



NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

**THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED**

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

**LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE**

MICROPROGRAMMATION ET TRANSDUCTEURS
D'ARBORESCENCES ET DE GRAPHS

par

Philip G. Churchill

Cette thèse fut soumise à l'Ecole des Etudes Supérieures
conformément aux exigences du degré de
Maîtrise en Science Appliquée.

Département de Génie Electrique
Faculté de Sciences et de Génie
Université d'Ottawa
Ottawa , Ontario

Avril 1980



P.G. Churchill, Ottawa, Canada, 1980

REMERCIEMENTS

J'aimerais, par la présente, exprimer mes sincères remerciements au Dr. Jacques Raymond, mon conseiller sur cette thèse, ainsi qu'à Benoît Thouin, pour leurs innombrables conseils et suggestions, leur support et leur encouragement.

De plus, je suis très reconnaissant de l'aide fournie par Luc Laplante en ce qui a trait à la vérification de l'utilisation du système développé.

Enfin, je voudrais exprimer mon appréciation à tous les membres du Département d'Informatique ainsi qu'au personnel du Centre de Calcul de l'Université d'Ottawa pour leur assistance au cours des dernières années.

RESUME.

Cette thèse présente un jeu d'instructions conçu spécialement pour les algorithmes qui manipulent des structures de graphes et d'arborescences. Ce jeu d'instructions machine incorpore des modes d'adressage ainsi que des instructions spéciales qui traitent exclusivement la représentation interne d'arbres et de graphes. De toutes les représentations possibles, la représentation binaire des graphes a été choisie comme structure de données de base puisqu'elle est simple, chaque noeud n'exigeant que deux liens comme pour les arbres binaires, et peut être utilisée pour représenter tout arbre ou forêt d'arbres. Cette représentation particulière est soutenue par quatre modes d'adressage ainsi que des instructions permettant;

- a) la création de structures d'arborescences,
- b) le parcours de ces structures et la recherche d'éléments particuliers,
- c) la comparaison de deux arbres, de deux listes de sous-arbres ou de deux éléments de noeud.

TABLE DES MATIERES

CHAPITRES.

1. INTRODUCTION	1
2. SYSTEME-Q	2
2.1 Eléments de base	2
2.2 Désintégrateur d'étiquette	4
2.3 Arbres et listes généralisés	5
2.4 Relations entre listes	6
2.5 Graphes d'arborescences	8
2.6 Règles	9
2.7 Traitement-q	12
3. TRAITEMENT PAR ORDINATEUR	13
3.1 Compilateur	13
3.2 Interpréteur	17
4. REPRESENTATION INTERNE	22
4.1 Représentation interne d'une règle	22
4.1.1 Descripteurs	24
4.1.2 Arbres généralisés	26
4.1.3 Expression booléenne	30
4.2 Représentation interne d'un graphe	34
4.3 Adresses	37
4.4 Etiquettes	38
4.5 Variables	39

5. DESCRIPTION DU SYSTEME	42
5.1 Mémoire	42
5.2 Compteur d'instructions	43
5.3 Pile	44
5.4 Registres	44
5.5 Indicateurs de présence des variables	45
5.6 Code conditionnel	45
5.7 Interruptions	47
5.8 Entrées / Sorties	49
5.9 Modes d'adressage	51
5.10 Adressage relatif	54
5.11 Jeu d'instructions	54
5.11.1 Instructions de contrôle	56
5.11.2 Instructions de comparaison	68
5.11.3 Instructions de gestion des pointeurs ..	82
5.11.4 Instructions de gestion des variables ..	96
5.11.5 Instructions d'entrées / sorties	100
6. CONCLUSION	116

APPENDICES.

A. LISTE DU PROGRAMME	117
B. LANGUAGE ASSEMBLEUR	118
B.1 Language source	118
B.2 Constantes	119
B.3 Adressage	120
B.4 Instructions machine	120
B.5 Commandes assembