⟨Section de tableaux⟩ ::= ⟨ident. de tableau⟩ [⟨liste de paires de bornes⟩] | 76

⟨ident. de tableau⟩ , ⟨section de tableau⟩ 77

⟨liste de tableaux⟩ ::= ⟨section de tableaux⟩ | ⟨liste de tableaux⟩ , ⟨section de tableaux⟩ 78

⟨déclaration de tableau sans type⟩ ::= TABLEAU ⟨liste de tableaux⟩ 79

⟨Déclaration de tableaux Réels⟩ ::= REEL ⟨Déclaration de tableaux sans type⟩ 80

⟨Déclaration de tableaux Entiers⟩ ::= ENTIER ⟨Déclaration de tableaux sans type⟩ 81

⟨Déclaration de tableaux Booléens⟩ ::= BOOLEEN ⟨Déclaration de tableaux sanstype⟩ 82

⟨Déclaration locale de tableaux⟩ ::= ⟨Déclaration de tableaux Réels⟩ | ⟨Déclaration de tableaux entiers⟩ | ⟨Déclaration de tableaux Booléens⟩

⟨Déclaration Rémanente de tableaux⟩ ::= REMANENT ⟨Déclaration locale de tableaux⟩ 83

⟨Déclaration de tableau⟩ ::= ⟨Déclaration de tableaux sans type⟩ | ⟨Déclaration locale de tableaux⟩ | ⟨Déclaration Rémanente de tableaux⟩

5.3.1. Déclaration d'aiguillage

⟨liste d'aiguillage⟩ ::= ⟨Exp. de désignation⟩ | ⟨liste d'aiguillage⟩ , ⟨Exp. de désignation⟩ 84

⟨Tête d'aiguillage⟩ ::= AIGUILLAGE ⟨ident. d'aiguillage⟩ 85

⟨Déclar. d'aiguillage⟩ ::= ⟨Tête d'aiguillage⟩ :=⟨liste d'aig.⟩ 86

5.4.1. Déclaration de procèdures

On ne s'occupe pas dans ce travail des déclarations de procèdures : voir [8].

2. Matrice de priorité :

Rappelons d'abord les relations de priorité utilisées :

(1) $x \lessdot y$ si et seulement si il existe une règle $A ::= B\ x\ C$ telle que y soit descendant gauche de C.

(2) $x \gtrdot y$ si et seulement si il éxiste :

- soit une régle du type $A ::= B\ y\ C$ telle que x soit descendant droit de B.
- soit une règle du type $A ::= B\ x\ C\ y$

Notations : $x \in T$; $y \in T$; $A \in N$; $B, C \in N$ ou vide.

Rappelons que pour qu'on puisse utiliser facilement l'algorithme d'analyse décrit au chapitre I ($x = S(o)$, $y = S(E)$), il faut que ces deux relations soient incompatibles.

La recherche de ces relations est aisée à partir des règles écrites au § 1. Nous les présentons par un tableau (matrice de priorité) :

chaque case correspond à une "ligne" x et à une "colonne" y.
Elle comprend outre la relation existant éventuellement entre x et y les numéros de règles du § 1 :

- où entre y comparé à x si la relation est du type (1) (règles originant des sous-propositions dont y est le symbole médian).
- où entre x comparé à y si la relation est du type (2) (règles originant des sous-propositions qui sortent de la pile V).

Ces numéros de règles seront utilisés pour la détermination d'ordre Λ' où Λ (cf § 4).

Remarque :

Nous avons vu au chapitre I que dans le cas d'une régle du type $A ::= B\ x\ C\ y$, y n'entre pas dans la pile 0 ; par conséquent, la "ligne" y sera vide. Il est donc inutile de faire figurer cette ligne dans la matrice de priorité.

Ceci se produit pour les symboles)] Fin A1.

La matrice de priorité est donnée hors texte.

3. Etude des cas où les relations sont compatibles :

Il s'agit des couples :

① (:= SINON)

② (ALLER A SINON)

③ (SINON FAIRE)

④ (, ,)

1°) Les ambiguités dans ① , ② et ③ proviennent du fait que le "SINON d'instruction" et le "SINON d'expression" ont une valeur sémantique différente.

Il suffit alors de dédoubler SINON en $\text{SINON}_{\text{inst.}}$ et $\text{SINON}_{\text{exp.}}$. Cette distinction ne peut toutefois se faire à l'identification de SINON dans la suite d'entrée sans imposer un grand nombre de tests pour distinguer les règles. Elle est par contre aisée en procédant ainsi :

a) à l'identification, on considère SINON comme $\text{SINON}_{\text{inst.}}$

b) on a introduit auparavant dans le Λ de (SI, AL) un séparateur sur la pile V.

c) dans le Λ de (ORS , $\underline{\text{SINON}}_{\text{inst.}}$) , on teste si S (V) est ou non ce séparateur : en effet, dans le cas d'une expression conditionelle, S (V) est alors une adresse (cf chapitre IV) ; donc si le test répond OUI, $\underline{\text{SINON}}_{\text{inst.}}$ subsiste dans la suite d'entrée. Dans le cas contraire on le remplace par $\underline{\text{SINON}}_{\text{exp.}}$.

Remarques :

α) La distinction précédente permet de détecter aisément un certain nombre d'erreurs syntaxiques.

β) Dans la matrice de priorité, on doit substituer à la ligne et à la colonne "SINON" les deux lignes et deux colonnes "$\underline{\text{SINON}}_{\text{inst.}}$" et "$\text{SINON}_{\text{exp.}}$". Les nouvelles relations de priorités et leurs règles respectives se déduisent des anciennes en sélectant les règles d'instructions pour $\underline{\text{SINON}}_{\text{inst.}}$ d'une part et les règles d'expressions pour $\underline{\text{SINON}}_{\text{exp.}}$ d'autre part.

2°) L'ambiguité dans (4) provient du fait qu'on rencontre la "virgule" dans des sous-propositions dont les règles origines sont du type

(a) A ::= C u B / B

ou (b) A ::= B u A

Considèrons alors une règle du type (a) et une sous-proposition $\alpha = \alpha' u \beta$ dont elle est règle origine :

La sous-proposition α peut être remplacée par β, autrement dit $\alpha' u$ peut être supprimé. Il en résulte que les tests permettant l'entrée et la sortie des signes qui composent β donneront le même résultat

et détermineront la même règle (règle origine d'une sous-proposition de β) si on enlève n de la pile 0.

Si de plus le Λ correspondant à la règle est vide (c'est-à-dire en particulier : m (α) = m (β) ; i (α) = i (β) ; m (α') et i (α') sortis au moment de l'entrée de u) , il est inutile de faire entrer u dans la pile 0. Cependant, au moment où u aurait dû entrer, le Λ' correspondant sera bien exécuté.

On raisonne de la même manière à partir des règles du type (b) .

Comme on trouve "virgule" dans des règles du type (a) (avec d'ailleurs C = A) ou (b) , dont les premiers membres sont :

< liste d'indices > ;
< liste de paramètres effectifs >
< liste de paires de bornes >
< section de tableaux >
< liste de tableaux >
< liste d'aiguillage >
< liste de pour de type 1 >
< liste de pour de type 2 >
< liste de pour de type 3 >
< liste de type >
< liste de tableaux >

on peut appliquer ce qui précéde à ce symbole. La "ligne" référenceé "virgule" sera donc supprimée.

4. Utilisation de la matrice pour la détermination des Λ et Λ'

On a vu au chapitre I qu'à chaque entrée ou sortie d'un élément de la pile 0, on doit exécuter une séquence d'ordres Λ ou Λ' dépendant d'une règle.

Cette règle est celle dont le numéro se trouve dans la case de la matrice qui provoque cette entrée ou cette sortie ($\mathcal{G}2$).

Certaines cases contiennent plusieurs règles. Il faudra :

1°) Soit que Λ ou Λ' soit le même.

Exemple :

ORS $\succ$; Règles 44, 45, 48.

Dans tous les cas, le Λ calcule l'adresse où on sautera lorsqu'à l'éxecution du programme objet la variable booléenne prendra la valeur FAl

2°) Soit distinguer les différents Λ ou Λ' par d'autres tests

Exemple :

ORS $\succ$ SINON $_{exp}$ Règles 15, 29, 33.

On engendre des $\vdash$ différents suivant qu'il sagit d'une expression arithmétique (15), booléenne (29) ou de désignation (33).

Nous verrons au chapitre IV sous forme d'ordinogrammes commen on fait les tests permettant ces distinctions, grâce à un indicatif i se trouvant en S (V).

En fait, on distingue les Λ' des Λ, car d'une part les Λ' non vides sont peu nombreux, d'autre part les Λ ne dépendent très souvent que de la ligne (cf chapitre IV).

On procède ainsi :

a) A chaque case (x,y) pour laquelle il existe au moins un Λ' non vid on fait correspondre une séquence d'ordres $\Sigma'(x,y)$ qui contient :

- ou le Λ' s'il est unique
- ou les différents Λ' correspondants aux différentes règles

$\Sigma'(x,y)$ les distingue par différents tests (cf. chapitre IV).

b) A chaque ligne x on fait correspondre une séquence d'ordres $\Sigma(x)$ qui contient :

- ou le Λ s'il est unique
- ou les différents Λ correspondants aux différents y d'une part et aux différentes règles pour un y donné d'autre part. $\Sigma(x)$ les distingue par différents tests.

3°) Soit dédoubler un symbole :

a) cas de + et - :

Les règles 10 et 11 se distinguent respectivement des règles 12 et 13 par le fait que, dans E, + et - succédent :

- à un symbole de base autre que) et] pour les premières
- à un identificateur ou) ou] pour les secondes.

Il est donc possible de faire les distinctions dès l'identification, en dédoublant les symboles.

Nous avons alors quatre nouveaux symboles :

(+u) (à un opérande), (-D) (à deux opérandes), (-u) et (-D).

D'où de nouvelles priorités déterminées de la même manière. En rèalité, si on supprime les (+u), un programme syntaxiquement correct le reste et garde même signification ; donc, on effectue d'abord le test de priorité pour détecter une erreur éventuelle, car un programme incorrect pourrait être transformé en un programme correct par la simple suppression du (+u) (ce qu'il faut éviter) ; mais on ne fait pas entrer (+u) dans la pile 0.

Par conséquent, dans la matrice, aux deux "lignes" et aux deux "colonnes" + et - on substitue :

- une colonne (+ u)
- une ligne et une colonne pour chacun des symboles (+ D) , (- u) , (- D).

b) cas de :=

$\sum$(:=) doit faire des tests, dans le cas où S (E) est "point virgule", pour distinguer "Instruction d'affectation" (Règle 42) de "déclaration d'aiguillage" (Règle 86). De même, $\sum'$(:=,) doit distinguer "Instruction Pour" (Règles 52 à 60) de "déclaration d'aiguillage" (Règle 84). On évite ces tests en transformant le := de la déclaration d'aiguillage, grâce au symbole AIGUILLAGE qui sort juste avant l'entrée de ce :=. Pour éviter un nouveau symbole et des sorties inutiles, on transformera := en AIGUILLAGE, de sorte que :

α) tout se passe comme si pour le Λ associé à AIGUILLAGE et le Λ' associé à := , := était détruit et AIGUILLAGE subsistait.

β) Les éléments qui étaient comparés à S (0) = : = à cause de la Règle 86 sont comparés à AIGUILLAGE.

Nous transformons la matrice de priorité en ce sens.

Remarque : Cette manière de faire n'a pas d'incidences sur le test "S(0) déclarateur" car les déclarations d'aiguillage et les déclarations de types sont traitées différemment (cf chapitre IV).

5. Réalisations pratiques :

a) Symboles :

On fait correspondre à tout symbole, dés son identification, un équivalent numérique de deux positions. Ces équivalents sont pris de 1 en 1 à partir de 01. Alors tous les tests effectués ensuite sur les symboles se feront à partir de leur équivalent numérique.

b) AL et ORS :

AL est une "parenthèse fermée", c'est-à- dire que, comparé à SI , il fait sortir ce dernier de la pile 0 et s'élimine lui-même de la pile E. Vient ensuite ORS qui entrera obligatoirement sur la pile 0, car la colonne de ORS à les mêmes priorités que la colonne de SI , qui vient de sortir, donc qui était entré précédemment sur le même élément de la pile.

Alors, on peut garder un seul symbole ALORS tel que :

- Les priorités relatives à ALORS = S (E) sont celles de AL c'est-à-dire que la "colonne" ALORS est l'ancienne "colonne" AL.
- Le nouveau $\sum$ (SI) , qui se réduit en fait au Λ (SI, ALORS) génère le même Γ que l'ancien, mais substitue ensuite ALORS à SI dans la pile 0.
- Les priorités relatives à ALORS = S (0) sont celles de ORS, c'est-à-dire que la "ligne" ALORS est l'ancienne ligne ORS.

c) Mémorisation de la matrice :

D'après tout ce qui précéde c'est une matrice (42,46). Chaque élément occupe une position de mémoire. Dans cette position on convient de charger :

0 si S (0) $\lessdot$ S (E) et le Λ' est vide (cas fréquent)
3 si S (0) $\lessdot$ S (E) et le Λ' n'est pas vide
1 si S (0) $\gtrdot$ S (E)
2 si S (0) et S (E) non comparable (erreur syntaxique)

La matrice est rangée ligne par ligne ; l'ordre de rangement de ces lignes se fait suivant l'ordre croissant des équivalents numériques des symboles les référençant. De même, l'ordre des colonnes pour chaque ligne correspond à l'ordre croissant des équivalents numériques des symboles référençant ces colonnes.

Alors :

- On calcule l'adresse d'un élément quelconque (S(0), S(E)) par

$$K + 46 \times (S(0) - 1) + S(E)$$

K étant l'adresse du premier élément de la matrice

- Les références des $\Sigma(x)$ étant rangées en séquence, ordonnées suivant les valeurs croissantes de x, l'adresse de la référence de $\Sigma(S(0))$ est donnée par :

$$A + 5 \times (S(0) - 1)$$

A étant l'adresse de la première référence rangée.

La matrice sous sa forme définitive est donnée hors texte.

CHAPITRE III

ORGANISATION GENERALE

Notation : Nous écrivons entre parenthèses les références du compilateur.

1. Ecriture du programme source

1°) Représentation des symboles de base :

on convient de représenter les symboles de base par :

- ou un caractère spécial de 1620 I. B. M [10]
- ou un "point" suivi d'un caractère spécial
- ou une suite de lettres entre deux "points".

Cette nomenclature facilitera les tests d'identification à la compilation.

Algol	VRAI	FAUX	+	-	x
Compilateur	. VRAI.	. FAUX.	+	-	*

÷	↑	<	≤	=	≥	>	≠
. DIVENT.	$	. INF.	. ING.	=	. SUG.	. SUP.	. DIF.

≡	⊃	∨	∧	¬	ALLERA	SI
. IDEN.	. IMP.	. OU.	. ET.	. NON.	. ALLERA.	. SI.

ALORS	SINON	POUR	FAIRE	REMANENT	BOOLEEN
. ALORS.	. SINON.	. POUR.	. FAIRE.	. REMANENT.	. BOOLEEN.

ENTIER	REEL	TABLEAU	AIGUILLAGE	PROCEDURE
. ENTIER.	. REEL.	. TABLEAU.	. AIGUILLAGE.	. PROCEDURE.

CHAINE	ETIQUETTE	VALEUR	,	.	10	:	;
. CHAINE.	. ETIQUETTE.	. VALEUR.	,	.	. Z	..	. ,

: =	␣	PAS	JUSQU'A	TANT QUE	COMMENTAIRE
. =	ƀ	.PAS.	.JUSQUA.	.TANTQUE.	.COMMENTAIRE.

(	)	⊏	⊐	`	'	DEBUT	FIN
(	)	.(	.)	.GUO.	.GUF.	.DEBUT.	.FIN.

2°) Représentation des identificateurs :

Tout identificateur est formé d'une suite de lettres majuscules et de chiffres commençant par une lettre. Le nombre de caractères n'est pas limité. Toutefois, pour des raisons de réservations dans la table à la compilation, un identificateur de plus de 7 caractères sera traité par le compilateur comme un identificateur comprenant les 7 premiers caractères seulement.

3°) Représentation des nombres :

C'est la représentation prévue par [6], elle est détaillée dans [illegible]

2. Compilation :

1°) Entrée du programme source :

Le programme Algol est perforé sur cartes. Ces cartes sont lues une à une en mode alphanumérique à partir de (SE), et leur contenu est analysé à l'aide :

a) d'un compteur (MEMSE) de 5 positions, qui contient l'adresse "moins un" du prochain caractère à identifier. Ce compteur est donc initialisé par (SE-2).

b) d'un sous-programme (DECAL) qui rend dans (MEMSE) l'adresse de la zone réduite au caractère suivant. Ce sous-programme se charge en outre d'introduire une nouvelle carte par (SSP4), et de supprimer les "blancs" par (SSP5).

2°) Identifications :

Lorsqu'on veut reconnaître un nouvel élément du programme source, on appelle (DECAL) et on effectue au retour des tests sur la zone dont l'adresse est dans (MEMSE). Cette zone est la représentation alphanumérique d'un caractère qui peut être :

a) une lettre :

On entre alors dans la formation d'un identificateur. On s'adresse au sous-programme (FORMAT), qui appelle (DECAL) jusqu'à la rencontre, dans E, d'un caractère autre qu'une lettre ou un chiffre ; puis, il envoie l'identificateur ainsi formé dans une zone de travail (ATENT2) de 14 positions ; l'identificateur est cadré à gauche s'il comprend moins de 7 caractères. S'il comprend plus de 7 caractères, seuls les 7 premiers sont pris en considération.

Remarque, un identificateur peut être perforé sur deux cartes consécutives. Mais, lorsque (DECAL) "tourne la page", la nouvelle carte lue se substitue à la précédente. Alors, avant une nouvelle lecture, la dernière zone reconnue dans E est transférée devant la bande de lecture.

b) Un chiffre. :

On entre dans la formation d'un nombre. On s'adresse alors à un sous-programme spécial de [5] (cf. chapitre I)

c) Un "point" :

On appelle le caractère suivant par (DECAL). On a cette fois dans E :

α) Z ou un chiffre : il s'agit là encore d'un nombre

β) un caractère spécial : on reconnait un symbole de base. Le sous-programme (SSP1) l'identifie et transfère son équivalent numérique dans E. (cf chapitre II).

γ) Une lettre : on reconnait un symbole de base écrit entre deux "points". On s'adresse alors à (SSP3) qui :

- identifie le symbole de base à partir des 3 premiers caractères suivant le "point".

- fait progresser (MEMSE) jusqu'au "point" suivant.

Puis on transfère l'équivalent numérique du symbole dans E.

Remarque : (SSP3) fait progresser (MEMSE) sans appeler (DECAL) et sans autres tests dans E. Celui-ci doit alors "tourner la page" en respectant une "marge de sécurité", c'est-à-dire sans attendre l'identification des derniers caractères (qui peuvent être une "partie" de symbole). Les caractères non analysés au moment d'une nouvelle lecture sont alors transférés devant (SE). La longueur de ce transfert détermine la nouvelle initialisation de (MEMSE).

d) un caractère spécial :

on reconnait un symbole de base. (SSP2) le recherche par consultation de table, et transfère dans E son équivalent numérique.

3°) Piles :

a) pile E :

on a convenu d'appeler la suite d'entrée "pile E". (MEMSE) indique à chaque instant l'adresse du sommet de la pile. On réserve à cette pile une bande fixe de 211 positions, réparties ainsi, suivant les adresses croissantes :

- 50 positions utilisées par (DECAL) pour "tourner la page" (cf les remarques précédentes).

- 160 positions utilisées pour la lecture alphanumérique d'une carte du programme source.

- 1 position occupée par une marque d'enregistrement.

b) pile 0

Elle reçoit les équivalents numériques des symboles de base (cf chapitre II). A tout instant, l'adresse de la zone située au sommet de cette pile est dans un compteur (MEMSO). Ce compteur est initialisé par (SØ). La pile 0 se remplit suivant les adresses croissantes.

c) pile V :

Elle reçoit les adresses α des zones affectées aux identificateurs, nombres, variables, ou sous-propositions (lorsqu'elles sont "composantes" de sous-propositions plus grandes), ainsi que leur type I.

C'est-à-dire qu'un élément de V s'écrit α I, et :

- l'adresse α occupe 5 positions

- le type I occupe 1 position

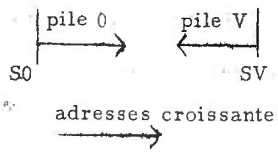
I prend les valeurs :

0 pour un type Réel
1 pour un type Entier
2 pour un aiguillage
3 pour un type booléen
4 pour une étiquette

d'autres valeurs comprises entre 5 et 9 sont utilisées par les procédures [8] et les tableaux [9] ;

Quand I = 0, 1, 3 ou 4, on lui adjoint un "flag" lorsque l'adresse α indique un résultat intermédiaire.

Les opérations sur la pile V se font à l'aide d'un compteur (MEMSV), qui indique, en dehors des Σ et Σ', l'adresse du type I de l'élément situé en S(V). (MEMSV) est initialisé par (SV). La pile se remplit suivant les adresses décroissantes. Elle est "en face" de la pile 0, c'est-à-dire que nous avons en mémoire :



Cette représentation facilite les réservations.

Par ailleurs, la pile V est également utilisée pour stocker :

α) en début de bloc, les contenus du compteur d'affectation (MRINT) et de l'indicateur d'adresse de table (CPTR2). Les adresses ainsi stockées seront renvoyées dans ces compteurs à la fin de la compilation du bloc, c'est-à-dire dans le Λ (DEBUT FIN). Cela revient à libérer à la fin de

chaque bloc les zones réservées aux variables locales et les positions de la table utilisées par les identificateurs de ce bloc.

β) Lors d'une déclaration, l'adresse dans la table où on doit stocker le type I de l'identificateur déclaré.

4°) Table des identificateurs :

On range dans une table les identificateurs du programme source, leur "adresse" dans le programme objet (ce sera pour 1 réel ou 1 entier l'adresse de la zone qui lui sera affectée à l'exécution), et leur type ; on effectue des transferts dans la table :

- Lorsqu'on entre dans un bloc, c'est-à-dire à l'identification de DEBUT suivi d'un déclarateur. Le transfert porte sur les aiguillages, les étiquettes et les procédures : Ces identificateurs ont été répertoriés lors d'une première lecture (cf § 3).

- au moment des déclarations des Réels, Entiers, Booléens, et des tableaux. Une "ligne" de la table occupe 22 positions, réparties de la manière suivante :

- l'identificateur occupe 14 positions
- l'adresse correspondante α occupe 5 positions
- le type I occupe 1 position
- Une zone de 2 positions utilisée par les étiquettes et les aiguillages pour garder le "niveau de procédure", ainsi que par les tableaux [9].

Un compteur (CPTR2) indique l'adresse de la dernière position de la dernière ligne construite.

a) formation de la table :

On ne parle ici que de la formation lors des déclarations :

Lorsqu'un identificateur est formé dans (ATENT2) (cf § 2), et si S (0) e un déclarateur, autre que AIGUILLAGE ou PROCEDURE, alors :

α) on introduit l'identificateur dans une nouvelle ligne de la table,

β) on envoie sur la pile V l'adresse de la table à laquelle on doit range le type de cet identificateur. On rangera le type et l'adresse "d'exécution" affectée à l'identificateur dans le Λ(S (0) ;) ou le Λ' (S (0) ,) suivant. Cette adresse "d'exécution" sera prise dans un compteur d'affectation de zone, (MRINT) s'il s'agit d'une variable locale, (REMAN) s'il s'agit d'une variable rémanente (reconnaissable par le symbole situé sous S (0) qui peut être REMANENT). Le compteur utilisé régressera ensuite d'une quantité en liaison ave le type. Ainsi, on réserve à un Réel ou à un entier une zone de 12 positions, à un booléen une seule position.

b) utilisation de la table :

Si S (0) n'est pas un déclarateur ou si S (0) est AIGUILLAGE, l'identificateur stocké dans (ATENT2) se trouve déjà dans la table (cf § 3). On consulte alors la table par le sous-programme (CONSUL), qui :

- envoie en S (V) l'adresse associée à l'identificateur, et le type,
- laisse dans un compteur (CPTR3), jusqu'à la prochaine consultation de table, l'adresse de la dernière zone de la "ligne". Cette zone peut être utilisée dans le Λ' ou le Λ suivant (cf chapitre IV)

c) portée de la table :

Un identificateur, ne pouvant être utilisé à l'extérieur du bloc dans lequel il a été déclaré, ne restera pas dans la table après la compilation de ce bloc. Pour la même raison, on libére à la fin d'un bloc les zones d'exécution réservées pour ses variables locales. On procéde ainsi :

- quand on entre dans un bloc, c'est-à-dire à l'identification de DEBUT suivi d'un déclarateur, on garde sur la pile V les contenus de (CPTR2) et de (MRINT). On effectue ces transferts dans la partie (SDEB).

- A la fin de la compilation du bloc, c'est-à-dire dans le Λ (DEBUT FIN), on forcera (CPTR2) et (MRINT) aux adresses envoyées précédemment dans V.

5°) Programme généré :

Le programme objet est construit par concaténation (cf chapitre I). Les Γ et Γ' sont construits dans une zone de travail (C4), puis sont transférés dans une zone de stockage (C3) ; l'adresse à partir de laquelle on peut stocker une nouvelle séquence est dans un compteur (MPGØB). Toutefois, à cause des réservations, on ne peut stocker tout le programme objet en mémoire. Alors :

1°) En initialisation, on perfore une carte de chargement pour le programme objet, et on affecte à un compteur (CHØB) l'adresse de la première instruction du programme objet à l'exécution.

2°) Avant d'entrer dans un Σ' ou un Σ, on appelle un sous-programme (PFØ) qui, si le stockage dans (C3) dépassse 60 caractères numériques, perfore dans une zone (C1) une carte comprenant :

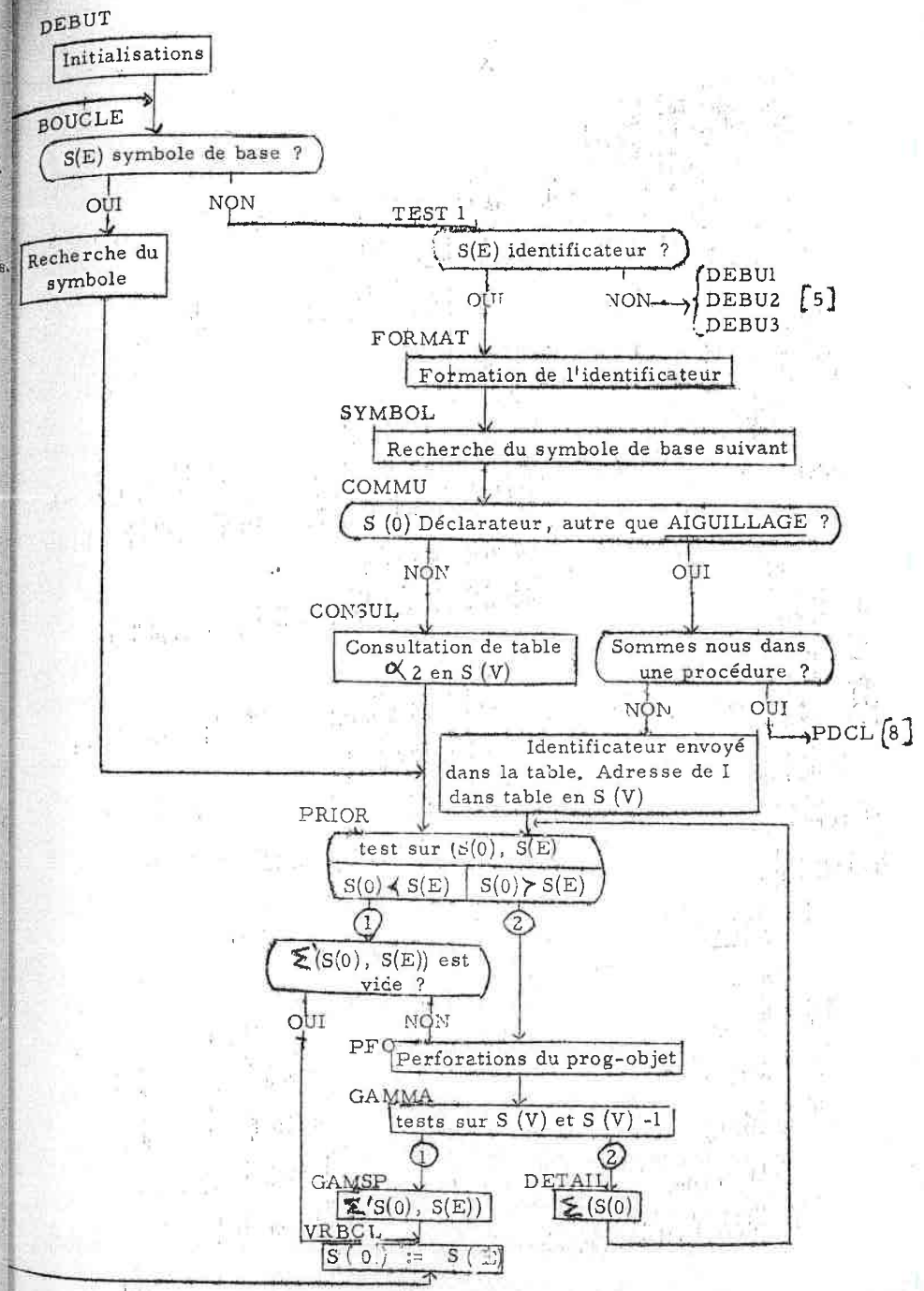
Colonnes 1 à 60 : les 60 premières positions à partir de (C3).

Colonne 61 : une marque d'enregistrement

Colonnes 70 à 74 : l'adresse "plus un" de la fin de cet enregistrement dans le programme objet.

Colonnes 75 à 79 : l'adresse de cet enregistrement dans le programme objet.

Ensuite, (PFØ) supprime dans (C3) les 60 positions perforées, diminue (MPGØB) de 60 et augmente (CHØB) de 60.



3. Première lecture :

L'algorithme décrit dans le paragraphe précédent ne permet de traiter les identificateurs qu'après leur déclaration :

(a) Les réels, entiers et booléens déclarés en-tête d'un bloc B et utilisés dans une déclaration D d'aiguillage ou de procédure située en-tête du même bloc B, doivent être déclarés avant D.

(b) Si on considère qu'une étiquette est déclarée lorsqu'on la rencontre suivie de "deux points", une étiquette ne pourrait être utilisée dans une expression de désignation avant sa "déclaration".

(c) Enfin, on n'accepterait pas les aiguillages ou les procédures "croisés".

(a) n'entraîne aucune restriction sur les possibilités d'ALGOL. Par contre, (b) et (c) impliquent des restrictions importantes.

Alors, une première lecture du programme source permettra, lorsqu'on entrera dans un bloc à la compilation, de connaître à priori les identificateurs du bloc qui sont déclarés étiquettes, aiguillages et procédures.

1°) Description de la première lecture :

On ne parle pas ici du traitement des procédures [8].

La référence d'appel du programme est (BEGIN). Le programme source est entré en mémoire de la même manière qu'à la compilation. Ici encore, on appelle (DECAL) pour identifier le caractère suivant. Toutefois, on ne s'intéresse qu'à un nombre limité de symboles et d'identificateurs.

La première lecture forme une table, à l'aide d'un compteur (LTAB).

On construit une "ligne" de cette table lorsqu'on rencontre un séparateur de bloc DEBUT ou FIN, une étiquette suivie de "deux points", un identificateur d'aiguillage déclaré. Chaque "ligne" occupe 22 positions :

DEBUT ≠ | ① 5 pos. | ② 5 pos. | ③ 5 pos. | ④ 5 pos. | X

Etiquette suivie de "deux points" : étiquette cadrée à gauche (14 pos.) | 000004 | ⑤ 2 pos.

Identificateur d'aiguillage : identificateur cadré à gauche (14 pos.) | 000002 | ⑤ 2 pos.

FIN ≠ | ① 5 pos. | ② 5 pos. | ③ 5 pos. | Vide 6 pos.

① contient l'adresse de ② du séparateur de bloc DEBUT ou FIN suivant.

Il indique "jusqu'où transférer à partir d'ici".

② Contient | pour DEBUT, l'adresse de ① du FIN correspondant
| pour FIN, un indicatif d'arrêt 00000

Il indique "où reprendre le transfert interrompu ici".

③ contient la longueur de l'enregistrement compris entre ce séparateur et le séparateur suivant.

④ Contient l'adresse de ① du DEBUT suivant

⑤ Reçoit le "niveau de procédure" se trouvant dans une zone (CTNIV).

Cette représentation nécessite l'utilisation :

De deux compteurs fixes (LC1) et (LC2), qui gardent respectivement l'adresse de ④ du dernier DEBUT rencontré et l'adresse de ① du dernier DEBUT ou FIN rencontré.

- D'une pile destinée à recevoir les adressses de (2) des DEBUT, car ces adresses sont utilisées lorsqu'on rencontre les FIN correspondants. On utilise la pile 0 de la compilation. On y introduit, chaque fois que l'on identifie un DEBUT de bloc l'équivalent numérique de DEBUT suivi de l'adresse de (2) (un élément de la pile occupe donc 7 positions). Par ailleurs, cette pile est utilisée pour la reconnaissance des étiquettes. En effet, l'identification de "deux points" ne suffit pas, car on peut se trouver dans une "paire de bornes" d'un déclaration de tableau, ou dans un séparateur. Alors, on fait entrer [dans la pile, et on le supprime lorsqu'on rencontre le] correspondant dans E. En définitive, lorsqu'on identifie "deux points", on sait qu'il appartient à une déclaration d'étiquette si S (0) n'est pas [et si "deux points" n'est pas suivi de (dans E.

Remarque :

Les identificateurs d'aiguillage sont facilement reconnaissables car ils suivent toujours le symbole AIGUILLAGE.

2°) Utilisation de la table construite à la première lecture :

Cette table sera utilisée à la compilation :

La nomenclature de la table permettra, lorsqu'on entrera dans un bloc (c'est-à-dire dans la partie (SDEB)), de transférer dans la table des identificateurs les étiquettes et les identificateurs d'aiguillage de ce bloc sans transférer ceux des blocs intérieurs s'ils éxistent.

Toutefois, la première lecture ne met pas dans la table qu'elle construit les adresses "d'éxécution" des étiquettes et des identificateurs d'aiguillage. Donc, à la compilation, un identificateur trouvé par consultation

de table peut être :

a) une étiquette, reconnaissable à l'indicatif 4, et l'adresse correspondante peut être $\bar{0}0000$. Alors, si le symbole suivant de E est "deux points", on calcule à l'aide de (CHØB), (MPGØ B) et (C3) l'adresse réelle correspondant à l'étiquette dans le programme objet ; on substitue l'adresse calculée à $\bar{0}0000$ dans la table et on saute à (BOUCLE). Si le symbole suivant dans E n'est pas "deux points", c'est-à-dire si l'étiquette est utilisée dans une expression de désignation, on lui affecte une zone de 5 positions de (REMAN) ; cette zone sera utilisée comme adresse indirecte à l'éxécution. On substitue cette adresse indirecte à $\bar{0}0000$ dans la table, et on l'envoie en S (V) suivie de l'indicatif 4 ; Alors, lorsqu'on rencontrera la même étiquette, mais suivie de "deux points", on perforera une carte pour le programme objet, à l'aide du sous-programme (PFØ SI). Cette carte chargera à l'exécution l'adresse réelle de l'étiquette dans la zone d'adressage indirecte.

b) un aiguillage, reconnaissable à l'indicatif 2, et l'adresse correspondante peut être $\bar{0}0000$. Alors, on utilise là encore une adresse indirecte, qu'on substitue à $\bar{0}0000$ dans la table, et qu'on envoie en S (V) suivie de l'indicatif 2. Si l'adresse correspondante dans la table n'est pas $\bar{0}0000$, c'est-à-dire si on ne rencontre pas l'identificateur d'aiguillage, on procéde comme pour un identificateur normal.

On précisera au chapitre IV l'utilisation de cette adresse indirecte dans une déclaration d'aiguillage.

4. Traitement des commentaires :

A la première lecture et à la compilation :

1°) lorsqu'on identifie COMMENTAIRE , on progresse dans E jusqu'au premier "point virgule" inclus. De plus, à la compilation, on vérifie que COMMENTAIRE suit DEBUT ou "point virgule".

2°) Dans le Σ(DEBUT, FIN) , et si S (0) n'est pas le premier début du programme source, on progresse dans E jusqu'au premier "point virgule", FIN , ou SINON.

5. Traitement des valeurs logiques :

Quand on identifie VRAI, on envoie en S (V) l'adresse d'une position qui contiendra 0 à l'éxécution, puis le type Booléen c'est-à-dire I = 3.

Quand on identifie FAUX , on envoie en S (V) l'adresse d'une position qui contiendra 1 à l'exécution, puis le type booléen.

CHAPITRE IV

DESCRIPTION DES Σ et Σ'

I - RAPPELS et NOTATIONS

1°) Lorsqu'un symbole entre ou sort de la pile 0, on exécute une séquence Σ'(S (0), S (E)) ou Σ(S (0)) du programme, qui :

- détermine la règle par différents tests (cf chapitre II)
- construit le Γ' ou Γ correspondant, les opérandes étant sur la pile V (cf chapitre I).
- modifie les piles : En général, dans les Σ, ces modifications consistent à supprimer les opérandes de la pile V, à envoyer dans cette pile l'adresse du Résultat si la sous-proposition compilée est une expression, et à supprimer S (0). Dans l'énoncé des Σ, nous préciserons seulement les modifications de piles qui offrent d'autres particularités.

2°) Une sous-proposition entre dans la pile V sous la forme αI; α représente une adresse "d'exécution" et I un type (cf chapitre III).

L'adresse α peut être indirecte. C'est alors :

- une étiquette si I = 4 ou $\bar{4}$ (cf chapitre III)
- un identificateur d'aiguillage si I = 2 (cf chapitre III)
- le résultat d'un calcul d'adresse d'élément de tableau [9] si I = 0, 1 ou 3. α est alors l'adresse d'une zone de 5 positions des résultats intermédiaires. Cette zone est gérée à la compilation par le compteur

(MRINT), déjà utilisé dans les déclarations pour l'affectation de zones aux variables locales (cf chapitre II, § 2). Les positions occupées par α devront être libérées lorsque α sortira de la pile V

Notations :

a) on appelle (α) la quantité adressée par α.

b) on appelle : $\alpha_2\ I_2$ le sommet S (V)

et $\alpha_{I_1}\ I_1$ l'élément situé sous S (V)

c) on numérote les cas :

$\alpha_2\ I_2$	$\alpha_2\ \bar{I}_2$	$\alpha_2\ I_2$	$\alpha_2\ \bar{I}_2$
$\alpha_1\ I_1$	$\alpha_1\ I_1$	$\alpha_1\ \bar{I}_1$	$\alpha_1\ \bar{I}_1$
①	②	③	④

II - SOUS-PROGRAMMES

De nombreux tests sont communs à plusieurs Σ. Ces tests figurent alors sous forme de sous-programmes :

1°) Le sous-programme (GSI) formedes positions "flag".

(FLR2) si α_2 est REEL.

(FLB2) si α_2 est BOOLEEN.

(FLEG) si α_1 et α_2 sont de même type.

(FL2RP) si α_2 est un Résultat partiel.

(FL1RP) si α_1 est un Résultat partiel.

(FL2T) si α_2 est l'adresse d'un élément de table

2°) Le sous-programme (SPG1) libére les positions occupées dans (MRINT) par l'opérande de type BOOLEEN qu'on lui désigne (adresse de tableau ou Résultat intermédiaire).

Ce sous-programme est commun aux Σ de $\wedge\ \vee\ \supset\ \equiv\ \Gamma$ SI ou TANT QUE .

3°) Le sous-programme (SPG2) construit le transfert dans une zone fixe d'exécution de l'opérande désigné, si cet opérande représente l'adresse (indirecte d'après ce qui précéde) d'un élément de tableau.

Ce sous-programme est commun aux Σ des opérateurs arithmétiques, excepté (- U) , et des opérateurs de relation.

III ENONCE DES Σ ET Σ'

1. Opérateurs logiques : $\wedge\ \vee\ \supset\ \equiv\ \Gamma$

Les Σ' sont vides et pour chaque Σ la règle est unique.

1a) $\Sigma(\wedge)$ est référencé (ET). Il construit le Γ :

$$R := (\alpha_1) + (\alpha_2) - '(\alpha_1) \times (\alpha_2)$$

à l'exécution, on aura :

$(R) = 0 \Longleftrightarrow {}'(\alpha_1) = (\alpha_2) = 0$

Donc : R est | VRAI si α_1 et α_2 sont VRAI
| FAUX dans tout autre cas

2°) Σ (V) est référencé (ou). Il construit le Γ :

$$R := (\alpha_1) \times (\alpha_2)$$

$(R) = 1 \Longleftrightarrow (\alpha_1) = (\alpha_2) = 1$

Donc: R est | FAUX si α_1 et α_2 sont tous les deux FAUX
| VRAI dans tout autre cas.

3°) $\Sigma(\supset)$ est référencé (IMP). Il construit le Γ :

R := SI (α_1) < (α_2) ALORS 1 SINON 0

(R) = 1 $\Longleftrightarrow$ (α_1) = 1 et (α_2) = 0

Donc: R est | FAUX si α_1 est FAUX et α_2 est VRAI
| VRAI dans tout autre cas.

4°) $\Sigma(\equiv)$ est référencé (IDEN). Il construit le Γ :

R := SI (α_1) = (α_2) ALORS 0 SINON 1

(R) = 0 $\Longleftrightarrow$ (α_1) = (α_2)

Donc : R est | VRAI si α_1 et α_2 sont tous les deux VRAI ou tous les deux FAUX
| FAUX dans tout autre cas.

5°) $\Sigma(\neg)$ est référencé (NON). Il construit le Γ :

R := 1 - (α_2)

(R) = 0 $\Longleftrightarrow$ (α_2) = 1

Donc : R est | VRAI si α_2 est FAUX
| FAUX si α_2 est VRAI

Dans tous les cas, R est l'adresse de la première position libre dans la zone des résultats intermédiaires, étant entendu que les positions occupées par α_1 ou α_2 dans cette zone ont été libérées (cf sous-programme (SPG1))

2. Opérations arithmétiques : (+ U) (- U) (+ D) (- D) X / ↑ ÷

Les Σ' des opérateurs autres que (+ U) sont vides, et la régle est unique pour chaque Σ.

1°) Σ'(S (0), (+ U)) ne fait que supprimer (+ U) dans la pile E (cf chapitre II, § 4).

2°) Σ ((- U)) est référencé (SOUS). Il construit le Γ :

R := - (α_2)

α_2 est REEL ou ENTIER. Le résultat R est de même type, et c'est :

cas (1) et (2) : l'adresse de (MRINT). Ce compteur régresse alors de 12.

cas (3) et (4) : α_2 .

3°) Les Σ de (+ D) (- D) X et / sont décrits par des ordinogrammes.

Σ((+ D)) est référencé (PLUS) : ordinogramme 1

Σ((- D)) est référencé (MOINS) : ordinogramme 2

Σ(X) est référencé (MUL) : ordinogramme 1.

Σ(/) est référencé (DIV) : ordinogramme 2.

4°) Σ(↑) est référencé (EXPO) :

Le résultat est toujours REEL, contrairement à [6] . Cela provient du fait que le type du résultat prévu par [6] dépend dans certains cas de la valeur (α_2), connue seulement à l'exécution. Or les opérations arithmétiques effectuées par le programme objet ne sont pas "interprétées".

Alors, il est nécessaire de connaître le type du Résultat à la compilation.

Le Γ construit dans ce Σ est une séquence qui effectuera les transferts :

- de (α_1) et (α_2) à un sous-programme d'exécution décrit par [5] ;
- du Résultat en "Résultat intermédiaire".

L'adresse R du résultat est :

Cas ① : l'adresse de (MRINT). Ce compteur régresse alors de 12.

Cas ② : α_1

Cas ③ : α_2

Cas ④ : α_1 et (MRINT) progresse de 12, afin de "libérer" la zone occupée par (α_2).

5°) Σ (↑) est référencé (DENT) :

D'après [6] , α_1 et α_2 sont du type ENTIER, et le résultat est ENTIER.

Ici nous devons tenir compte de la modification faite à l'exponentiation (Résultat toujours REEL). En effet, avec les conditions de [6] , il serait impossible de compiler, par exemple, une expression de la forme :

A ↑ B ↑ C avec A, B et C ENTIER.

Alors, nous acceptons des opérandes du type REEL. Ils seront convertis au type ENTIER par un sous-programme d'exécution décrit dans [5].

Le Γ construit dans ce Σ est donc constitué par :

- les ordres de conversion éventuelle

- R := (α_1) ↑ (α_2)

Le résultat (R) est ENTIER. L'adresse R du résultat est choisie comme dans 4°).

3. Relations : $< \leqslant = \geqslant > \neq$

Les Σ sont vides et la règle est unique pour chaque Σ.

α_1 et α_2 sont du type REEL ou ENTIER. Si les types sont différents, on construit les ordres de conversion en REEL de l'opérande ENTIER.

Les Σ sont identiques, exception faite de la construction de l'ordre "Saut si indicateur" des Γ, qui représentent, en plus des conversions eventuelles :

R := SI (α_1) <op. de Relation> (α_2) ALORS 0 SINON 1

l'adresse R du résultat est :

cas ① : l'adresse de (MRINT), et ce compteur régresse de 1

cas ② : α_1, et (MRINT) progresse de 11

cas ③ : α_2, et (MRINT) progresse de 11

cas ④ : α_1, et (MRINT) progresse de 23.

4. Expressions et instructions conditionnelles : SI ALORS SINON

- A l'identification, SINON est représenté SINON inst (cf chapitre II).

On peut rencontrer dans le programme source :

ⓐ SI A ALORS B SINON C

ou ⓑ SI A ALORS B ; FIN

avec A = Expression Booléenne

B = | Expression arithmétique simple } ⓐ
| Expression booléenne simple
| Expression de désignation
| Instruction inconditionnelle } ⓐ ou ⓑ
| Instruction Pour } ⓑ

C = | Expression arithmétique
| Expression booléenne
| Expression de désignation
| Instruction

Les Σ' sont vides.

1°) Σ (SI) est référencé (SI)

a) on construit le Γ :

SI Γ A ALORS ALLERA S1

S1 est l'adresse : du "calcul" de C dans le cas ⓐ , de l'instruction suivante dans le cas ⓑ .

L'adresse S1 n'est pas connue au moment de cette construction. On procéde alors par adressage indirect : on réserve pour S1 une zone de 5 positions dans (REMAN) ; c'est l'adresse de cette zone, surmontée d'un "flag", qui entre dans le Γ précédent. De plus, on envoie cette adresse sur la pile V. L'adresse réelle S1, qui sera connue au Σ de ALORS , sera perforée dans ce Σ sous la forme "carte de chargement" à l'adresse indiquée sur V.

b) on envoie un séparateur $\bar{9}9999$ sur la pile V.

c) on "libére" la position occupée par (α_2) si cet opérande est un Résultat intermédiaire.

2°) Σ (ALORS) est référencé (ORS).

On doit distinguer les 5 règles.

a) On reconnait une instruction conditionnelle si :

α) S (E) = FIN ou "point virgule".

Alors, on perfore S1 et on supprime le séparateur $\bar{9}9999$ dans la pile V.

β) S (E) = SINON $_{\text{inst.}}$ et S (V) = $\bar{9}9999$.

-on perfore S1

-on construit le Γ :

ALLERA S2

S2 est l'adresse de l'instruction suivante.

On procéde comme pour S1 (adressage indirect)

- on substitue S2 à S1 dans la pile V
- on supprime le séparateur $\bar{9}9999$ dans V.

b) on reconnait une expression conditionnelle dans tout autre cas.

Alors :

α) on transforme SINON $_{\text{inst.}}$ en SINON $_{\text{exp}}$ dans E

β) on construit les ordres de transfert de l'opérande (α_2) en "Résultat intermédiaire", s'il n'y est déjà. La zone qu'occupe cet opérande dans (MRINT) n'est pas réservée , ou est libérée. Cette construction dépend de la règle :

- Expression booléenne simple : on construit le transfert du digit (α_2)
- Expression arithmétique simple : on construit le transfert de la zone (α_2). De plus, si (α_2) est ENTIER , on construit des

ordres de conversion au type REEL

- Expression de désignation : α_2 est une étiquette. On construit alors le "transfert immédiat" de l'adresse α_2 et du niveau de procédure correspondant. Ce "niveau" se trouve dans la table. (cf chapitre III, 1ère lecture). On utilise donc 7 positions dans la zone des résultats intermédiaires.

γ) on construit :

ALLERA S2 (cf partie β de a))

et on substitue S 2 à S1 dans V.

δ) on supprime le séparateur 99999 dans V.

3°) Σ (SINON$_{inst.}$) est référencé (SNINS).

on perfore S 2 et on le supprime dans V

4°) Σ (SINON$_{exp.}$) est référencé (SNEXP).

α) on construit les mêmes transferts que dans la partie β du Σ (ALORS). La zone occupée par α_2 dans (MRINT) est cette fois réservée.

Ainsi, l'adresse où on transfère est la même pour les Γ de ALORS et SINON$_{exp.}$. C'est une adresse de (MRINT). C'est l'adresse R du résultat de l'expression conditionnelle.

β) on perfore S 2, et on le supprime dans V

γ) on envoie en S (V) l'adresse R

Nota. L'ordinogramme 3 décrit les Σ de ALORS, SINON$_{inst.}$ et SINON$_{exp.}$

5. Instruction Pour : POUR FAIRE PAS JUSQUA TANTQUE "Virgule" :=

Notations : on appelle :

- INS1 la première zone de (MRINT) disponible lorsqu'on entre dans une instruction POUR . Cette zone occupe 5 positions
- INS 2 la zone formée des 5 positions suivantes
- TR 1 la zone formée des 12 positions suivantes
- TR 2 la zone formée des 12 positions suivantes
- S1 l'adresse de l'instruction d'affectation de la variable contrôlée.
- S'1 l'adresse de l'instruction qui suit cette affectation.
- S 2 l'adresse de l'instruction qui succède à l'instruction POUR.

S1, S'1 et S 2 sont des adresses indirectes. Elles occupent chacune 5 positions de la zone réservée aux Rémanents.

Certains Σ et Σ' doivent distinguer les règles (cf chapitre II). Ces distinctions consistent à déterminer pour chaque élément de liste, s'il est le dernier élément de la liste ou non, c'est-à-dire s'il est suivi de FAIRE ou de "virgule". Nous schématisons ici les cas pouvant se présenter pour chaque forme, à l'aide d'exemples. Ces exemples ont été choisis pour faciliter l'enoncé des Σ et Σ', mais ne limitent pas les possibilités de formation d'une liste de POUR à partir d'un nombre quelconque d'éléments de liste de la forme 1, 2 ou 3 pris dans un ordre quelconque.

Notations: dans les schémas : a) Seuls les ordres construits pour le programme objet ne figurent pas entre crochets.

b) les références du programme objet, utilisées dans des constructions sans adressage indirect, sont écrites E_i et sont suivies de "deux points".

1°) Forme 1 :

POUR X := X1 , X2 FAIRE S

X1 et X2 sont des expressions arithmétiques

S est une instruction

- Réservation de INS1 et INS2
- X et son indicatif stockés dans (PX)
- S1 et S'1 réservés dans (REMAN)
- S1 substitué à X dans V
- 6 positions de V sont réservées à E'_1 (Forme 2)
- TR1 envoyé en S (V) avec le type de X
- POUR supprimé

INS1 := E_0

} Σ(POUR)

E_0 : calcul de X1

TR1 := X1
INS1 := E_1
ALLERA S1

- TR1 non réservé
- "Virgule" n'entre pas dans 0

} Σ'(:= ,)

E_1 : calcul de X2

TR1 := X2
INS1 := S2

- S1 et S'1 perforées
- S2 substitué à S1 dans V

} [1]

S1 : X := TR1 } [2]

} Σ(:=)

S'1 : "calcul" de S

ALLERA INS1

- S2 perforé. INS1 et INS2 libérés
- ainsi que la zone occupée par X
- si X est un tableau

} Σ(FAIRE)

Forme 2 :

POUR X := A PAS B JUSQUA C, D PAS E JUSQUA F FAIRE S

A, B, C, D, E et F sont des expressions arithmétiques.

cf Forme 1 } Σ(POUR)

E_0 : calcul de A

X := A
INS1 := E_1

- A supprimé dans V

} Σ'(:= PAS)

E_1 : calcul de B

TR 1 : = B
[TR1 réservé dans (MRINT]

} Σ(PAS)

Calcul de C

TR 2 := C
INS 2 := E_2
E'_1 : $SAUT_1$
X := X + TR 1
$SAUT_2$
Suppression $SAUT_1$
SI (TR 2 - X) x TR1 < 0 ALORS
DEBUT 'établissement' de SAUT 1 ; INS1 := INS2 ;
ALLERA INS2 FIN SINON ALLERA S'1
[E'_1 placé dans V ; TR1 libéré ;]
["Virgule" n'entre pas dans 0]

} Σ(JUSQUA)
S (E)="virg

E_2 : Calcul de D

X := D
INS 1 := E_3
[D supprimé dans V]

} Σ'(:= PAS)

E_3 : Calcul de E

TR 1 := E
[TR1 réservé dans (MRINT)]

} Σ(PAS)

Calcul de F

TR 2 := F
INS 2 := S 2
ALLERA E'_1
(E'_1 se trouve dans V)
[TR1 libéré ; Perforation de S1 et S'1 ;
S 2 entre dans V]

} Σ(JUSQUA)
S(E) = FAIRE

S 1 : X := TR1

} 2 Σ(:=)

S'1 : "calcul de S

cf Forme 1

} Σ(FAIRE)

Forme 3 :

POUR X := A TANTQUE B, C TANTQUE D FAIRE S

A et C sont des expressions arithmétiques
B et D sont des expressions booléennes.

cf Forme 1

} Σ(POUR)

E_0 : Calcul de A

TR 1 := A
[TR1 réservé dans (MRINT)]

} Σ'(:= TANTQUE)

Calcul de B

SI B ALORS ALLERA S1 SINON INS1 := E_1
[TR1 est libéré ; "Virgule" n'entre pas dans 0]

Σ(TANTQUE)
S (E)= "Virgul

E_1 : calcul de C

TR1 := C
[TR1 réservé dans MRINT]

Σ'(:= TANTQUE)

Calcul de D

SI D ALORS ALLERA S1 SINON
ALLERA S2
[TR1 est libéré ; S1 et S'1 sont perforés ;
S2 entre dans V]

Σ(TANTQUE)
S (E) = FAIRE

S1 : X := TR1

2 Σ(:=)

S'1 : "calcul" de S

cf Forme 1

Σ(FAIRE)

Remarque :

L'organigramme 4 décrit :

- Σ(PAS) , référencé (PAS)
- Σ(JUSQUA), référencé (JUSQA)
- Σ(TANTQUE), référencé (TANT)
- La partie 1 de Σ(:=), référencée (VIRG1).
- Les Σ' de (:= "virgule"), (:= PAS) et (:= TANTQUE), référencés également (VIRG1).

6. Instruction d'affectation : :=

Les Σ' sont vides, à l'éxception du Σ'(AIGUILLAGE :=) de la déclaration d'aiguillage (§ 11)

Σ(:=) est référencé (AFF). Il comprend deux parties :

a) La partie 1 est décrite au § 5 (Forme 1)

b) La partie 2 correspond à toutes les autres règles :

α) on construit le Γ' :

$\Phi_1 := (\alpha_2)$

α_1 et α_2 sont : - ou tous les deux BOOLEEN (transfert de digit)
- ou ENTIER ou REEL (transfert de zone) ;

si les types sont différents, on génére les ordres de conversion de α_2 au type de α_1 .

β) on supprime α_1 et α_2 dans V, sauf si S (0) -1 est un nouveau symbole d'affectation, auquel cas α_1 subsiste.

En outre, si α_2 est un résultat intermédiaire, on libère la position ou la zone qu'il occupe.

7. Instruction ALLER A : ALLER A

Les Σ' sont vides et la règle est unique pour le Σ.

Σ (ALLERA) est référencé (ALEA) :

on construit le Γ' :

SI "non Flag" ALORS ALLERA S1 ; ALLERA α_2 ; S1 "poser Flag" ;

α_2 est une étiquette, adressée directement ou non, et peut être une zone des résultats intermédiaires. On libère en ce cas les 7 positions occupées (cf Σ (ALORS))

Les deux instructions qui "encadrent" l'instruction "ALLERA α_2" permettent à un sous-programme d'exécution de supprimer le "saut" dans certains cas et quand α_2 est un indicateur d'aiguillage (cf § 11).

Remarque :

On utilise l'adresse Q d'une instruction ALLERA pour indiquer la différence entre le niveau de procédure de la référence où on saute et le niveau actuel. Ce renseignement est utilisé par les procédures [8] ;

8. Instruction - Instruction composée - Bloc : ; DEBUT

Seul le Σ de DEBUT de bloc présente des particularités : Dans ce Σ, référencé (SFIN1), on force les compteurs (MRINT) et (CPTR2) aux adres

qui se trouvent en S (V) et en S (V) -1. Celles-ci ont été envoyées dans V à l'identification du même DEBUT (cf chapitre III, § 2).

9. Déclaration de type :

1°) Les Σ' de (REEL ,) , (ENTIER ,) et (BOOLEEN ,), ainsi que les Σ de REEL , ENTIER et BOOLEEN permettent la formation de la table des identificateurs. Les opérations effectuées sont décrites dans la partie "formation de la table" (Chap. III, §2).

2°) A REMANENT ne correspond aucun Σ et les Σ' sont vides (cf chapitre II). Ce symbole, lorsqu'il est en S (0) -1, permet de substituer le compteur d'affectation (REMAN) au compteur d'affectation (MRINT) dans les Σ et Σ' précédents.

10. Indicateur de fonction :

Les Σ et Σ' sont décrits dans [8] pour les procédures, et dans [5] pour les fonctions standard.

11. Déclaration et indicateur d'aiguillage :

1°) Déclaration d'aiguillage :

a) Σ' (AIGUILLAGE :=) est référencé (AFAIG).

α) On construit le Γ' :

ALLERA S

qui permettra, à l'exécution, de "sauter" la déclaration d'aiguillage. S est une adresse indirecte; c'est l'adresse correspondant à

l'identificateur d'aiguillage dans la table (cf chapitre III).

β) L'adresse dans le programme objet du calcul de la première expression de désignation de la liste d'aiguillage est envoyée en S (V).

- Le symbole (:=) ne rentre pas dans la pile 0 (cf chapitre II)

b) Σ'(AIGUILLAGE ,) est référencé (VGAIG) :

α) on construit un Γ qui comprend :

- Le transfert du résultat de l'expression de désignation en Résultat intermédiaire, sans réservation dans (MRINT).
- Un saut à un sous-programme de "fin d'exécution de procédures" [8].

β) - L'adresse, dans le programme objet, du calcul de l'expression de désignation suivante est envoyée en S (V).

- La "Virgule" n'entre pas dans la pile 0 (cf chapitre III)

c) Σ(AIGUILLAGE ;) est référencé (AIG) :

α) On construit un Γ qui comprend :

- La suite des adresse stockées dans V par les Σ' précédents.
- Le nombre d'expressions de désignation qui forment la liste d'aiguillage.

β) On "perfore" l'adresse S (cf § 4)

- on supprime dans V les adresses stockées et l'adresse S.

2°) Indicateur d'aiguillage :

Σ([ ) est référencé (1CROCH).

α) On construit un Γ qui comprend :

- Le transfert à un sous-programme d'exécution de l' adresse S de l'identificateur d'aiguillage.
- Le transfert au même sous-programme du résultat du calcul d'indice.
- Une séquence qui correspond à un "appel de procédure". Ces ordres sont décrits dans [8] .

Remarque :

Le sous-programme d'exécution mentionné ici calcule, à partir des renseignements reçus, l'adresse de l'expression de désignation qui correspond à la valeur de l'indice. Quand cette valeur est négative, ou supérieure au nombre d'expressions de désignation qui forment la liste d'aiguillage, le sous-programme neutralise, à l'aide d'une position flag, l'instruction ALLERA qui utilise cet indicateur d'Aiguillage (cf § 7).

β) - On supprime dans la pile V l'adresse de l'identificateur d'aiguillage ainsi que l'adresse R du résultat du calcul d'indice.

- on envoie en S (V) le contenu de (MRINT) avec l'indicatif I = $\bar{4}$. Ce compteur régresse de 7.

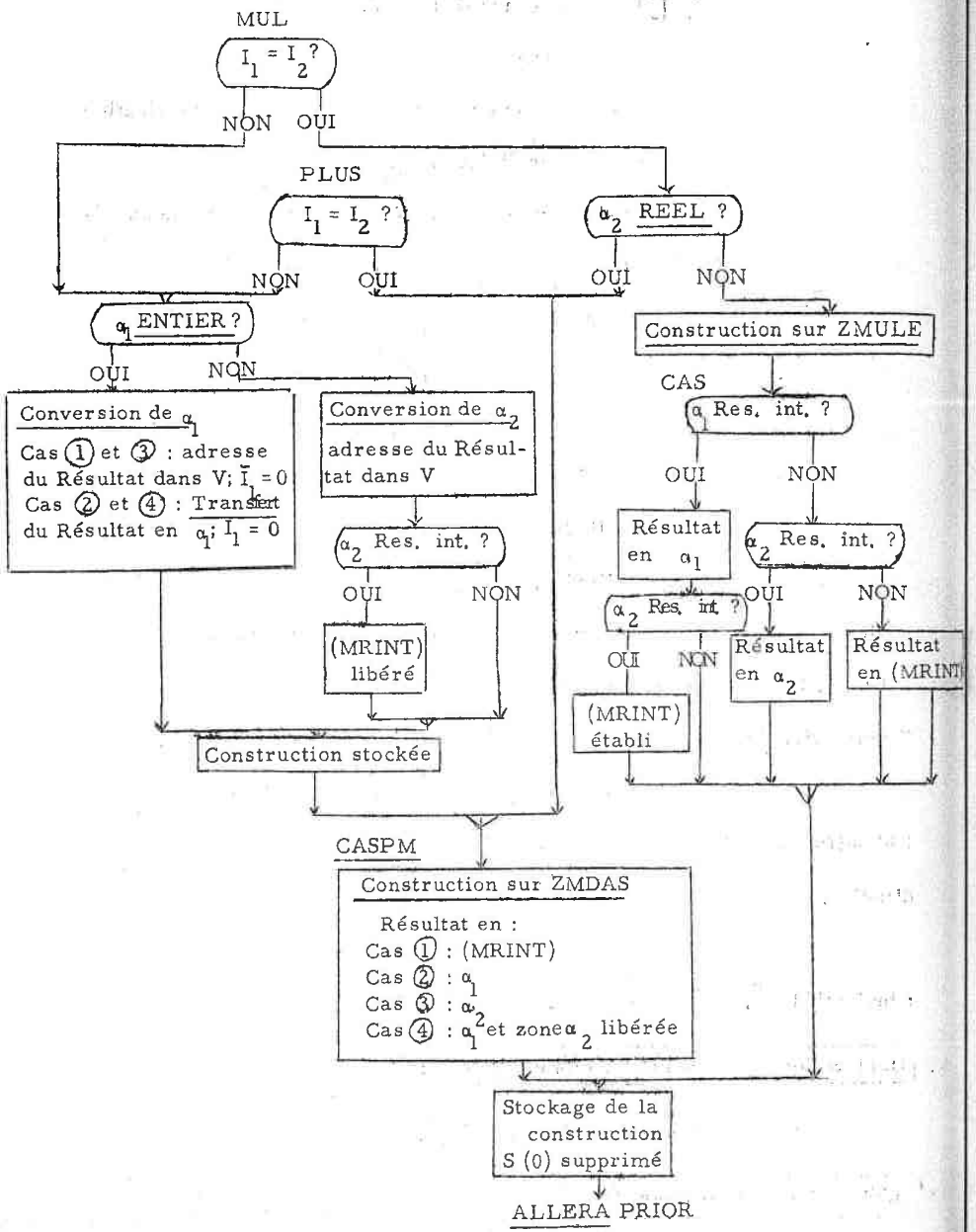
12. Déclaration de Tableau et Variable indicée :

Les Σ et Σ' sont décrits dans [9] .

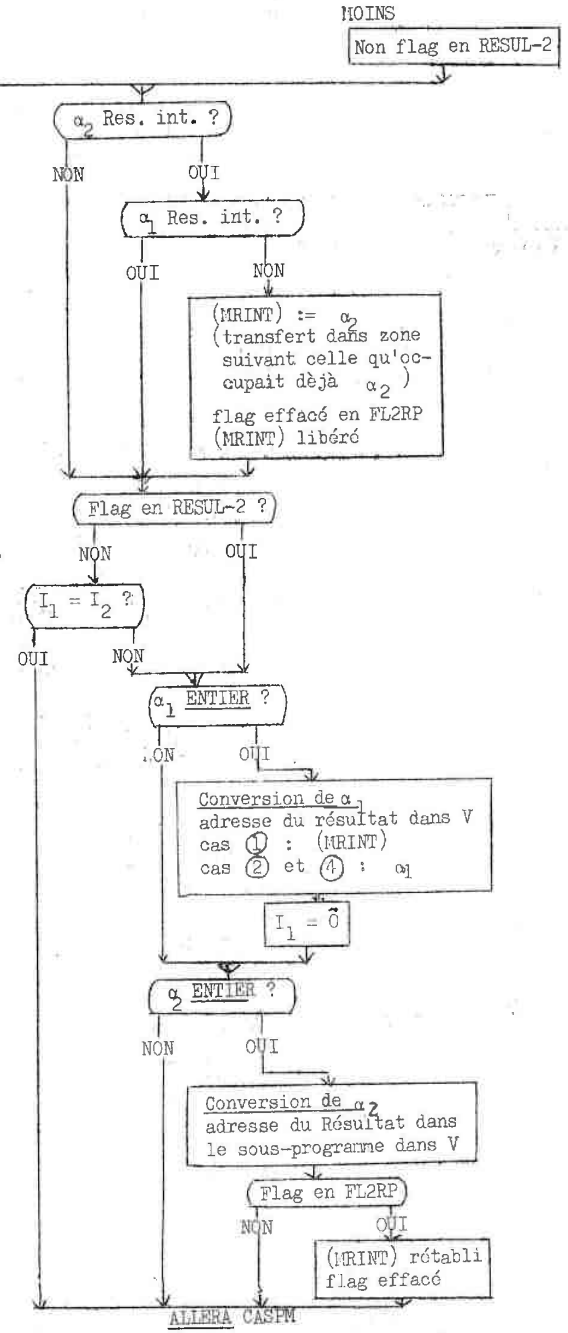
13. Déclaration de Procédure :

Les Σ et Σ' sont décrits dans [8] .

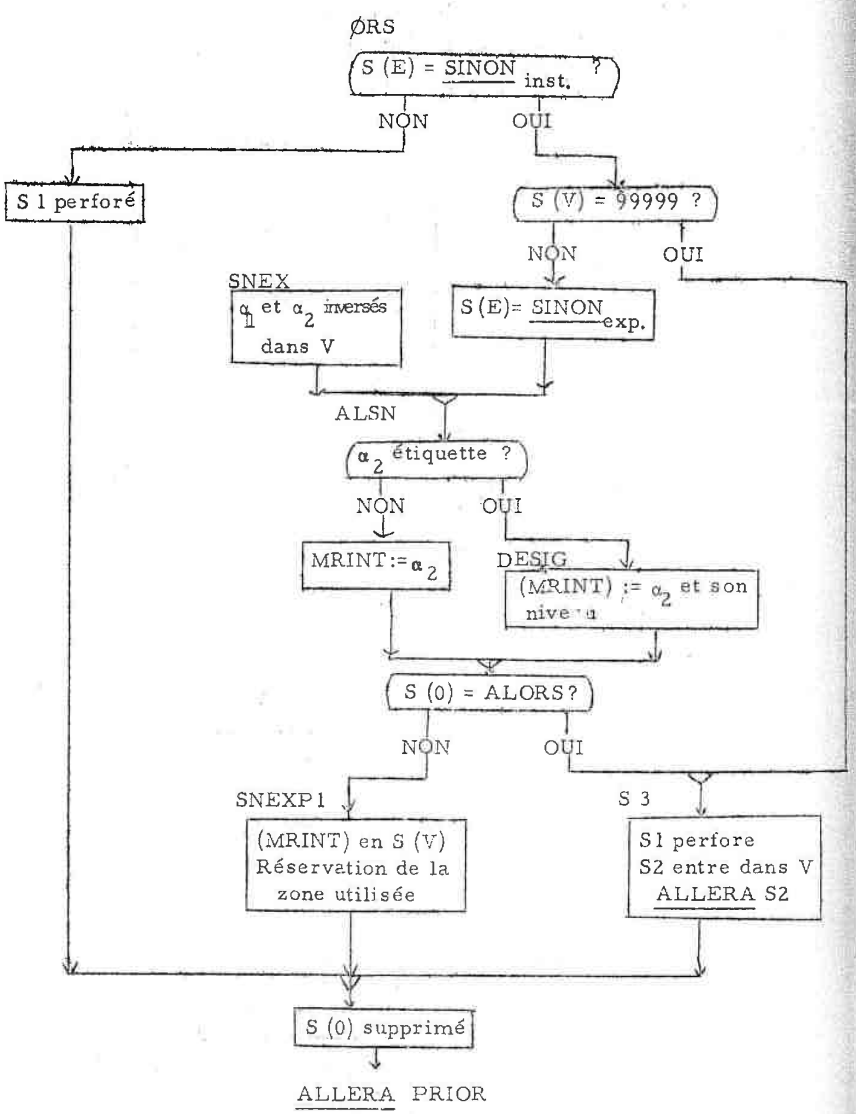
ORDINOGRAMME 1



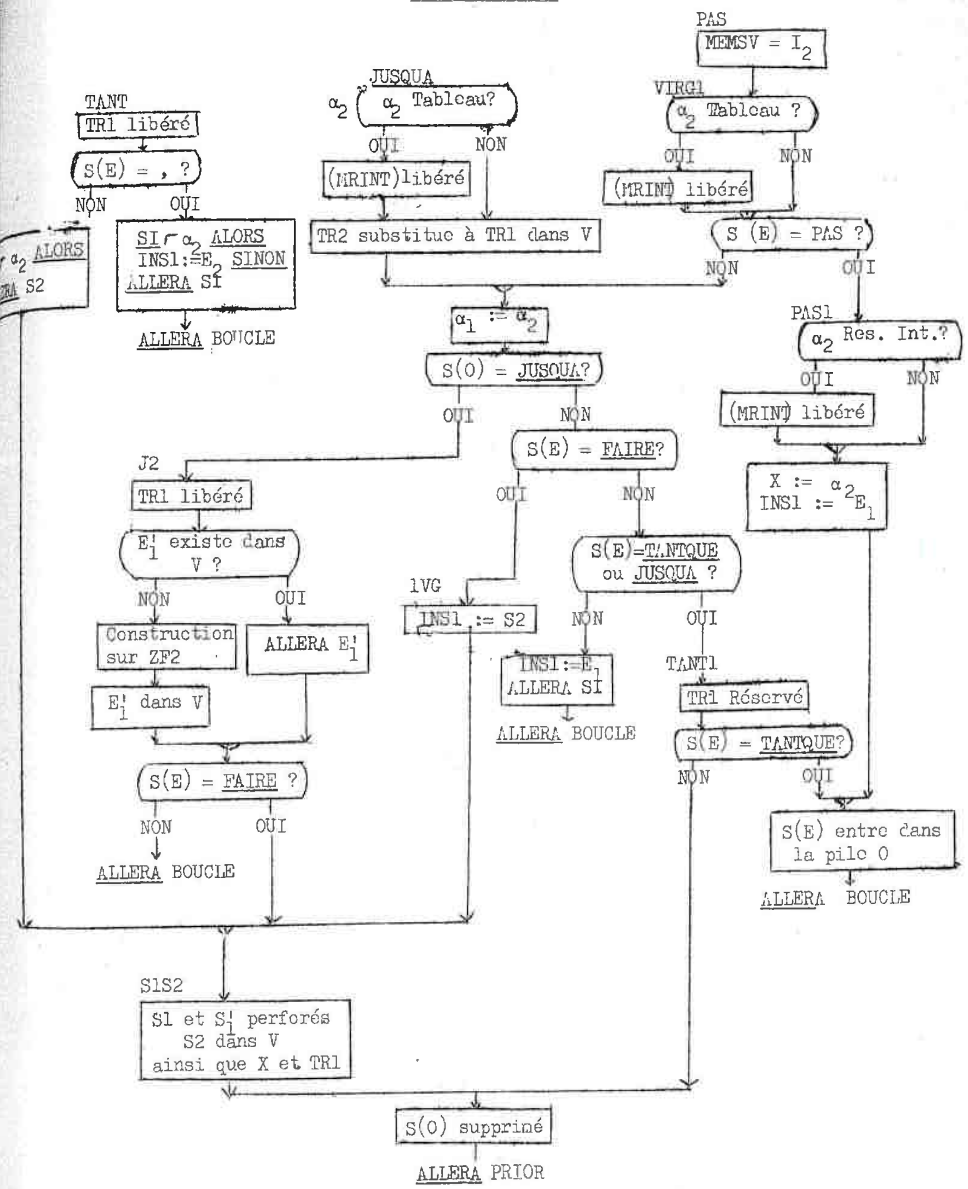
ORDINOGRAMME 2



ORDINOGRAMME 3



ORDINOGRAMME 4



ANNEXE

NOTA : Les signes + du programme sont représentés par &.

```
*  DECLARATIONS
*
   MEMSF DS  5
   MEMSO DS  5
   MEMSV DS  5
   MATR  DS  5
   MPGOB DS  5
   MRINT DS  5
   CHOB  DS  5
   CALADRDS  5
   RESUL DS  5
   CTSOR DS  2
   PX    DS  6
   REMAN DS  5
         DORG*&50
   SE    DAS 80
         DAC 1,@
   SO    DSS 200
   SV    DS  1000
   C3    DSS 260
   C4    DSS 200
   C1    DSS 81
   FLEG  DS  1
   FLR2  DS  1
   FLB2  DS  1
   FL1RP DS  1
   FL2RP DS  1
   FL2T  DS  1
   MRINT1DS  5
   CO    DS  2
   MEMI1 DS  2
   MEMI2 DS  2
   MEMK1 DS  2
   MEMK2 DS  2
   MEMJ1 DS  2
   MEMJ2 DS  2
   ZONE  DS  2
   CPTR  DS  5
   CPTR2 DS  5
   CPTR3 DS  5
   CP1   DS  2
   ATENT1DS  14
   ATENT2DS  14
   CTNIV DS  2
   LC1   DS  5
   LC2   DS  5
   LTAB  DS  5
   LSYMB DS  2
   IDPRO DS  5
   ADER  DS  3
   TYPE  DS  2
   TPRIM DS  2
   LTAB1 DS  5
   CT    DS  5
   SPCOV DS  ,00600
   C5    DS  ,20000
   C12   DS  ,40000
   C13   DS  ,49999
   C14   DS  ,45000
   C15   DS  ,00403
```

```
C16    DS   ,00404
ADIN1  DS   ,00415
ADIN2  DS   ,00420
TABETIDS   ,56000
CAIG   DS   ,90000
       DORGC13-307
       DC   14,48415363450000
       DC   8,69924000
       DC   14,55564441634100
       DC   8,69929000
       DC   14,43534546000000
       DC   8,69934000
       DC   14,41426200000000
       DC   8,69939000
       DC   14,62494755450000
       DC   8,69944000
       DC   14,45556349455900
       DC   8,69949000
       DC   14,59414372000000
       DC   8,69954000
       DC   14,53564700000000
       DC   8,69959000
       DC   14,45675700000000
       DC   8,69964000
       DC   14,41594363470000
       DC   8,69969000
       DC   14,43566200000000
       DC   8,69979000
       DC   14,62495500000000
       DC   8,69974000
       DC   14,62565963495900
       DC   8,70000000
       DC   14,45556359455900
       DC   8,70001000
       DORGCT&1
C17    DSC 23,360050000500250000110057
       DC   7,3310057
       DC   12,800500250057
       DC   43,300011470000000090048000000000049200000000000@
LC3    DC   1,@
PR1    DSC 6,-080090
       DC   6,100000
       DC   6,000000
       DC   9,90000900@
PR2    DSC 4,8009
       DC   3,3@
PR3    DSC 3,-800
       DSC 2,-90
       DSC 5,50000
       DC   16,0000000000000000@
ZADR   DSC 13,160000000000@
ZP1    DSC 21,160000000000049000000@
ZF2    DSC 42,490000000000210000000000490000001600000000
       DC   48,412200000000002300000000046000000130016000000000
       DC   23,492600000000049000000@
ZPREP1DSC 13,260000000000@
ZCOVR  DSC 25,270100000000060000000600@
ZEXPO  DSC 49,260157000000260158200000170158800000060000001582
ZMDAS  DSC 13,020000000000@
ZMULE  DSC 37,230000000000320008800000260000000099@
```



```
ZDIVF  DSC 7,1600079
       DC   42,280009900000290008800000260000000087@
ZET    DSC 48,250048700000250048500000230048700485210048700485
       DSC 37,320009800000220048700099250000000487@
ZOU    DSC 49,250048700000250048500000230048700485250000000099@
ZIMP   DSC 48,250048700000250048500000240048700485460000001300
       DSC 33,150000000001490000001500000000000@
ZSSE   DSC 45,440000000000330000000004900000032000000000@
ZNON   DSC 49,150048700001250048500000220048700485250000000487@
ZAFF1  DSC 25,070100000000260000000600@
ZDEB   DSC 9,49100000@
ZSI    DSC 13,430000000000@
ZORS   DSC 9,49000000@
ZRELA  DSC 36,240000000000000000000000150000000001
       DSC 21,490000001500000000000@
ZALEA  DSC 19,440000000024900000
       DC   3,0
       DC   15,003200402000000@
ZETI1  DSC 25,160000000000160000000000@
ZAIG1  DSC 25,260000000000260000000000@
ZAIG2  DSC 37,269999900000269999900000179999900000@
CAIG1  DSC 9,49111110@
C2     DNB 40
       DNB 40
       DC   1,@
C7     DC   2,0.
C8     DC   2,0
CFIN   DC   8,46495503
CSN    DC   12,624955565503
CGUF   DC   10,0347644603
C9     DSC 46,111102200000000000000222321102121122222221121?
       DSC 46,011102200000000000000222321102121122222221121?
       DSC 46,001102200000000000000222321102121122222221121?
       DSC 46,000102200000000000000222321102121122222221121?
       DSC 46,111122200000000000000222321102121122222221121?
       DSC 46,000000200000000000000222321002222222222222223
       DSC 46,000000200000000000000222322002222122222212223
       DSC 46,111122111111111111122113211021211122222221111?
       DSC 46,111122101111111111122113211021211122222221111?
       DSC 46,111122101111111111122113211021211122222221111?
       DSC 46,111122100011011111122113211021211122222221111?
       DSC 46,111122100011011111122113211021211122222221111?
       DSC 46,111122101111111111122113211021211122222221111?
       DSC 46,111122200000022222202223211021211222222221121?
       DSC 46,111122200000022222202223211021211222222221121?
       DSC 46,111122200000022222202223211021211222222221121?
       DSC 46,111122200000022222202223211021211522222221121?
       DSC 46,111122200000022222202223211021211522222221121?
       DSC 46,111122100011011111122113211021211122222221111?
       DSC 46,222222222222222222222231222222222222222222222
       DSC 46,222220200000022222202213220022222222222222223
       DSC 46,222220200000022222202223220022221222222212223
       DSC 46,222220200000022222202223220022222322222232123
       DSC 46,000000300000000000002303020021211222222231223
       DSC 46,000002200000000000000223021001010222222221223
       DSC 46,000000100000000000002113210021211122222211113
       DSC 46,000000200000000000002223220022222222222232213
       DSC 46,222220222222222222222232200?121222222222221222
       DSC 46,222220222222222222202230220010002000000021222
```

```
        DSC 46,22222022222222222222022302200000020000000212222
        DSC 46,22222022222222222222022302200100022222222212222
        DSC 46,22222022222222222222022302200100022222222212222
        DSC 46,22222020000002222220222322002222222222222212123
        DSC 46,22222022222222222222222332022122222222222232222
        DSC 46,22222222222222222222222222221222222222200322222
        DSC 46,22222222222222222222222222221222222222200322222
        DSC 46,22222222222222222222222222221222222222200322222
        DSC 46,22222222222222222222222222222222222220002222222
        DSC 46,22222222222222222222222322222122222222222322222
        DSC 46,22222222222222222222222222202122222222222222222
        DC  2,03
C10     DS  ,*-1
        DC  2,70
        DC  2,04
        DC  2,45
        DC  2,10
        DC  2,11
        DC  2,13
        DC  2,08
        DC  2,14
        DC  2,09
        DC  2,20
        DC  2,12
        DC  2,21
        DC  2,10
        DC  2,23
        DC  2,42
        DC  2,24
        DC  2,28
        DC  2,33
        DC  2,16
        DC  6,414947
C11     DS  ,*-5
        DC  2,16
        DC  2,06
        DC  2,35
        DC  6,415353
        DC  2,08
        DC  2,00
        DC  2,29
        DC  6,415356
        DC  2,06
        DC  2,00
        DC  2,26
        DC  6,425656
        DC  2,10
        DC  2,00
        DC  2,38
        DC  6,434941
        DC  2,08
        DC  2,00
        DC  2,48
        DC  6,435654
        DC  2,18
        DC  2,08
        DC  2,52
        DC  6,444542
        DC  2,06
        DC  2,01
```



```
        DC  2,31
        DC  6,444946
        DC  2,02
        DC  2,00
        DC  2,19
        DC  6,444965
        DC  2,08
        DC  2,00
        DC  2,13
        DC  6,455563
        DC  2,08
        DC  2,00
        DC  2,37
        DC  6,455864
        DC  2,04
        DC  2,00
        DC  2,04
        DC  6,456303
        DC  2,00
        DC  2,00
        DC  2,01
        DC  6,436549
        DC  2,14
        DC  2,00
        DC  2,47
        DC  6,464149
        DC  2,06
        DC  2,00
        DC  2,33
        DC  6,464164
        DC  2,04
        DC  2,03
        DC  2,00
        DC  6,464955
        DC  2,02
        DC  2,02
        DC  2,43
        DC  6,476446
        DC  2,02
        DC  2,00
        DC  2,50
        DC  6,476456
        DC  2,02
        DC  2,07
        DC  2,49
        DC  6,495457
        DC  2,06
        DC  2,00
        DC  2,03
        DC  6,495546
        DC  2,02
        DC  2,00
        DC  2,14
        DC  6,495547
        DC  2,02
        DC  2,00
        DC  2,15
        DC  6,516462
        DC  2,08
        DC  2,00
```

```
DC  2,23
DC  6,555655
DC  2,02
DC  2,00
DC  2,05
DC  6,565962
DC  2,02
DC  2,00
DC  2,00
DC  6,566403
DC  2,00
DC  2,00
DC  2,02
DC  6,574162
DC  2,02
DC  2,00
DC  2,22
DC  6,575664
DC  2,04
DC  2,00
DC  2,21
DC  6,575956
DC  2,14
DC  2,05
DC  2,41
DC  6,594545
DC  2,04
DC  2,00
DC  2,36
DC  6,594554
DC  2,12
DC  2,00
DC  2,39
DC  6,624903
DC  2,00
DC  2,00
DC  2,06
DC  6,624955
DC  2,06
DC  2,00
DC  2,32
DC  6,625747
DC  2,02
DC  2,00
DC  2,17
DC  6,626457
DC  2,02
DC  2,00
DC  2,18
DC  6,634142
DC  2,10
DC  2,00
DC  2,40
DC  6,634155
DC  2,10
DC  2,00
DC  2,07
DC  6,654153
DC  2,08
DC  2,00
```



```
DC  2,51
DC  6,655941
DC  2,04
DC  2,04
DC  2,00
```

```
*  PREMIERE LECTURE
*
      BEGIN SF  ATENT2-13
            TFM CTNIV,0,10
            TFM LTAB,TABETI
            CF  LPTVG
            TFM LC1,0
            TFM LC2,0
            TD  LTAB,LC3,6
            SF  SSP7
            TFM MEMSO,SO-1
            TFM MEMSE,SE-2
            SF  SE-3
            TFM *&18,SE-1
            CF  0
            AM  *-6,1,7
            CM  *-18,SE&159
            BNF *-36
            CF  DKL
            RACDSE
            CF  DECAL
            BLC *&24
            SF  DECAL
      DKL   BTM DECAL,0,10
            CM  MEMSE,40,610
            BNH 1DKL
            BNF *&60,DKL
            SM  MEMSE,1,10
            CF  MEMSE,,6
            AM  MEMSE,1,10
            B   DKL
            SF  DKL
            B   DKL
      1DKL  CF  DKL
            CM  MEMSE,3,610
            BNE DKL
            SM  MEMSE,1,10
            CF  MEMSE,,6
            AM  MEMSE,1,10
            BTM DECAL,0,10
            CM  MEMSE,69,610
            BNL DKL
            CM  MEMSE,40,610
            BH  TABL3
            CM  MEMSE,4,610
            BE  LCRE
            CM  MEMSE,23,610
            BE  LPTVG
            CM  MEMSE,24,610
            BE  LCRO
            CM  MEMSE,3,610
            BNE DKL
            SM  MEMSE,1,10
            CF  MEMSE,,6
            AM  MEMSE,1,10
            BTM DECAL,0,10
            CM  MEMSE,24,610
            BE  DKL
            SM  MEMSE,2,10
            B   LDP
```



```
      TABL3 BTM SSP3,0,10
            BD  *&48,99,11
            AM  99,2,10
            TF  LSYMB,99,11
            B   DKL
            CM  99,8,610
            BNE *&24
            BTM LCTAIR,DKL
            CM  99,2,610
            BH  *&60
            CF  SEPAR
            BE  3SEPAR
            SF  SEPAR
            B   2SEPAR
            CM  99,5,610
            BE  LPROC
            CM  99,6,610
            BE  LAIG
            CM  99,7,610
            BNE DKL
            BTM DECAL,0,10
            CM  MEMSE,3,610
            BNE *-24
            AM  MEMSE,8,10
            C   MEMSE,CGUF,6
            BE  DKL
            SM  MEMSE,8,10
            B   *-84
      SEPAR AM  LTAB,5,10
            CM  LC2,0,7
            BE  1SEPAR
            AM  LTAB,5,10
            TF  LC2,LTAB,6
            AM  LC2,10,10
            TF  LC2,LTAB,6
            S   LC2,LC2,6
            SM  LC2,17,610
            SM  LTAB,5,10
      1SEPARTF LC2,LTAB
            AM  LTAB,5,10
            BNF LFIN,SEPAR
            B   LDEB
      LFIN  SM  MEMSO,5,10
            CM  MEMSO,31,610
            BE  *&24
            BTM IMPER,38,9
            AM  MEMSO,5,10
            SF  MEMSO
            TF  MEMSO,LC2,6
            CF  MEMSO
            SM  MEMSO,7,10
            TFM LTAB,0,67
            AM  LTAB,12,10
            CM  MEMSO,SO-1
            BNE 1LFIN
            BNF *&36,DECAL
            RACDSE
            BNLC*-12
            H
            B   DEBUT
```

```
1LFIN BNF *&36,LPTVG
      CF  LPTVG
      B   DKL
      BTM DECAL,0,10
      CM  MEMSE,3,610
      BNE 1LFIN
      BTM DECAL,0,10
      CM  MEMSE,23,610
      BE  LPTVG
      AM  MEMSE,6,10
      C   MEMSE,CFIN,6
      BE  3SEPAR
      AM  MEMSE,4,10
      C   MEMSE,CSN,6
      BE  DKL
      SM  MEMSE,12,10
      B   1LFIN
2SEPARBTM DECAL,0,10
      CM  MEMSE,3,610
      BE  *&36
      SM  MEMSE,2,10
      B   IC2
      BTM DECAL,0,10
      BTM SSP3,1,10
      AM  99,6,10
      SM  MEMSE,8,10
      CM  99,52,610
      BNE *&24
      BTM LCTAIR,2SEPAR
      CM  99,35,610
      BL  IC2
      CM  99,41,610
      BNH SEPAR
IC2   AM  MEMSO,2,10
      TFM MEMSO,31,610
      AM  MEMSO,5,10
      TFM MEMSO,88888,67
      B   DKL
      DORG*&6
LCTAIRBTM DECAL,0,10
      CM  MEMSE,3,610
      BNE LCTAIR
      BTM DECAL,0,10
      CM  MEMSE,23,610
      BNE LCTAIR
      B   LCTAIR-1,,6
3SEPARCM  MEMSO,88888,67
      BNE SEPAR
      SM  MEMSO,7,10
      B   1LFIN
LDEB  AM  MEMSO,2,10
      TFM MEMSO,31,610
      AM  MEMSO,5,10
      TF  MEMSO,LTAB,6
      CM  LC1,0,7
      BE  *&36
      TF  LC1,LTAB,6
      SM  LC1,5,610
      AM  LTAB,10,10
      TF  LC1,LTAB
```



```
      TFM LC1,0,67
      AM  LTAB,2,10
      B   DKL
LCRO  AM  MEMSO,2,10
      TFM MEMSO,24,610
      AM  MEMSO,5,10
      B   DKL
LCRF  SM  MEMSO,5,10
      CM  MEMSO,24,610
      BE  *&24
      BTM IMPER,39,9
      SM  MEMSO,2,10
      B   DKL
LPTVG TFM LSYMB,30,10
      SM  MEMSO,12,10
      CM  MEMSO,41,610
      BE  *&36
      AM  MEMSO,12,10
      B   DKL
      SM  CTNIV,1,10
      TFM MEMSO,31,610
      AM  MEMSO,12,10
      TF  *&35,MEMSO
      SM  MEMSO,7,10
      TF  MEMSO,0,6
      CF  SEPAR
      SF  LPTVG
      B   SEPAR
      DORG*&6
SSP7  S   ATENT2,ATENT2
      TFM CPTR,2,7
      BTM DECAL,0,10
      CM  MEMSE,40,610
      BL  *&36
      CM  MEMSE,69,610
      BNH *&24
      BTM IMPER,40,9
2SSP7 BTM DECAL,0,10
      CM  MEMSE,40,610
      BH  FORMA1
      SM  MEMSE,2,10
      TF  ATENT1,MEMSE,11
      AM  CPTR,ATENT1
      TF  CPTR,ATENT1,6
1SSP7 BNF SSP7-1,SSP7,6
      AM  LTAB,13,10
      TF  LTAB,ATENT2,6
      B   SSP7-1,,6
FORMA1SM  MEMSE,1,10
      CF  MEMSE,,6
      AM  MEMSE,1,10
      AM  CPTR,2,10
      CM  CPTR,14,7
      BNE 2SSP7
      TF  ATENT2,MEMSE,11
      BTM DECAL,0,10
      CM  MEMSE,40,610
      BH  *-24
      SM  MEMSE,2,10
      B   1SSP7
```

```
LDP    SM   MEMSO,5,10
       CM   MEMSO,24,610
       BNE  *&36
       AM   MEMSO,5,10
       B    DKL
       TDM  LDP&9,4,9
       SM   MEMSE,5,10
       AM   MEMSO,5,10
       BNF  *&36,MEMSE,11
       SM   MEMSE,1,10
       BTM  SSP7,1LDP
       SM   MEMSE,2,10
       B    *-48
1LDP   AM   MEMSE,4,10
       AM   LTAB,5,10
       TFM  LTAB,0,67
       AM   LTAB,1,10
       TD   LTAB,LDP&9,6
       AM   LTAB,2,10
       TF   LTAB,CTNIV,6
       A    LTAB,LDP&8,6
       AM   LTAB,1,10
       TD   LTAB,LC3,6
       B    DKL
LAIG   TFM  LDP&9,14,9
       BTM  SSP7,1LDP
```



```
*  COMPILATION
*
     DEBUT RACDSE
           TFM  MEMSE,SE-2
           SF   SE-3
           TFM  MEMSO,SO-1
           TFM  MEMSV,SV
           TFM  MRINT,C12
           TFM  MPGOB,C3
           TFM  CHOB,C5
           TFM  CPTR2,C13
           TFM  CT,C13
           TR   FCFL,ZERO-20
           TDM  SV,2
           CF   S1S2
           TFM  CTSOR,0,10
           TFM  REMAN,C14
           TFM  CTNIV,00,10
           SF   ATENT2-13
           SF   C4-1
           CF   TBLEA
           TFM  T1,0,10
           TFM  N,0,10
           TFM  NP,0,10
           TFM  T2,0,10
           CF   JCROCH
           WNCDC17
           TFM  LTAB,TABETI
           BNR  *&24,LTAB,11
           B    *&84
           AM   CPTR2,1,10
           TR   CPTR2,LTAB,611
           AM   LTAB,22,10
           AM   CPTR2,22,10
           BNR  *-24,LTAB,11
           SM   CPTR2,1,10
           TF   LC1,LTAB
           AM   LC1,5,10
           CF   DECAL
           BLC  *&24
           SF   DECAL
           TFM  *&18,SE-1
           CF   SE
           AM   *-6,1,7
           CM   *-18,SE&159
           BNL  BOUCLE
           B    *-48
     BOUCLEBTM  DECAL,00,10
           CM   MEMSE,40,610
           BH   TEST1
           CM   MEMSE,03,610
           BNE  EXIST
           SM   MEMSE,1,10
           CF   MEMSE,,6
           AM   MEMSE,1,10
           BTM  DECAL,00,10
           CM   MEMSE,69,610
           BE   DEBU3
           BH   DEBU2
           CM   MEMSE,40,610,
```

```
        BH   TABLE
        BTM  SSP1
        TF   MEMSE,CO,6
        B    PRIOR
EXIST   BTM  SSP2,0,10
        CM   MEMSE,10,610
        BNE  *&36
        TFM  MEMSE,46,610
        B    PRIOR
        CM   MEMSE,20,610
        BNE  *&36
        TFM  MEMSE,20,610
        B    PRIOR
        AM   00099,2,10
        TF   MEMSE,00099,611
        B    PRIOR
TABLE   BTM  SSP3,0,10
        BD   *&48,99,11
        AM   00099,2,10
        TF   MEMSE,00099,611
        B    PRIOR
        BTM  SSP6
        B    *-48
TEST1   CM   MEMSE,69,610
        BH   DEBU1
        SF   SYMBOL
        S    ATENT2,ATENT2
        TFM  CPTR,2,7
        BTM  DECAL,00,10
        CM   MEMSE,40,610
        BH   FORMAT
        SM   MEMSE,2
        TF   ATENT1,MEMSE,11
        AM   CPTR,ATENT1
        TF   CPTR,ATENT1,6
        B    SYMBOL
FORMATSM     MEMSE,01,10
        CF   MEMSE,,6
        AM   MEMSE,01,10
        AM   CPTR,2,10
        CM   CPTR,14,7
        BNE  TEST1&60
        TF   ATENT2,MEMSE,11
        BTM  DECAL,00,10
        CM   MEMSE,40,610
        BH   FORMAT&84
        SM   MEMSE,2,10
        B    SYMBOL
COMMU   CM   MEMSO,35,610
        BNH  CONSUL
        CM   MEMSO,41,610
        BH   CONSUL
        CM   CTNIV,0,10
        BNE  PDCL
        AM   CPTR2,14,10
        TF   CPTR2,ATENT2,6
        AM   CPTR2,6,10
        SM   MEMSV,6,10
        TF   MEMSV,CPTR2,6
        AM   CPTR2,2,10
```



```
        B    PRIOR
SYMBOLBTM    DECAL,00,10
        CM   MEMSE,03,610
        BNE  EXIST2
        SM   MEMSE,1,10
        CF   MEMSE,,6
        AM   MEMSE,1,10
        BTM  DECAL,00,10
        CM   MEMSE,40,610
        BH   TABLE2
        BTM  SSP1
        TF   MEMSE,CO,6
        BNE  PRIOR,SYMBOL
        B    COMMU
EXIST2BTM    SSP2,0,10
        AM   00099,2,10
        TF   MEMSE,00099,611
        BNE  PRIOR,SYMBOL
        B    COMMU
TABLE2BTM    SSP3,0,10
        BD   *&60,00099,11
        AM   00099,2,10
        TF   MEMSE,00099,611
        BNE  PRIOR,SYMBOL
        B    COMMU
        BTM  SSP6
        B    *-60
        DORG*&6
SSP1    CM   MEMSE,23,610
        BNE  *&36
        TFM  CO,30,10
        BB
        CM   MEMSE,33,610
        BNE  *&36
        TFM  CO,25,10
        BB
        CM   MEMSE,3,610
        BNE  *&36
        TFM  CO,34,10
        BB
        CM   MEMSE,24,610
        BNE  *&36
        TFM  CO,24,10
        BB
        CM   MEMSE,4,610
        BNE  ERSY4
        TFM  CO,44,10
        BB
        DORG*&6
SSP2    TFM  MEMI1,1,10
        TFM  MEMK1,10,10
        TFM  CP1,0,10
BCL2    AM   CP1,1,10
        CM   CP1,08,10
        BH   ERSY4
        LD   00099,MEMI1
        A    00099,MEMK1
        DM   00098,2,10
        TF   MEMJ1,00097
        MM   MEMJ1,4,9
```

```
        AM   00099,C10-3
        C    MEMSE,00099,611
        BE   EGAL1
        BH   *&48
        TF   MEMK1,MEMJ1
        SM   MEMK1,1,10
        B    BCL2
        TF   MEMI1,MEMJ1
        AM   MEMI1,1,10
        B    BCL2
EGAL1   BB
        DORG*&6
SSP3    TFM  MEMI2,1,10
        TFM  MEMK2,38,10
        AM   MEMSE,4,10
        TFM  CP1,0,10
BCL3    AM   CP1,1,10
        CM   CP1,08,10
        BH   ERSY4
        LD   00099,MEMI2
        A    00099,MEMK2
        DM   00098,2,10
        TF   MEMJ2,00097
        MM   MEMJ2,12,9
        AM   00099,C11-7
        C    MEMSE,00099,611
        BE   EGAL2
        BH   *&48
        TF   MEMK2,MEMJ2
        SM   MEMK2,1,10
        B    BCL3
        TF   MEMI2,MEMJ2
        AM   MEMI2,1,10
        B    BCL3
EGAL2   BD   *&48,SSP3-1
        AM   99,2,10
        A    MEMSE,99,11
        AM   99,2,10
        BB
        DORG*&2
DECAL   AM   MEMSE,1,10
        BNF  *&48,DECAL
        CM   MEMSE,SE&129
        BNH  1DECAL
        B    SSP4
        CM   MEMSE,SE&159
        BL   1DECAL
        SM   MEMSE,1,10
        BTM  IMPER,21,9
1DECALSF MEMSE,,6
        AM   MEMSE,1,10
        BD   *&48,DECAL-1
        CM   MEMSE,00,610
        BNF  *&24
        B    SSP5
        BB
SSP4    TF   *&71,MEMSE
        TF   *&71,MEMSE
        SM   MEMSE,160,7
        TF   *&30,MEMSE
```



```
        TF   *&30,MEMSE
        TR   00000,00000,611
        TF   00000,00000,611
        TFM  *&18,SE-1
        CF   SE-1
        AM   *-6,1,7
        CM   *-18,SE&159
        BNE  *-36
        RACDSE
        BNLC1DECAL
        CF   DECAL
        B    1DECAL
SSP5    TF   *&42,MEMSE
        SM   MEMSE,2,10
        TF   *&23,MEMSE
        TF   00000,00000,611
        AM   MEMSE,2,10
        B    DECAL
        DORG*&6
SSP6    CM   00099,01,610
        BF   SDEB
        CM   99,3,610
        BNE  *&96
        TFM  *&59,C16
        SM   MEMSV,6,10
        TDM  MEMSV,3,6
        SM   MEMSV,1,10
        TFM  MEMSV,,67
        AM   MEMSV,1,10
        B    DEBPAR&24
        CM   99,4,610
        BNE  *&36
        TFM  *-49,C15
        B    *-108
        CM   99,5,610
        BNE  GUO
        AM   CTNIV,1,10
        TFM  MEMSE,41,610
        CM   MEMSO,36,610
        BL   *&48
        CM   MEMSO,38,610
        BH   *&24
        SM   MEMSO,2,10
        SM   MEMSV,6,10
        B    3SDEB
GUO     CM   99,7,610
        BNE  *&84
        SM   MEMSE,10,10
        CM   MEMSE,42,610
        BF   *&24
        BTM  IMPER,98,9
        AM   MEMSE,10,10
        B    GUIMFT
        CM   99,8,610
        BNE  *&24
        BTM  CTAIR,BOUCLE
        BB
SDEB    SM   MEMSV,7,10
        TFM  MEMSV,88888,67
        AM   MEMSV,1,10
```

```
        TDM  MEMSV,2,6
        TFM  MEMSE,31,610
        BTM  DECAL,0,10
        CM   MEMSE,3,610
        BE   *&36
        SM   MEMSE,2,10
        B    4SDEB
        SM   MEMSE,1,10
        CF   MEMSE,,6
        AM   MEMSE,1,10
        BTM  DECAL,0,10
        BTM  SSP3,1,10
2SDEB   AM   99,6,10
        CM   99,52,610
        BNE  *&48
        AM   MEMSE,18,10
        TFM  CTAIR-1,SDEB&48
        B    1CTAIR
        SM   MEMSE,8,10
        CM   99,35,610
        BL   4SDEB
        CM   MEMSE,41,610
        BH   4SDEB
3SDEB   TF   MEMSV,CPTR2,6
        SM   MEMSV,5,10
        TF   MEMSV,MRINT,6
        TF   CTAB1,CPTR2
        TF   LC2,LC1
        AM   LC1,15,10
        TF   LC1,LC1,11
1SDEB   TF   LTAB,LC2
        AM   LTAB,10,10
        CM   LTAB,0,67
        BE   *&84
        AM   CPTR2,1,10
        TF   *&35,LTAB
        AM   *&23,7,10
        TR   CPTR2,0,6
        A    CPTR2,LTAB,11
        SM   CPTR2,1,10
        TF   LC2,LC2,11
        CM   LC2,0,67
        BNE  *&24
        B    4SDEB
        TF   LC2,LC2,11
        B    1SDEB
4SDEB   CM   MEMSO,SO-1
        BNE  PRIOR
        B    VRBCL
CONSULTF   CPTR3,CPTR2
        SM   CPTR3,8,10
        C    ATENT2,CPTR3,11
        BE   *&60
        CM   CPTR3,C13-293
        BNH  EDER7
        SM   CPTR3,22,10
        B    CONSUL&24
        BNE  *&36,TBLEA
        C    CTAB1,CPTR3
        BNH  EDER5
```



```
        AM   CPTR3,6,10
        TD   C7,CPTR3,11
        SM   CPTR3,1,10
        SM   MEMSV,7,10
        CM   C7,8,10
        BE   PNDCL
        CM   C7,4,10
        BE   CSETI
        CM   C7,2,10
        BE   CSAIG
        CM   CPTR3,70000,67
        BL   1CONS
        AM   MEMSO,2,10
        TFM  MEMSO,45,610
        AM   MEMSV,1,10
        AM   CPTR3,1,10
        TF   MEMSV,CPTR3,611
        CM   MEMSE,28,610
        BE   VRBCL
        BTM  IMPER,94,9
1CONS   TF   MEMSV,CPTR3,611
        AM   MEMSV,1,10
        TD   MEMSV,C7,6
        AM   CPTR3,3,10
        B    PRIOR
CSETI   CM   MEMSE,34,610
        BNE  1CSETI
        TF   CALADR,CHOB
        A    CALADR,MPGOB
        SM   CALADR,C3
        BNF  *&36,CPTR3,11
        TF   MEMSV,CPTR3,611
        BT   PFOSI,CALADR
        TF   CPTR3,CALADR,6
        AM   MEMSV,7,10
        B    BOUCLE
1CSETICM   CPTR3,0,67
        BNE  1CONS
        TF   CPTR3,REMAN,6
        SM   REMAN,5,10
        SF   CPTR3,,6
        B    1CONS
CSAIG   BNF  1CSETI&24,CPTR3,11
        B    1CONS
VRBCL   AM   MEMSO,2,10
        TF   MEMSO,MEMSE,611
        B    BOUCLE
PRIOR   TFM  MATR,C9-47
        MM   MEMSO,46,69
        A    99,MEMSE,11
        A    MATR,99
        CM   MATR,C9&41*46
        BNL  EDER2
        BD   *&24,MATR,11
        B    VRBCL
        TD   C7,MATR,11
        SF   CALADR-1
        CM   C7,2,10
        BE   EDER2
        BL   *&24
```

```
        CF   CALADR=1
        BTM  PFO,0,10
        B    GAMMA
PFO     CM   MPGOB,C3&59
        BH   *&24
        BB
        TR   C1,C2=39
        TD   C8,C3&60
        TD   C3&60,C2&41
        TR   C1,C3
        BNC2 *&36
        RCTY
        WNTYC3
        TD   C3&60,C8
        TR   C3,C3&60
        SM   MPGOB,60,10
        TF   C1&78,CHOB
        AM   CHOB,60,10
        TF   C1&73,CHOB
        WNCDC1
        B    PFO
SPG1    BNF  *&24,MEMSV,11
        AM   MRINT,1,10
        SM   MEMSV,1,10
        BNF  *&24,MEMSV,11
        AM   MRINT,5,10
        AM   MEMSV,1,10
        CF   MEMSV,,6
        BB
SPG2    SM   MEMSV,1,10
        BNF  1SPG2&12,MEMSV,11
        TR   C4,ZPREP1
        TF   C4&11,MEMSV,11
        CF   C4&11
        AM   MRINT,5,10
        BD   *&72,SPG2=1
        TFM  C4&6,ADIN2
        TFM  MEMSV,ADIN2,67
3SPG2   TR   MPGOB,C4,6
        AM   MPGOB,12,10
        B    1SPG2
        TFM  C4&6,ADIN1
        TFM  MEMSV,ADIN1,67
        SM   MEMSV,5,10
        BNF  2SPG2,MEMSV,11
        TR   C4&12,ZPREP1
        BD   *&24,MEMSV,11
        TFM  C4&13,6,10
        SM   MEMSV,1,10
        TF   C4&23,MEMSV,11
        AM   MEMSV,5,610
        TF   C4&18,MEMSV,11
        AM   MEMSV,6,10
        TR   MPGOB,C4,6
        AM   MPGOB,24,10
1SPG2   SF   MEMSV,,6
        AM   MEMSV,1,10
        BB
2SPG2   AM   MEMSV,5,10
        B    3SPG2
```



```
GAMMA   CM   MEMSO,7,610
        BH   1GAM
        BTM  SPG1,0,10
        CM   MEMSO,4,610
        BH   GS1
        AM   MEMSV,6,10
        BTM  SPG1,0,10
        SM   MEMSV,6,10
        B    GS1
1GAM    CM   MEMSO,19,610
        BH   GS1
        BTM  SPG2,0,10
        AM   MEMSV,6,10
        BTM  SPG2,1,10
        SM   MEMSV,6,10
GS1     TD   C8,MEMSV,11
        TDM  C8-1,0,11
        MF   FL2RP,C8
        SM   MEMSV,1,10
        CF   FL2T
        BNF  *&24,MEMSV,11
        SF   FL2T
        AM   MEMSV,7,10
        TD   C7,MEMSV,11
        MF   FL1RP,C7
        CF   FLFG
        CF   FLR2
        CF   FLB2
        C    C8,C7
        BNE  *&24
        SF   FLFG
        CM   C8,0,10
        BNF  *&24
        SF   FLR2
        CM   C8,3,10
        BNF  *&24
        SF   FLB2
        A    C8,C7
        SM   MEMSV,6,10
        BNF  GAMSP,CALADR=1
        AM   MEMSV,5,10
        CM   MEMSO,4,610
        BH   *&48
        CM   C8,6,10
        BE   DETAIL
        B    EDER3
        CM   MEMSO,7,610
        BNH  DETAIL
        CM   MEMSO,13,610
        BH   *&48
        CM   C8,2,10
        BNH  DETAIL
        B    EDER4
        CM   MEMSO,19,610
        BH   DETAIL
        B    RELA
DETAILTF CALADR,CHOB
        A    CALADR,MPGOB
        SM   CALADR,C3
        MM   MEMSO,5,6911
```

```
      SM   99,*&23
      B    99,,6
      DSA  ET,OU,IMP,IDEN,NON,SI,TANT,EXPO
      DSA  MUL,DIV,PLUS,MOINS,DENT,PPT,PTEG,EGAL
      DSA  GDEG,GD,DIFF,SOUS,POUR,PAS,JUSQA,1CROCH
      DSA  AFF,ORS,SNEXP,PARENT,ALFA,PTVG,SFIN1,SNINS
      DSA  FAIRE,DEPOIN,AIG,DCL,DCL,DCL,99999,TAB2,PROC
EDER2 BTM  IMPER,30,9
EDER3 BTM  IMPER,31,9
EDER4 BTM  IMPER,32,9
EDER5 BTM  IMPER,5,9
EDER6 BTM  IMPER,33,9
EDER7 BTM  IMPER,35,9
EDER8 BTM  IMPER,37,9
      DORG*&6
CTAIR SM   MEMSE,26,10
      CM   MEMSE,30,610
      BE   *&24
      BTM  IMPER,41,9
      AM   MEMSE,26,10
1CTAIRBTM  DECAL,0,10
      CM   MEMSE,3,610
      BNE  1CTAIR
      BTM  DECAL,0,10
      CM   MEMSE,23,610
      BNE  1CTAIR
      B    CTAIR-1,,6
DEB   TR   MPGOB,ZDEB,6
      TR   C1,C2-39
      TR   C1,FCFL
      TR   C1&69,FCFAD
      BNC2*&36
      RCTY
      WNTYC1
      WNCDC1
      TR   C1,C2-39
      TR   C1,C3
      TF   C1&78,CHOB
      TF   C1&73,MPGOB
      SM   C1&73,C3-7
      A    C1&73,CHOB
      BNC2*&36
      RCTY
      WNTYC3
      WNCDC1
      BNF  *&36,DECAL
      RACOSE
      BNLC*-12
      H
      B    BEGIN
SFIN1 SM   MEMSV,6,10
      CM   MEMSV,88888,67
      BNE  *&36
      AM   MEMSV,7,10
      B    *&72
      AM   MEMSV,1,10
      TF   MRINT,MEMSV,11
      AM   MEMSV,5,10
      TF   CPTR2,MEMSV,11
      AM   MEMSV,6,10
```



```
      CM   MEMSO,SO&1
      BE   DEB
      CM   MEMSO,41,610
      BNF  *&36
      SM   MEMSO,2,10
      B    PRIOR
      SM   MEMSO,2,10
SF1   BTM  DECAL,0,10
      CM   MEMSE,3,610
      BNF  SF1
      BTM  DECAL,0,10
      CM   MEMSE,23,610
      BNE  *&36
      TFM  MEMSE,30,610
      B    PRIOR
      AM   MEMSE,6,10
      C    MEMSE,CFIN,6
      BNE  *&36
      TFM  MEMSE,43,610
      B    PRIOR
      AM   MEMSE,4,10
      C    MEMSE,CSN,6
      BNE  *&36
      TFM  MEMSE,32,610
      B    PRIOR
      SM   MEMSE,12,10
      B    SF1
      B    SF1
SF2   SM   MEMSO,2,10
      B    PRIOR
ORS   CM   MEMSE,32,610
      BE   *&60
      BT   PFOSI,CALADR
      AM   MEMSV,7,10
      SM   MEMSO,2,10
      B    PRIOR
      CM   MEMSV,99999,67
      BNE  S3
      SM   MEMSE,5,610
      SM   MEMSV,5,10
      TD   C7,MEMSV,11
ALSN  CF   C7
      CM   C7,4,10
      BE   DESIG
      TFM  C8,1,10
      SF   FLEG
      CM   C7,3,10
      BE   *&60
      CM   C7,0,10
      BE   *&36
      CF   FLEG
      TDM  MEMSV,0,6
      SM   MEMSV,1,10
      BNF  *&24,MEMSV,11
      AM   MRINT,5,10
      BNF  *&84,FL2RP
      AM   MRINT,12,10
      BNF  *&24,FLR2
      SM   MRINT,11,10
      BNF  *&36,FLEG
```

```
        AM   MEMSV,6,10
        B    S4&60
        AM   MEMSV,6,10
        TF   C1&73,MEMSV,11
        TF   MEMSV,MRINT,6
        TF   RESUL,CALADR
        BTM  AFECT,0,10
S4      AM   MEMSV,6,10
        TF   MEMSV,C1&73,6
        TR   MPGOB,C4,6,
        A    MPGOB,CALADR
        A    CALADR,RESUL
        CM   MEMSO,26,610
        BNE  SNEXP1
        AM   MEMSV,6,10
S3      AM   CALADR,8,10
        BT   PFOSI,CALADR
        TF   MEMSV,REMAN,6
        AM   MEMSV,1,10
        TFM  CALADR,8
        TR   C4,ZORS
        TF   C4&6,REMAN
        SM   REMAN,5,10
        SF   C4&6
        B    COMUN
DESIG   BNF  *&48,MEMSV,11
        AM   MRINT,7,10
        AM   MEMSV,5,10
        B    S4&60
        TR   C4,ZETI1
        TF   C4&6,MRINT
        SM   MEMSV,1,10
        TF   C4&11,MEMSV,11
        TF   C4&18,MRINT
        SM   C4&18,5,10
        TF   C4&23,CPTR3,11
        TR   MPGOB,C4,6,
        AM   MPGOB,24,10
        AM   MEMSV,6,10
        B    S4&60
SNEXP   TF   C4&8,MEMSV,11
        AM   MEMSV,1,10
        TD   C4&9,MEMSV,11
        SM   MEMSV,6,10
        TD   C7,MEMSV,11
        TD   MEMSV,C4&9,6
        SM   MEMSV,1,10
        TF   C4&18,MEMSV,11
        TF   MEMSV,C4&8,6
        AM   MEMSV,6,10
        TF   MEMSV,C4&18,6
        AM   MEMSV,1,10
        TD   MEMSV,C7,6
        B    ALSN
SNEXP1  SM   MEMSV,5,10
        SF   MEMSV,,6
        TD   C7,MEMSV,11
        SM   MEMSV,1,10
        TF   MEMSV,MRINT,6
        CM   C7,3,1011
```



```
        BE   *&48
        BH   *&60
        SM   MRINT,6,10
        SF   MEMSV,,6
        SM   MRINT,1,10
        B    SNINS
        SM   MRINT,12,10
        B    SNINS
SNINS   SM   MEMSV,6,10
        BT   PFOSI,CALADR
        AM   MEMSV,7,10
        SM   MEMSO,2,10
        B    PRIOR
ALEA    TR   C4,ZALEA
        SM   MEMSV,5,10
        TD   C7,MEMSV,11
        SM   MEMSV,1,10
        TF   C4&18,MEMSV,11
        TF   C4&6,CALADR
        AM   C4&6,24,10
        TFM  CALADR,36
        CF   MEMSV,,6
        CM   C7,4,1011
        BE   1ALEA
        CM   C7,4,10
        BNE  EDER8
        TF   C4&23,CTNIV
        S    C4&23,CPTR3,11
        B    2ALEA
1ALEA   TR   C4&37,ZETI1
        TF   C4&43,MRINT
        TF   C4&48,CTNIV
        TFM  C4&50,22,10
        TF   C4&55,MRINT
        TF   C4&60,MEMSV,11
        SM   C4&60,5,10
        TR   MPGOB,C4&36,6
        AM   MPGOB,24,10
        TF   C4&18,MRINT
        AM   MRINT,7,10
2ALEA   AM   MEMSV,7,10
        B    COMUN
        DORG*&6
PFOSI   TR   C1,C2-39
        TF   C1&4,PFOSI-1
        TD   C1&5,C2&41
        CF   MEMSV,,6
        TF   C1&73,MEMSV,11
        TF   C1&78,MEMSV,11
        SM   C1&78,4,10
        AM   C1&73,1,10
        BNC2*&60
        RCTY
        WNTYC1
        SPTY
        WNTYC1&69
        WNCDC1
        BR
EXPO    BNF  *&24,FL1RP
        AM   MRINT,12,10
```

```
        BNF  *&24,FL2RP
        AM   MRINT,12,10
        TR   C4,ZEXPO,,
        TF   C4&23,MEMSV,11
        SM   MEMSV,6,10
        TF   C4&11,MEMSV,11
        TF   C4&35,CALADR
        AM   C4&35,36,10
        TF   C4&42,MRINT
        AM   MEMSV,6,10
        TF   MEMSV,MRINT,6
        AM   MEMSV,1,10
        TDM  MEMSV,0,611
        SM   MRINT,12,10
        TFM  CALADR,48
        BNF  *&60,FLR2
        TFM  C4&13,6,10
        BNF  COMUN,FLEG
        TFM  C4&1,6,10
        B    COMUN
        BNF  *-24,FLEG
        B    COMUN
CAS     BNF  *&84,FL1RP
        AM   MEMSV,6,10
        TF   RESUL,MEMSV,611
        BNF  *&24,FL2RP
        AM   MRINT,12,10
        AM   MEMSV,1,10
        B    COMUN
        BNF  *&84,FL2RP
        TF   RESUL,MEMSV,611
        AM   MEMSV,6,10
        TF   MEMSV,RESUL,611
        AM   MEMSV,1,10
        SF   MEMSV,,6
        B    COMUN
        TF   RESUL,MRINT,6
        SM   MRINT,12,10
        B    *-84
MUL     TFM  RESUL,1,10
        BNF  *&132,FLEG
        BNF  *&36,FLR2
        SM   MEMSV,6,10
        B    CASPM
        TR   C4,ZMULF
        TF   C4&6,MEMSV,11
        SM   MEMSV,6,10
        TF   C4&11,MEMSV,11
        TFM  CALADR,36
        TFM  RESUL,C4&30
        B    CAS
        TR   C4,ZCOVR
        BNF  *&216,FLR2
        TF   C4&11,MEMSV,11
        BNF  *&96,FL1RP
        TF   C4&18,MEMSV,11
        TR   MPGOB,C4,6
        AM   MPGOB,24,10
        AM   MEMSV,1,10
        TDM  MEMSV,0,611
```



```
        SM   MEMSV,7,10
        B    CASPM
        TD   C4&12,C4&24
        TFM  MEMSV,SPCOV,67
        AM   MEMSV,1,10
        TDM  MEMSV,0,6
        SM   MEMSV,7,10
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        B    CASPM
        TD   C4&12,C4&24
        SM   MEMSV,6,10
        TF   C4&11,MEMSV,11
        TFM  MEMSV,SPCOV,67
        BNF  *&36,FL2RP
        AM   MRINT,12,10
        CF   FL2RP
        B    *-120
CASPM   TR   C4,ZMDAS
        A    C4&1,RESUL
        TFM  CALADR,12
        BNF  *&60,FL1RP
        TF   C4&11,MEMSV,11
        AM   MEMSV,6,10
        TF   C4&6,MEMSV,11
        B    CAS&36
        BNF  *&96,FL2RP
        TF   C4&6,MEMSV,11
        AM   MEMSV,6,10
        TF   C4&11,MEMSV,11
        TF   MEMSV,C4&6,6
        AM   MEMSV,1,10
        SF   MEMSV,,6
        B    COMUN
        TF   C4&11,MEMSV,11
        TR   C4&14,ZPREP1
        AM   MEMSV,6,10
        TF   C4&25,MEMSV,11
        TF   C4&20,MRINT
        TF   C4&6,MRINT
        TF   MEMSV,MRINT,6
        SM   MRINT,12,10
        AM   MEMSV,1,10
        BD   *&24,MEMSV,11
        TFM  C4&15,6,10
        TR   MPGOB,C4&14,6
        AM   MPGOB,12,10
        SF   MEMSV,,6
        B    COMUN
PLUS    TFM  RESUL,1,1011
        BNF  MUL&144,FLEG
        BNF  *&24,FLR2
        B    MUL&36
        TFM  RESUL,19,10
        B    MUL&36
DIV     SF   RESUL-2
        TFM  RESUL,7,10
        SM   MEMSV,6,10
        BNF  *&24,FL2RP
        BNF  *&36,FL1RP
```

```
        BNF  MOINS&36,RESUL-2
        B    *&144
        TR   C4,ZPREP1
        TF   C4&11,MEMSV,11
        TF   C4&6,MRINT
        TF   MEMSV,MRINT,6
        CF   FL2RP
        AM   MRINT,12
        BNF  *&24,FLR2
        SM   C4&1,20,10
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        B    *-144
        CM   C7,1,10
        BNE  *&192
        TR   C4,ZCOVR
        AM   MEMSV,6,10
        TF   C4&11,MEMSV,11
        BNF  *&36,FL1RP
        TF   C4&18,MEMSV,11
        B    *&60
        TF   C4&18,MRINT
        TF   MEMSV,MRINT,6
        SM   MRINT,12,10
        SF   FL1RP
        AM   MEMSV,1,10
        TDM  MEMSV,0,611
        SM   MEMSV,7,10
        TR   MPGOB,C4,6
        AM   MPGOB,24,10
        BNF  *&24,FLR2
        B    CASPM
        TR   C4,ZCOVR
        TD   C4&12,C4&24
        TF   C4&11,MEMSV,11
        TFM  MEMSV,SPCOV,67
        BNF  *&36,FL2RP
        AM   MRINT,12,10
        CF   FL2RP
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        B    CASPM
MOINS   CF   RESUL-2
        TFM  RESUL,0,10
        B    DIV&24
        BNF  DIV&216,FLEG
        BNF  *&24,FLR2
        B    CASPM
        TFM  RESUL,20,10
        B    CASPM
DENT    BD   *&24,C7
        BTM  1DENT,0,10
        BNF  *&48,FLR2
        SM   MEMSV,6,10
        BTM  1DENT,1,10
        AM   MEMSV,6,10
        TR   C4,ZDIVE
        TF   C4&23,MEMSV,11
        SM   MEMSV,6,10
        TF   C4&35,MEMSV,11
```



```
        TFM  CALADR,48
        TFM  RESUL,C4&42
        B    CAS
1DENT   TFM  *&47,FL2RP
        BD   *&24,1DENT-1
        TFM  *&23,FL1RP
        BNF  EDER4,0
        TR   C4,ZAFF1
        TF   C4&11,MEMSV,11
        TF   C4&18,MEMSV,11
        TR   MPGOB,C4,6
        AM   MPGOB,24,10
        AM   MEMSV,1,10
        TDM  MEMSV,1,611
        SM   MEMSV,1,10
        BR
ET      TR   C4,ZET
        TFM  CALADR,84
        TFM  MATR,C4&78
        TF   C4&11,MEMSV,11
        SM   MEMSV,6,10
        TF   C4&23,MEMSV,11
        B    BOOL
OU      TR   C4,ZOU
        TFM  CALADR,48
        TFM  MATR,C4&42
        B    ET&36
IMP     TR   C4,ZIMP
        TF   C4&42,CALADR
        AM   C4&42,72,10
        TF   C4&66,CALADR
        AM   C4&66,80,10
        TFM  CALADR,80
        TFM  MATR,C4&54
        SF   MATR-1
        B    ET&36
IDEN    TR   C4,ZIMP
        SM   C4&45,1,10
        B    IMP&12
BOOL    TF   MATR,MRINT,6
        BNF  *&24,MATR-1
        TF   C4&74,MRINT
        AM   MEMSV,6,10
        TF   MEMSV,MRINT,6
        AM   MEMSV,1,10
        SF   MEMSV,,6
        SM   MRINT,1,10
        B    COMUN
SOUS    BNF  *&24,FL2T
        AM   MRINT,5,10
        BNF  *&24,FLR2
        B    EDER4
        SM   MEMSV,5,10
        SF   MEMSV,,6
        SM   MEMSV,1,10
        BNF  S1,FL2RP
        TR   C4,ZSSE
        TF   C4&11,MEMSV,11
        TF   C4&18,MEMSV,11
        TF   C4&38,MEMSV,11
```

```
         TF   C4&6,CALADR
         AM   C4&6,32,10
         TF   C4&30,CALADR
         AM   C4&30,44,10
         AM   MEMSV,1,10
         TFM  CALADR,44
         BNF  COMUN,FLR2
         SM   C4&11,2,10
         SM   C4&18,2,10
         SM   C4&38,2,10
         B    COMUN
S1       TR   C4,ZPREP1
         TF   C4&6,MRINT
         TF   C4&11,MEMSV,11
         TF   MEMSV,MRINT,6
         SM   MRINT,12,10
         BNF  *&24,FLR2
         TFM  C4&1,6,10
         TR   MPGOB,C4,6
         AM   MPGOB,12,10
         AM   CALADR,12,10
         B    SOUS&96
NON      BNF  EDER3,FLR2
         TR   C4,ZNON
         SM   MEMSV,6,10
         TF   C4&23,MEMSV,11
         TF   MEMSV,MRINT,6
         AM   MEMSV,1,10
         TF   C4&42,MRINT
         SM   MRINT,1,10
         SF   MEMSV,,6
         TFM  CALADR,48
         B    COMUN
AFF      CM   MEMSF,33,610
         BNF  *&72
         BNF  *&36,S1S2
         CF   S1S2
         B    *&36
         SM   MEMSV,5,10
         B    VIRG1
         BNF  *&48,FL2RP
         AM   MRINT,12,10
         BNF  *&24,FLR2
         SM   MRINT,11,10
         BNF  *&24,FL1RP
         BTM  IMPER,34,9
         BNF  *&24,FL2T
         AM   MRINT,5,10
         BNF  *&24,MEMSV,11
         AM   MRINT,5,10
         BTM  AFECT,0,10
         AM   MEMSV,7,10
         SM   MEMSO,2,10
         CM   MEMSO,25,610
         BE   COMUN&12
         AM   MEMSV,6,10
         B    COMUN  &12
AFECT    BNF  S2,FLEG
         TR   C4,ZPREP1
         TF   C4&6,MEMSV,11
```



```
         SM   MEMSV,6,10
         TF   C4&11,MEMSV,11
         TFM  CALADR,12
         BNF  *&24,FLR2
         TFM  C4&1,6,10
         BNF  *&24,FLR2
         TFM  C4&1,25,10
         BB
S2       CM   C8,2,10
         BH   EDER6
         BNF  *&36,FLR2
         TR   C4,Z4FF1
         B    *&24
         TR   C4,ZCOVR
         TF   C4&18,MEMSV,11
         SM   MEMSV,6,10
         TF   C4&11,MEMSV,11
         TFM  CALADR,24
         BB
COMUN    SM   MEMSO,2,10
         TR   MPGOB,C4,6
         A    MPGOB,CALADR
         B    PRIOR
PTVG     SM   MEMSO,2,10
         SM   MEMSV,5,10
         B    PRIOR
PARENTCM MEMSV,69000,67
         BNH  *&36
         CM   MEMSV,70000,67
         BL   FCTSTD
         SM   MEMSO,2,10
         CM   MEMSO,45,610
         BE   PROCES
         CM   MEMSO,0,610
         BE
         AM   MEMSO,2,10
DERPARSM  MEMSV,5,10
         SM   MEMSO,2,10
         CF   SYMBOL
         B    SYMBOL
SI       TR   C4,ZSI
         SM   MEMSV,5,10
         BNF  EDER3,FLR2
         SM   MEMSV,1,10
         TF   C4&11,MEMSV,11
         TF   C4&6,REMAN
         SF   C4&6
         TF   MEMSV,REMAN,6
         SM   REMAN,5,10
         SM   MEMSV,6,10
         TFM  MEMSV,99999,67
         AM   MEMSV,1,10
         SM   MEMSO,2,10
         TR   MPGOB,C4,6
         AM   MPGOB,12,10
         B    VRBCL
RELA     CM   C8,2,10
         BH   EDER4
         BNF  *&24,FL1RP
         AM   MRINT,12,10
```

```
        BNF  *&24,FL2RP
        AM   MRINT,12,10
        BNF  R1,FLEG
        BNF  CONST,FLR2
R2      BNF  *&24,FL1RP
        B    R1-36
        TR   C4,ZPREP1
        TF   C4&11,MEMSV,11
        TF   C4&6,MRINT
        TFM  C4&1,6,10
        BNF  *&24,FL2PP
        SM   C4&6,12,10
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        TF   MEMSV,C4&6,6
        TR   C4,ZRELA
        TFM  C4&1,2,10
        B    CONST&12
R1      TR   C4,ZCOVR
        TD   C4&12,C4&24
        BNF  *&72,FLR2
        TF   C4&11,MEMSV,11
        TFM  MEMSV,SPCOV,67
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        B    R1-36
        SM   MEMSV,6,10
        TF   C4&11,MEMSV,11
        TFM  MEMSV,SPCOV,67
        CF   FL2RP
        AM   MEMSV,6,10
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        B    R2
CONST   TR   C4,ZRELA
        TF   C4&6,MEMSV,11
        TF   MEMSV,MRINT,6
        SM   MEMSV,6,10
        TF   C4&11,MEMSV,11
        AM   MEMSV,7,10
        TDM  MEMSV,3,611
        TF   C4&30,MRINT
        TF   C4&50,MRINT
        SM   MRINT,1,10
        B    DETAIL
PPT     TFM  C4&13,47,10
        TFM  C4&21,13,10
        B    RCOM
PTEG    TFM  C4&13,47,10
        TFM  C4&21,11,10
        B    RCOM
EGAL    TFM  C4&13,46,10
        TFM  C4&21,12,10
        B    RCOM
GDEG    TFM  C4&13,47,10
        TFM  C4&21,13,10
        B    RCOM
GD      TFM  C4&13,46,10
        TFM  C4&21,11,10
        B    RCOM
```



```
DIFF    TFM  C4&13,47,10
        TFM  C4&21,12,10
        B    RCOM
RCOM    TF   C4&18,CALADR
        AM   C4&18,44,10
        TF   C4&42,CALADR
        AM   C4&42,56,10
        TFM  CALADR,56
        B    COMUN
DCL     SM   MEMSV,5,10
        SM   MEMSO,2,10
        TFM  GREEL-1,MRINT,7
        CM   MEMSO,39,610
        BNE  *&24
        TFM  GREEL-1,REMAN,7
        AM   MEMSO,2,10
        B    1GAMSP
GAMSP   CM   MEMSO,40,610
        BE   TRLEA
        CM   MEMSE,24,610
        BE   PCROCH
        CM   MEMSO,35,610
        BNE  *&48
        CM   MEMSE,25,610
        BE   AFAIG
        B    VGAIG
        CM   MEMSE,46,610
        BE   BOUCLE
        CM   MEMSO,25,610
        BE   VIRG1
        BL   PVIRGU
        CM   MEMSO,28,610
        BE   PARVIR
        B    DCL&12
1GAMSPSF   MEMSV
        CM   MEMSO,37,610
        BE   GENT
        BL   GREEL
        TDM  MEMSV,3,6
        B    *&48
GENT    TDM  MEMSV,1,6
        B    *&24
GREEL   TDM  MEMSV,0,6
        CF   MEMSV
        SM   MEMSV,1,610
        SF   MEMSV
        TF   MEMSV,GREEL-1,611
        CF   MEMSV
        CM   MEMSO,38,610
        BE   *&24
        SM   GREEL-1,11,610
        SM   GREEL-1,1,610
        AM   MEMSV,6,10
        CM   MEMSE,42,610
        BE   BOUCLE
        SM   MEMSO,2,10
        B    PRIOR
PARVIRSM   MEMSO,2,10
        CM   MEMSO,45,610
        BE   *&24
```

```
        BTM  IMPER,36,9
        AM   CTSOR,1,10
        AM   MEMSO,2,10
        B    BOUCLE
PCROCHTD MI,MEMSV,11
        SM   MEMSV,1,10
        TF   T5,MEMSV,11
        AM   MEMSV,1,10
        TF   MEMSV,CPTR3,6
        SM   MEMSV,7,10
        TF   MEMSV,T5,6
        AM   MEMSV,1,10
        TD   MEMSV,MI,6
        B    VRBCL
POUR    SM   MEMSV,5,10
        TD   PX,MEMSV,11
        SM   MEMSV,1,10
        TF   PX=1,MEMSV,11
        TF   MEMSV,REMAN,6
        SM   REMAN,10,10
        SF   MEMSV,,6
        SM   MEMSV,5,10
        TDM  MEMSV,3,6
        SM   MEMSV,6,10
        TD   MEMSV,PX,6
        TR   C4,ZADR
        TF   C4&11,CALADR
        AM   C4&11,12,10
        TF   C4&6,MRINT
        SM   MRINT,10,10
        SM   MEMSV,1,10
        TF   MEMSV,MRINT,6
        AM   MEMSV,1,10
        TFM  CALADR,12
        B    COMUN
VIRG1   BNF  *&24,FL2T
        AM   MRINT,5,10
        CM   MEMSE,22,610
        BE   PAS1
        BNF  *&48,FL2RP
        AM   MRINT,12,10
        BNF  *&24,FLFG
        B    *&72
        AM   MEMSV,5,10
        BTM  AFECT,0,10
        AM   MEMSV,1,10
        TR   MPGOB,C4,6
        A    MPGOB,CALADR
        CM   MEMSO,23,610
        BE   J2
        CM   MEMSE,33,610
        BE   1VG
        BL   TANT1
        TR   C4,ZP1
        TF   C4&11,CHOB
        A    C4&11,MPGOB
        SM   C4&11,C3=20
        TF   C4&6,MRINT
        AM   C4&6,10,10
        AM   MEMSV,17,10
```



```
        TF   C4&18,MEMSV,11
        SM   MEMSV,11,10
        TR   MPGOB,C4,6
        AM   MPGOB,20,10
        B    BOUCLE
1VG     TR   C4,ZADR
        TF   C4&6,MRINT
        AM   C4&6,10,10
        TF   C4&11,REMAN
        SF   C4&11
        TFM  CALADR,12
        AM   MEMSV,12,10
        AM   MEMSO,2,10
S1S2    TD   MEMSV,PX,6
        SM   MEMSV,1,10
        TF   MEMSV,PX=1,6
        AM   MEMSV,6,10
        TF   RESUL,CHOB
        A    RESUL,MPGOB
        SM   RESUL,C3
        A    RESUL,CALADR
        BT   PFOSI,RESUL
        SM   MEMSV,5,610
        AM   RESUL,12,10
        BT   PFOSI,RESUL
        TF   MEMSV,REMAN,6
        SM   REMAN,5,10
        BNF  *&24,PX=1
        SF   MEMSV,,6
        SM   MEMSV,11,10
        SF   S1S2
        B    COMUN
FAIRE   AM   MRINT,10,10
        SM   MEMSV,6,10
        TR   C4,ZORS
        TF   C4&6,MRINT
        SF   C4&6
        TR   MPGOB,C4,6
        AM   MPGOB,8,10
        AM   CALADR,8,10
        BNF  *&24,MEMSV,11
        AM   MRINT,5,10
        BT   PFOSI,CALADR
        AM   MEMSV,7,10
        SM   MEMSO,2,10
        B    PRIOR
TANT1   AM   MEMSV,6,10
        SM   MRINT,12,10
        CM   MEMSE,7,610
        BE   VRBCL
        SM   MEMSO,2,10
        B    PRIOR
TANT    BNF  EDER3,FLR2
        SM   MEMSV,6,10
        AM   MRINT,12,10
        TR   C4,ZSI
        TF   C4&11,MEMSV,11
        CM   MEMSE,42,610
        BE   *&72
        TF   C4&6,REMAN
```

```
        SF   C4&6
        TFM  CALADR,12
        AM   MEMSV,13,10
        B    S1S2
        TR   C4&12,ZORS
        TR   C4&20,ZADR
        TF   C4&6,CALADR
        TF   C4&31,CALADR
        AM   C4&6,20,10
        AM   C4&31,32,10
        TF   C4&26,MRINT
        AM   C4&26,10,10
        AM   MEMSV,18,10
        TF   C4&18,MEMSV,11
        TR   MPGOB,C4,6
        AM   MPGOB,32,10
        SM   MEMSO,2,10
        SM   MEMSV,11,10
        B    BOUCLE
PAS1    BNF  *&24,FL2RP
        AM   MRINT,12,10
        AM   MEMSV,5,10
        TF   MEMSV,PX=1,6
        BTM  AFECT,0,10
        TR   MPGOB,C4,6
        A    MPGOB,CALADR
        AM   MEMSV,6,10
        TF   MEMSV,MRINT,6
        AM   MEMSV,1,10
        TR   C4,ZADR
        TF   C4&6,MRINT
        AM   C4&6,10,10
        TF   C4&11,CHOB
        A    C4&11,MPGOB
        SM   C4&11,C3=12
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        B    VRBCL
PAS     SM   MEMSV,5,10
        B    VIRG1
JUSQA   BNF  *&24,FL2T
        AM   MRINT,5,10
        TF   RESUL,CALADR
        SM   MEMSV,12,610
        SM   MEMSV,5,10
        TFM  CALADR,0
        B    VIRG1&48
J2      AM   MEMSV,5,10
        AM   MEMSV,12,610
        AM   MRINT,12,10
        A    RESUL,CALADR
        TR   C4,ZADR
        TF   C4&6,MRINT
        AM   C4&6,5,10
        AM   MEMSV,7,10
        TD   C7,MEMSV,11
        CM   C7,4,10
        BE   J1
        TR   C4&12,ZF2
        TDM  MEMSV,4,6
```



```
        SM   MEMSV,1,10
        TF   MEMSV,RESUL,6
        AM   MEMSV,12,610
        TF   C4&18,RESUL
        AM   C4&18,44,10
        TF   C4&30,PX=1
        TF   C4&35,MRINT
        TF   C4&42,RESUL
        AM   C4&42,56,10
        TF   C4&50,RESUL
        AM   C4&50,13,10
        TF   C4&62,MRINT
        SM   C4&62,12,10
        TF   C4&67,PX=1
        TF   C4&74,MRINT
        SM   C4&74,12,10
        TF   C4&79,MRINT
        TF   C4&98,RESUL
        AM   C4&98,13,10
        TF   C4&110,MRINT
        AM   C4&110,10,10
        TF   C4&115,MRINT
        AM   C4&115,5,10
        TF   C4&122,C4&115
        SF   C4&122
        AM   MEMSV,6,10
        TF   C4&86,MEMSV,11
        AM   C4&86,5,10
        BD   *&48,PX
        TFM  C4&25,1,10
        TFM  C4&57,2,10
        TFM  C4&69,3,10
        TFM  CALADR,124
        SM   MEMSV,6,10
        B    J1&48
J1      SM   MEMSV,1,10
        TR   C4&12,ZORS
        TF   C4&18,MEMSV,11
        TFM  CALADR,20
        CM   MEMSE,33,610
        BNE  *&60
        TF   C4&11,REMAN
        SF   C4&11
        AM   MEMSV,1,10
        B    S1S2
        TF   C4&11,RESUL
        A    C4&11,CALADR
        SM   MEMSV,5,10
        TR   MPGOB,C4,6
        A    MPGOB,CALADR
        SM   MEMSO,2,10
        B    BOUCLE
AFAIG   SM   MEMSV,1,10
        TR   C4,ZORS
        TF   C4&6,MEMSV,11
        TR   MPGOB,C4,6
        AM   MPGOB,8,10
        TF   MRINT1,MRINT
        TFM  MRINT,CAIG
        AM   CTNIV,1,10
```

```
1AFAIGSM  MEMSV,6,10
      TF  MEMSV,CHOR,6
      A   MEMSV,MPGOB,6
      SM  MEMSV,C3,67
      AM  MEMSV,1,10
      TDM MEMSV,4,6
      B   BOUCLE
VGAIG CF  VGAIG
      BNF 2VGAIG,FLEG
      BNF *&36,FL2RP
      AM  MRINT,7,10
      B   3VGAIG
      TR  C4,ZFTI1
      TF  C4&6,MRINT
      TF  C4&18,MRINT
      SM  MEMSV,1,10
      TF  C4&11,MEMSV,11
      TF  C4&23,CPTR3,11
      SM  C4&18,5,10
      TR  MPGOB,C4,6
      AM  MPGOB,24,10
      AM  MEMSV,1,10
3VGAIGTR  MPGOB,CAIG1,6
      AM  MPGOB,8,10
      AM  MEMSV,5,10
      BNF 1AFAIG,VGAIG
      B   2AIG
2VGAIGBTM IMPER,42,9
AIG   SF  VGAIG
      SM  MEMSV,5,10
      S   CALADR,MPGOB
      B   VGAIG&12
2AIG  SM  MEMSV,6,10
      TFM C8,0,10
      SM  MPGOB,1,10
1AIG  AM  MEMSV,6,10
      BNF *&24,MEMSV,11
      B   *&60
      AM  C8,1,10
      AM  MPGOB,5,10
      TF  MPGOB,MEMSV,611
      B   1AIG
      AM  MPGOB,2,10
      TF  MPGOB,C8,6
      TF  MRINT,MRINT1
      SM  CTNIV,1,10
      AM  MPGOB,1,10
      A   CALADR,MPGOB
      MM  CALADR,5,10
      BD  4AIG,99
3AIG  TD  MPGOB,C2&41,6
      BT  PFOSI,CALADR
      SM  MEMSO,2,10
      AM  MEMSV,7,10
      B   PRIOR
4AIG  TDM MPGOB,0,611
      AM  MPGOB,1,10
      AM  CALADR,1,10
      B   3AIG
1CROCHCM  C7,2,10
```



```
BNE CROCH
TR  C4,ZAIG2
TF  C4&23,MEMSV,11
CF  C4&23
SM  MEMSV,6,10
TF  C4&11,MEMSV,11
BNF *&24,FLR2
TFM C4&1,6,10
TF  C4&35,CALADR
AM  C4&35,36,10
TR  MPGOB,C4,6
AM  MPGOB,36,10
SM  MEMSO,2,10
AM  MEMSV,12,10
B   APAIG
```

MESSAGES D'ERREURS

Dans l'état actuel du compilateur, seules certaines erreurs syntaxiques sont décélées à la compilation, et signalées sur machine à écrire par la frappe du libellé ER suivi d'un numéro d'ordre :

ER 14 : Ecriture incorrecte d'un symbole de base.

ER 21 : Le nombre de DEBUT n'est pas égal au nombre de FIN.

ER 30 : S (E) et S (0) ne sont pas comparables (matrice de priorité).

ER 31 : L'opérande (ou les opérandes) d'un opérateur logique, de SI, ou de TANTQUE n'est pas du type BOOLEEN.

ER 32 : L'opérande (ou les opérandes) d'un opérateur arithmétique, d'un opérateur de relation, de PAS, de JUSQUA n'est pas du type ENTIER ou REEL.

ER 33 : Dans une instruction d'affectation, les types des opérandes ne sont pas REEL ou ENTIER, ou tous les deux BOOLEEN.

ER 34 : La partie gauche d'une instruction d'affectation est un résultat intermédiaire.

ER 35 : Un identificateur n'a pas été déclaré.

ER 37 : Le résultat d'une expression de désignation n'est pas un étiquette.

ER 41 : COMMENTAIRE ne suit pas DEBUT ou "point virgule".

Remarque : D'autres erreurs sont détectées par [5] , [8] , et [9] .

Le libellé est édité par un sous-programme (IMPER) de [5] , qui imprime ensuite la partie non encore analysée de la carte où a été détectée l'erreur, puis arrête la compilation.

Dans la mise au point définitive du compilateur, il faudra :

- détecter le maximum d'erreurs syntaxiques,
- inclure un sous-programme d'auto-correction de certaines erreurs
- offrir la possibilité de corriger les autres erreurs sans interrompre la compilation.

REFERENCES

[1] N. CHOMSKY : On certain formal properties of grammars . Information and Control . 2 (1959) p. 137-167

[2] R. W. FLOYD : Syntactic analysis and operator precedence. Journal of the A. C. M. 10 (1963) p. 316-333

[3] C. PAIR : Arbres, piles et compilation. Revue française de traitement de l'information. 7 (1964) p. 199-216

[4] C. PAIR : Sur la sémantique des langages de programmation . Centre de Calcul Automatique - NANCY .

[5] J. ANDRE : Participation à la construction d'un compilateur ALGOL pour 1620 I. B. M. - Thèse de 3ème cycle de mathématiques (1964) Centre de Calcul Automatique - NANCY

[6] P. NAUR (EDITOR) : Revised report on the algorithmic language ALGOL 60. Communication of the A. C. M. 6 (1963) p. 1 - 17

[7] L. BOLLIET - N. GASTINEL - P. J. LAURENT : Un nouveau langage scientifique : ALGOL - Hermann, PARIS, (1964).

[8] Traitement des procédures : Communication ultérieure du Centre de Calcul de NANCY.

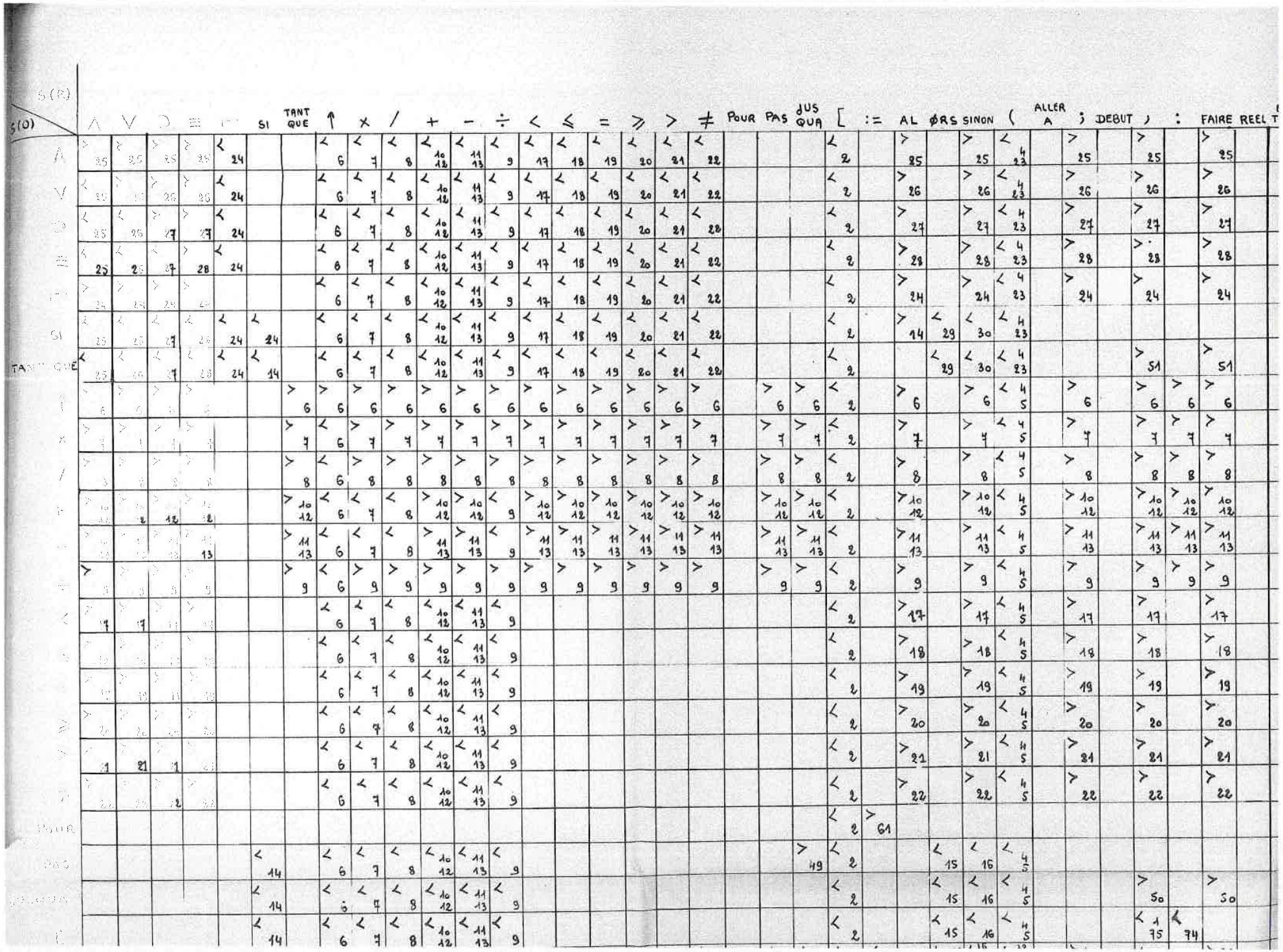
[9] J. M. LAPORTE : Traitement des tableaux ALGOL ; Extension à ALGOL matriciel - Thèse de 3ème cycle de Mathématiques (1964 Centre de Calcul Automatique - NANCY -

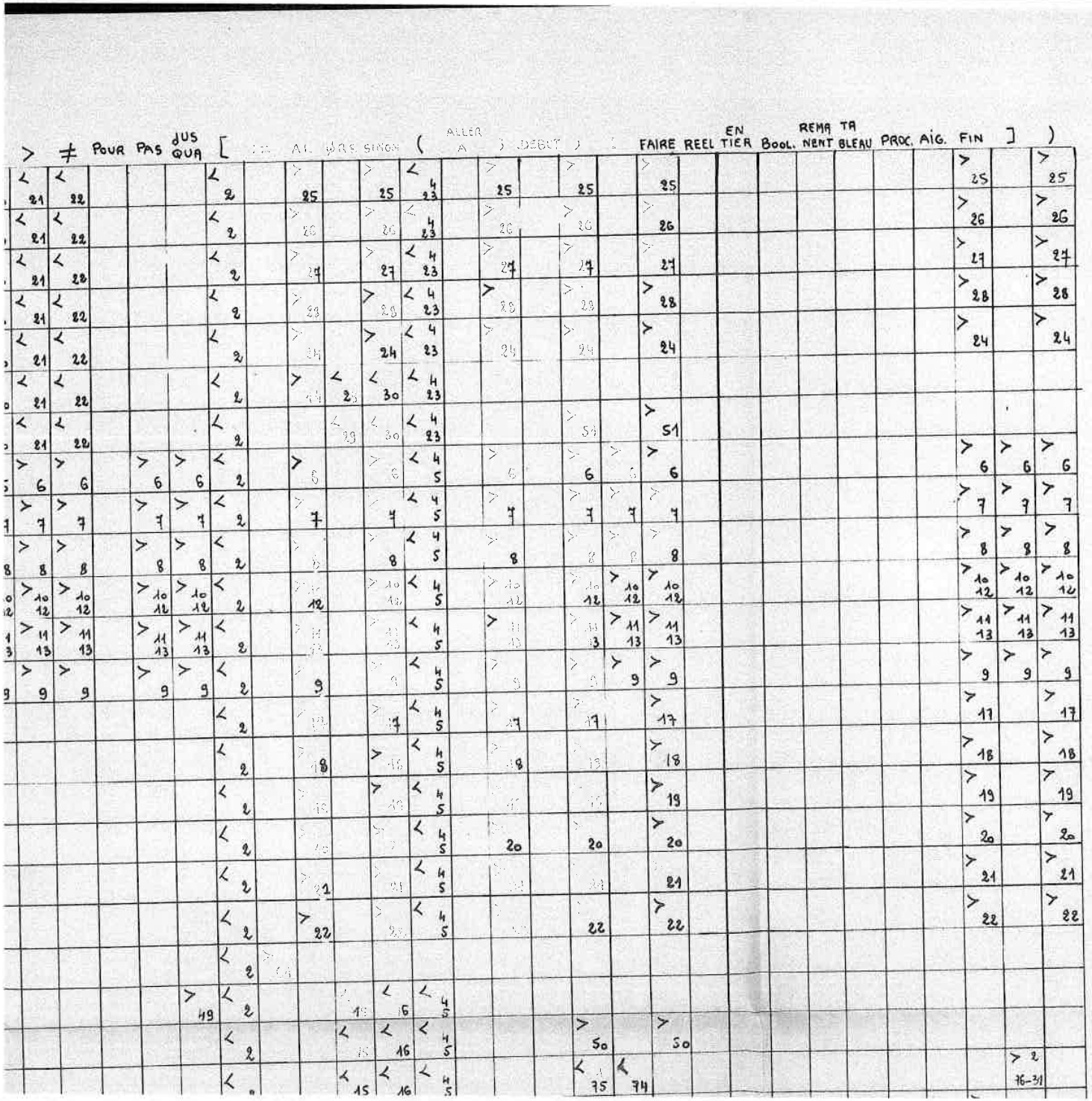
[10] Unités composantes du 1620 I. B. M. - 10. 1339, Mai 1962 .

Vu et Approuvé
Nancy, le
Le Doyen de la Faculté
des Sciences de NANCY

M. AUBRY

Vu et Permis d'imprimer
Nancy, le
le Recteur :
Président du Conseil de
l'Université
P. IMBS


S(F)
S(D)
SI
TANT QUE
POUR
PAS
DUS QUA
ALLER A
DEBUT
FAIRE
REEL



N 1 : 1 - 3 - 67 - 69 - 75 - 77 - 78 - 84

N 2 : 4 - 5 - 23 - 32 - 68

N 3 : 14 - 29 - 33 - 44 - 45 - 48

N 4 : 15 - 29 - 33 - 44 - 45

N 5 : 16 - 30 - 34 - 46 - 47

N 6 : 16 - 30 - 34 - 46 - 47

N 7 : 16 - 30 - 34 - 46 - 47

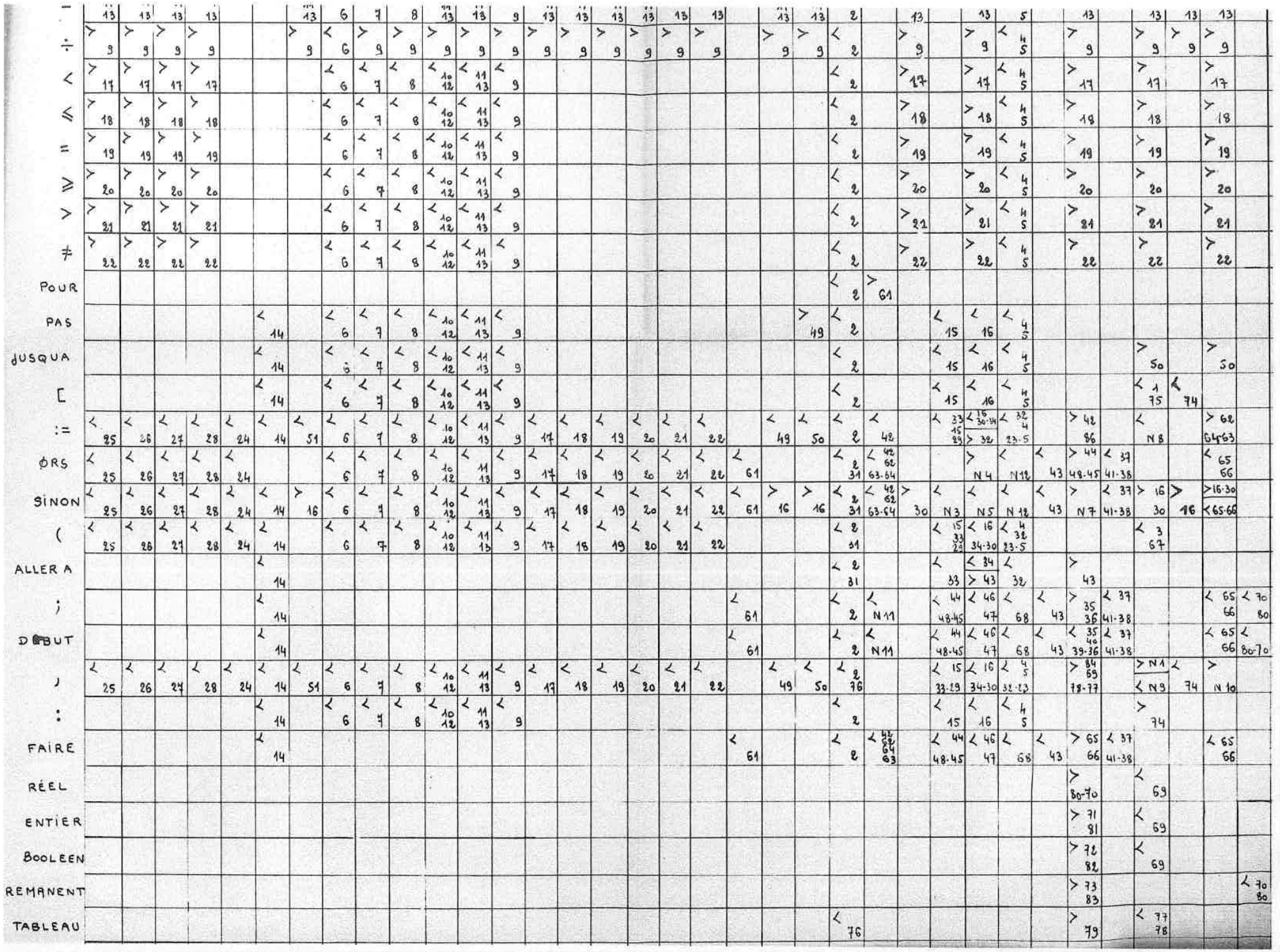
N 8 : 52 - 53 - 54 - 55 - 56 - 57 - 58 - 59 - 60 -

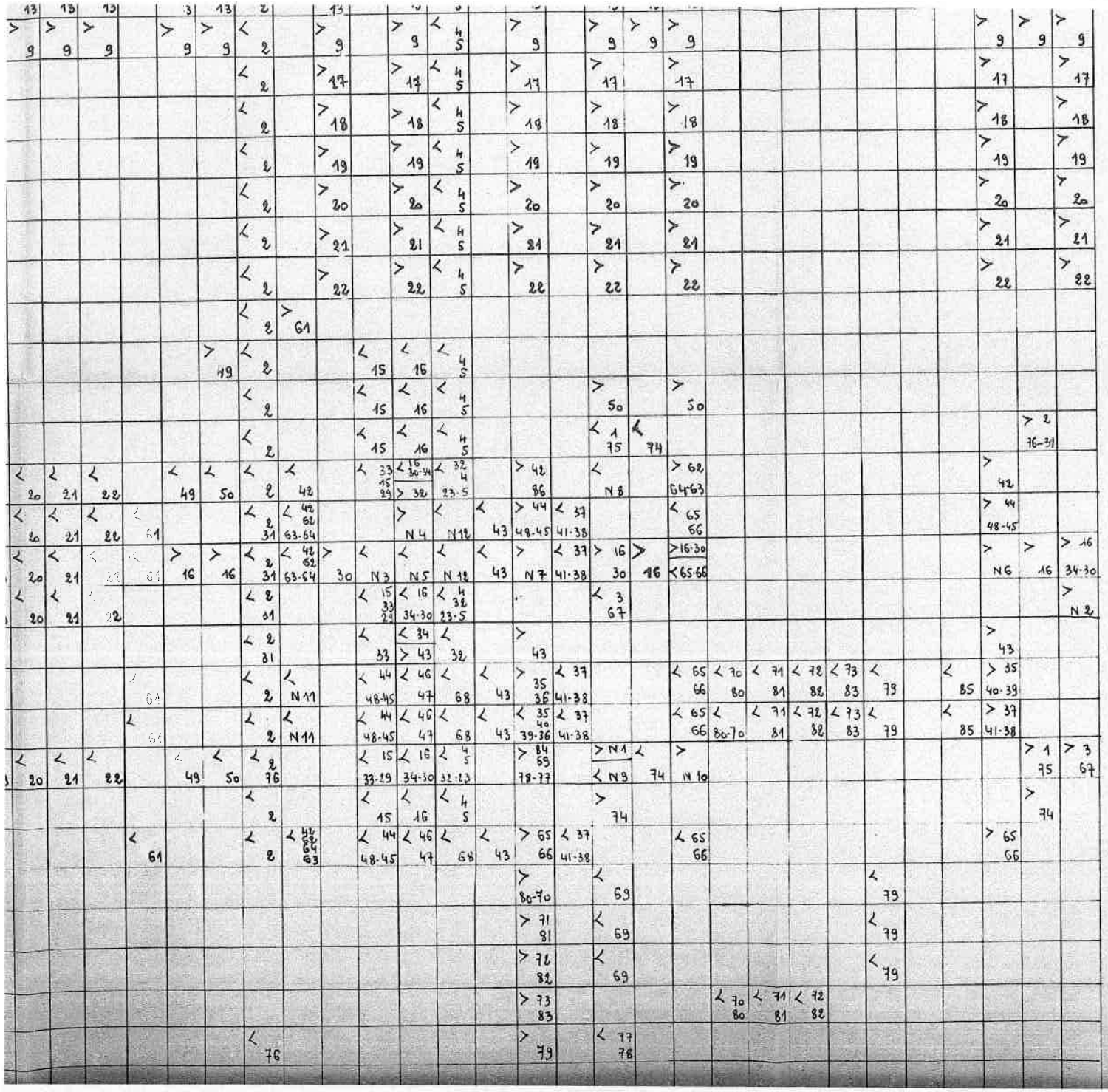
N 9 : 52 - 53 - 54 - 55 - 56 - 57 - 58 - 59 - 60

N 10 : 52 - 53 - 54 - 55 - 56 - 57 - 58 - 59 - 60

N 11 : 42 - 62 - 63 - 64 - 86

N 12 : 4 - 5 - 23 - 32 - 68


=
:
<
≤
=
≥
>
≠
POUR
PAS
dUSQUA
[
:=
ORS
SINON
(
ALLER A
;
DEBUT
)
:
FAIRE
RÉEL
ENTIER
BOOLEEN
REMANENT
TABLEAU

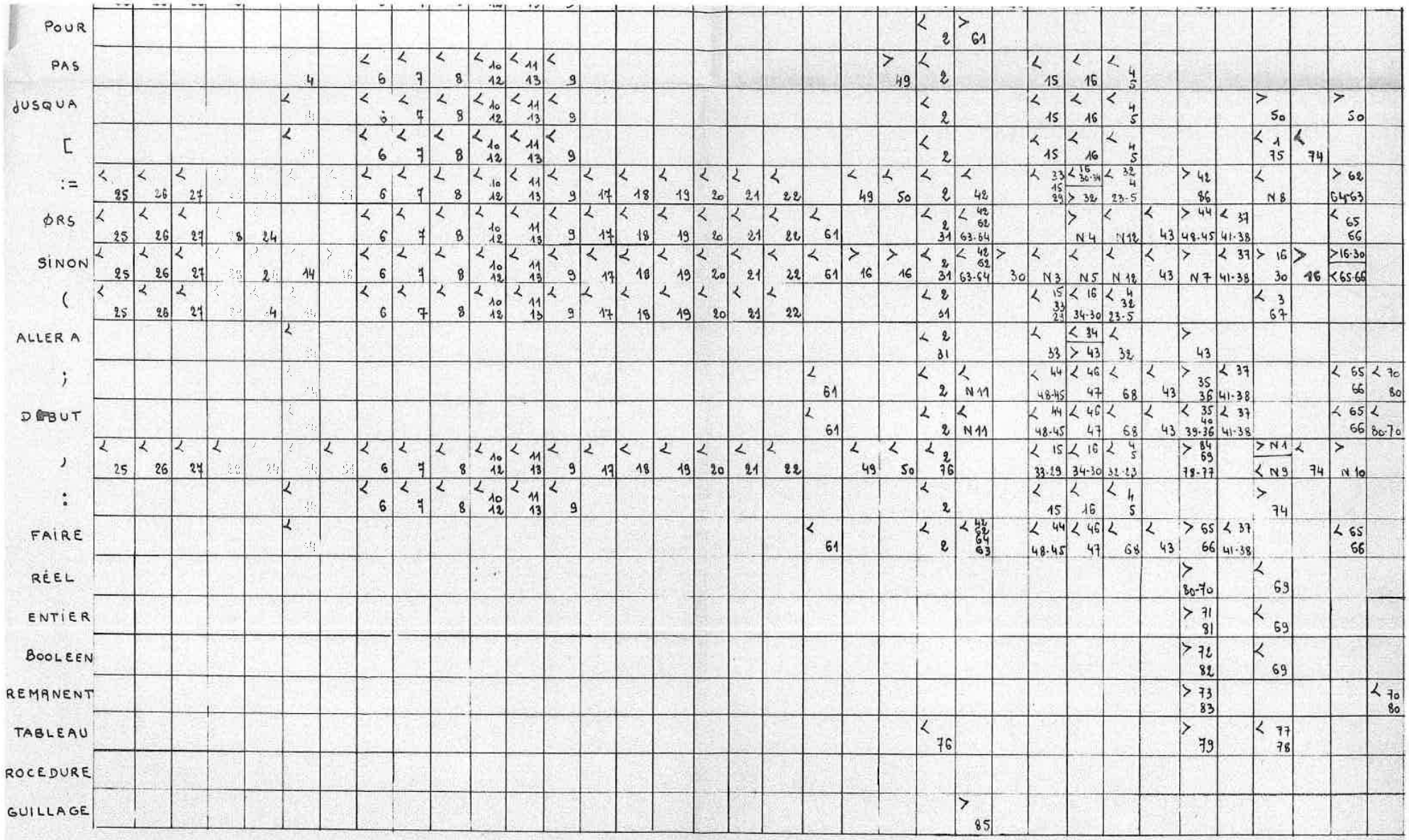


N 9 : 52 - 53 - 54 - 55 - 56 - 57 - 58 - 59 -

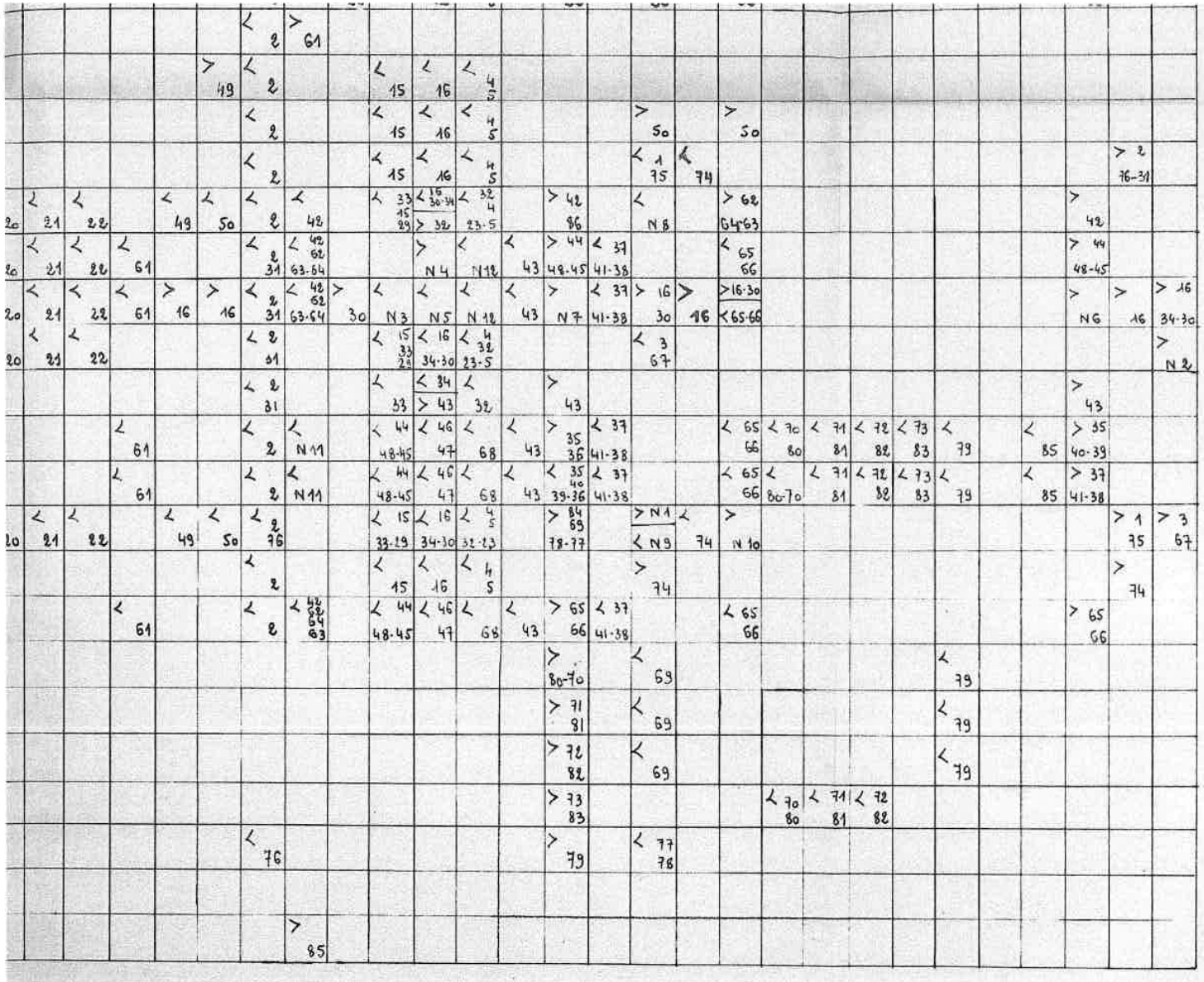
N 10 : 52 - 53 - 54 - 55 - 56 - 57 - 58 - 59 -

N 11 : 42 - 62 - 63 - 64 - 66

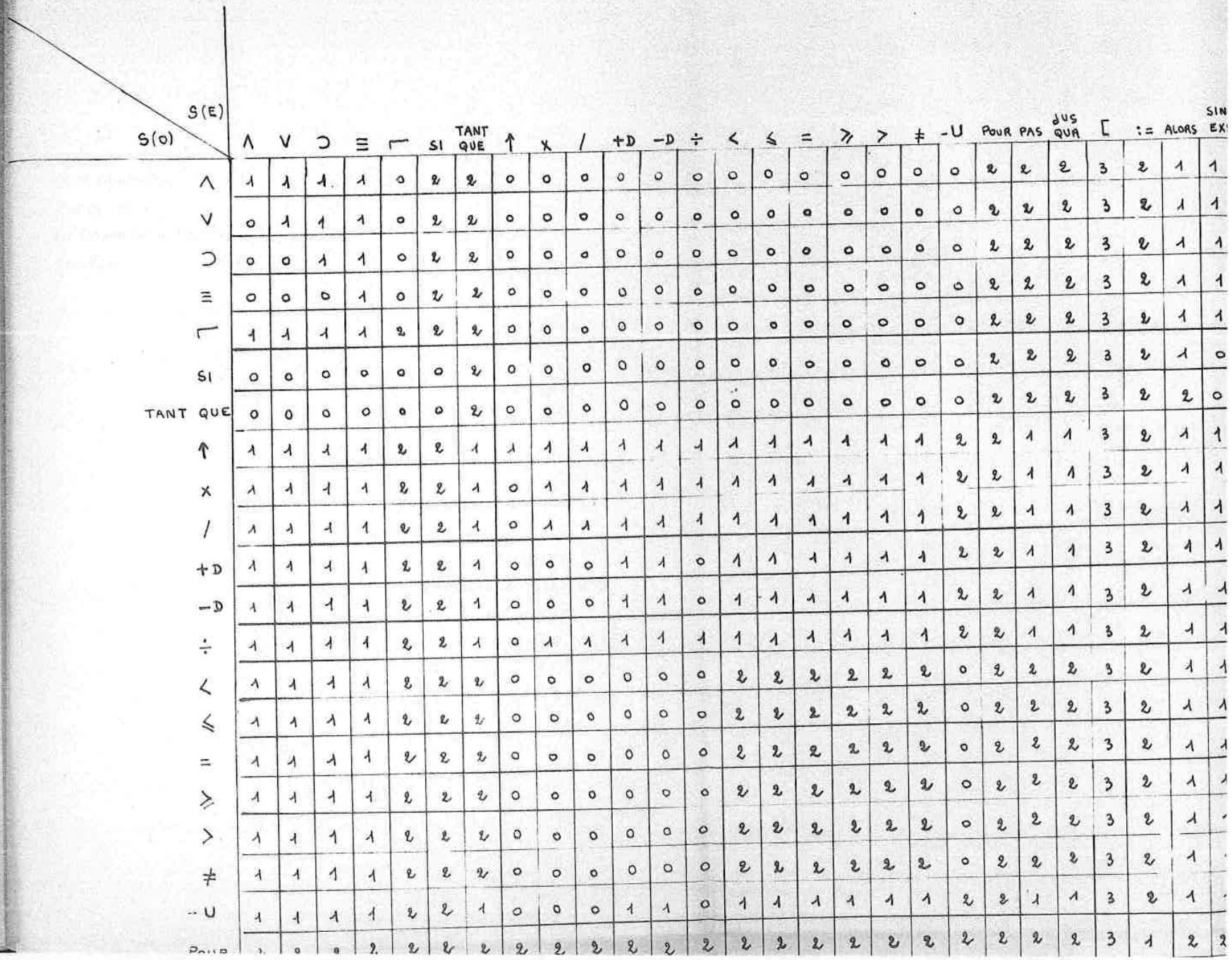
N 12 : 4 - 5 - 23 - 32 - 68



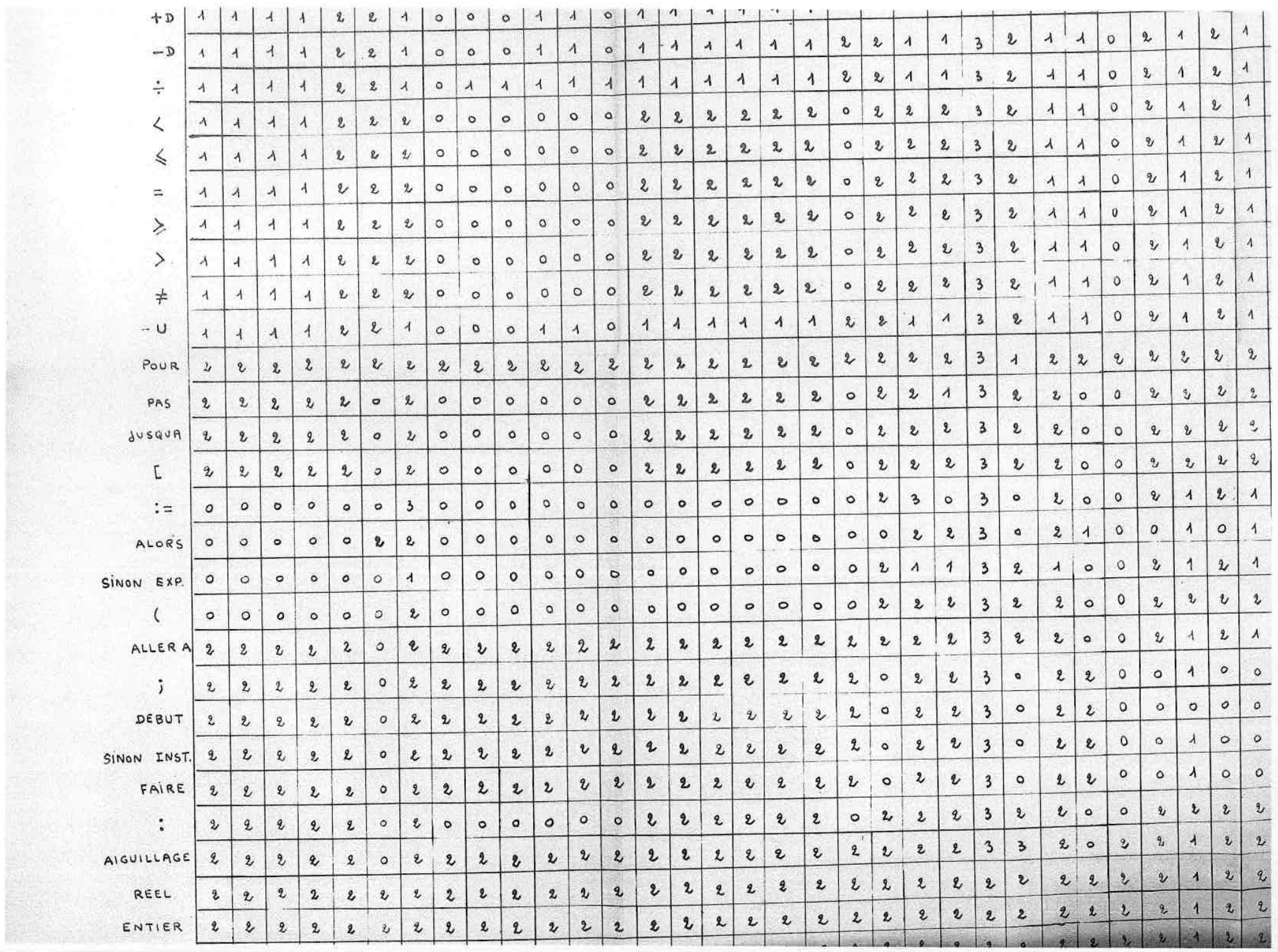
MATRICE de PRIORITE

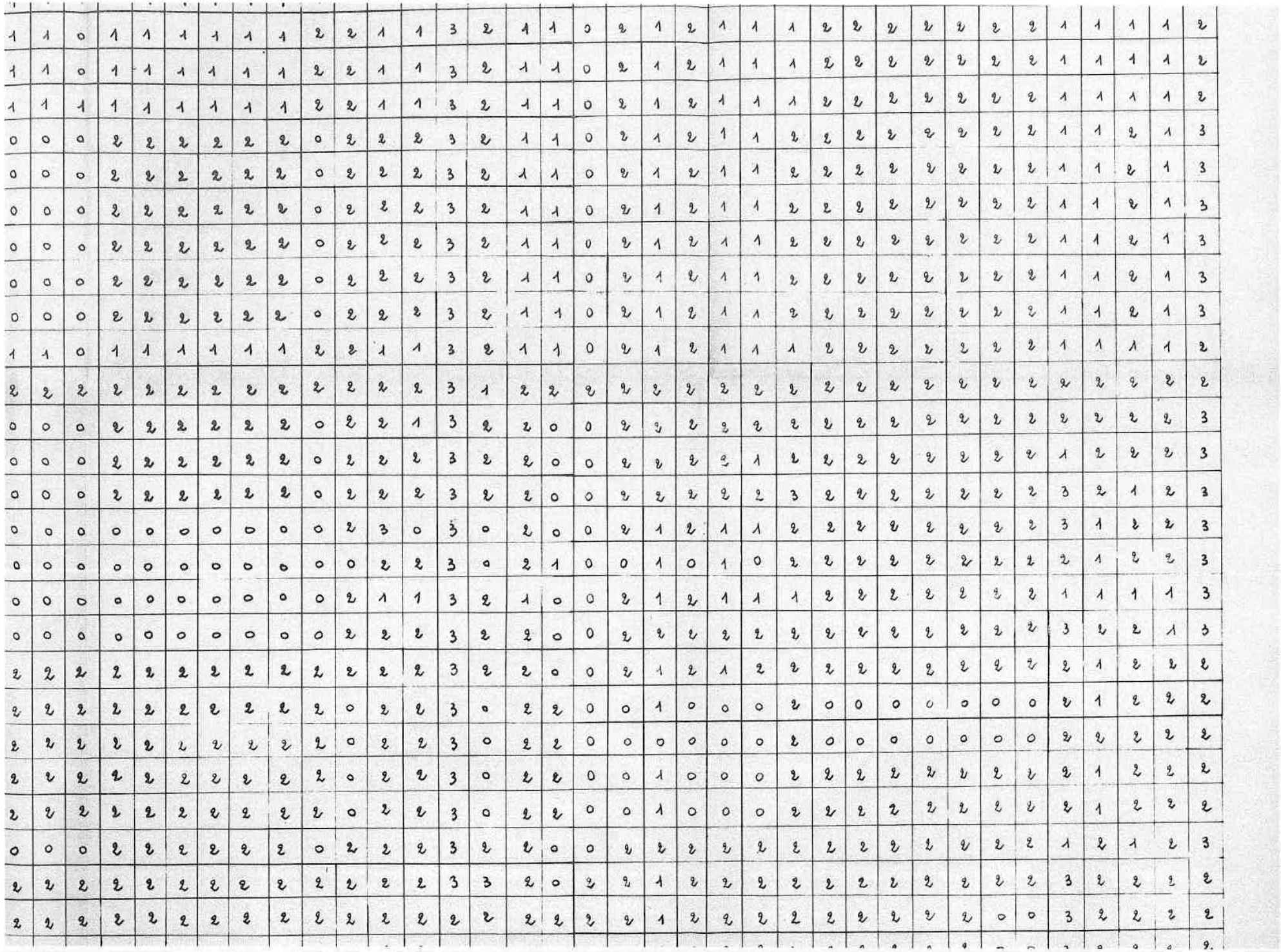


TRICE de PRIORITE



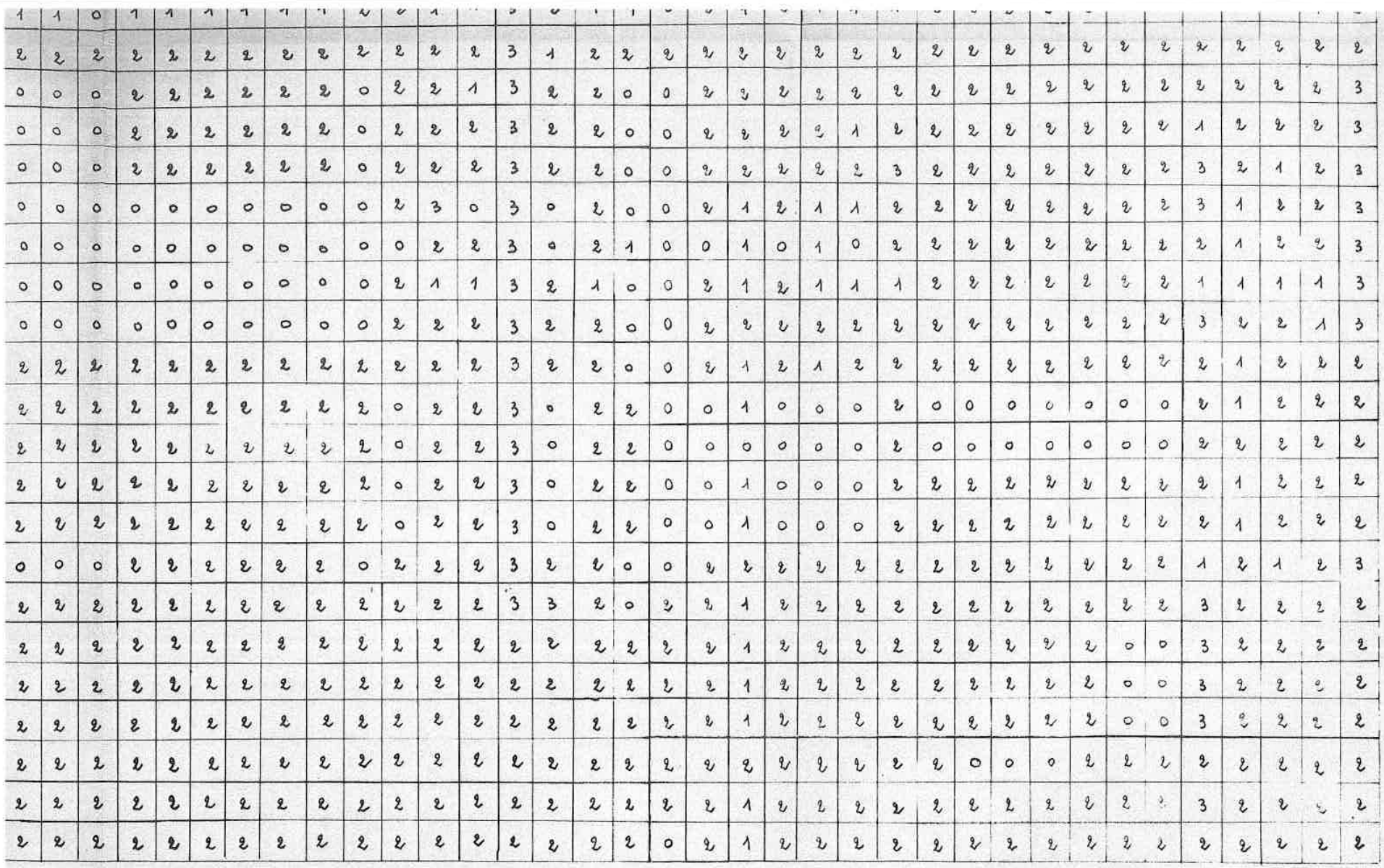
-D	÷	<	≤	=	≥	>	≠	-U	POUR	PAS	JUSQUA	[	:=	ALORS	SINON EXP.	(	ALLER A	;	DEBUT	SINON INST	FAIRE	:	AIG	REEL	ENTIER	BOOL.	REMANENT	TABLEAU	PROC	,	FIN	]	)	+U
0	0	0	0	0	0	0	0	0	2	2	2	3	2	1	1	0	2	1	2	1	1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	0	0	0	0	0	0	0	2	2	2	3	2	1	1	0	2	1	2	1	1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	0	0	0	0	0	0	0	2	2	2	3	2	1	1	0	2	1	2	1	1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	0	0	0	0	0	0	0	2	2	2	3	2	1	1	0	2	1	2	1	1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	0	0	0	0	0	0	0	2	2	2	3	2	1	1	0	2	1	2	1	1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	0	0	0	0	0	0	0	2	2	2	3	2	1	0	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3
0	0	0	0	0	0	0	0	0	2	2	2	3	2	2	0	0	2	2	2	2	1	2	2	2	2	2	2	2	2	1	2	2	2	3
1	1	1	1	1	1	1	1	2	2	1	1	3	2	1	1	0	2	1	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	2
1	1	1	1	1	1	1	1	2	2	1	1	3	2	1	1	0	2	1	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	2
1	1	1	1	1	1	1	1	2	2	1	1	3	2	1	1	0	2	1	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	2
1	0	1	1	1	1	1	1	2	2	1	1	3	2	1	1	0	2	1	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	2
1	0	1	1	1	1	1	1	2	2	1	1	3	2	1	1	0	2	1	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	2
1	1	1	1	1	1	1	1	2	2	1	1	3	2	1	1	0	2	1	2	1	1	1	2	2	2	2	2	2	2	1	1	1	1	2
0	0	2	2	2	2	2	2	0	2	2	2	3	2	1	1	0	2	1	2	1	1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	2	2	2	2	2	2	0	2	2	2	3	2	1	1	0	2	1	2	1	1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	2	2	2	2	2	2	0	2	2	2	3	2	1	1	0	2	1	2	1	1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	2	2	2	2	2	2	0	2	2	2	3	2	1	1	0	2	1	2		1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	2	2	2	2	2	2	0	2	2	2	3	2	1	1	0	2	1	2		1	2	2	2	2	2	2	2	2	1	1	2	1	3
0	0	2	2	2	2	2	2	0	2	2	2	3	2	1	1	0	2	1	2			2	2	2	2	2	2	2	2	1	1	2	1	3
1	0	1	1	1	1	1	1	2	2	1	1	3	2	1	1	0	2	1	2		1	1	2	2	2	2	2	2	2	1	1	1	1	2
2	2	2	2	2	2	2	2	2	2	2	2	3	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2


+D
-D
÷
<
≤
=
≥
>
≠
-U
POUR
PAS
JUSQUA
[
:=
ALORS
SINON EXP.
(
ALLER A
;
DEBUT
SINON INST.
FAIRE
:
AIGUILLAGE
REEL
ENTIER



POUR	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	1	2	2	2	2	2	2	2		
PAS	2	2	2	2	2	0	2	0	0	0	0	0	0	2	2	2	2	2	2	0	2	2	1	3	2	2	0	0	2	2	2	2		
JUSQUA	2	2	2	2	2	0	2	0	0	0	0	0	0	2	2	2	2	2	2	0	2	2	2	3	2	2	0	0	2	2	2	2		
[	2	2	2	2	2	0	2	0	0	0	0	0	0	2	2	2	2	2	2	0	2	2	2	3	2	2	0	0	2	2	2	2		
:=	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	0	3	0	2	0	0	2	1	2	1		
ALORS	0	0	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	3	0	2	1	0	0	1	0	1		
SINON EXP.	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	1	3	2	1	0	0	2	1	2	1		
(	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	3	2	2	0	0	2	2	2	2		
ALLER A	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	2	2	0	0	2	1	2	1		
;	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2	3	0	2	2	0	0	1	0	0	
DEBUT	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2	3	0	2	2	0	0	0	0	0		
SINON INST.	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2	3	0	2	2	0	0	1	0	0		
FAIRE	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	2	3	0	2	2	0	0	1	0	0		
:	2	2	2	2	2	0	2	0	0	0	0	0	0	2	2	2	2	2	2	0	2	2	2	3	2	2	0	0	2	2	2	2		
AIGUILLAGE	2	2	2	2	2	0	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	2	0	2	2	1	2	2		
REEL	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2		
ENTIER	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2		
BOOLEEN	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2		
REMANENT	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2		
TABLEAU	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	2	2		
PROC.	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	0	2	1	2	2		

REPRESENTATION en MEMOIRE de la MATRICE de PRIORITE



REPRESENTATION en MEMOIRE de la MATRICE de PRIORITE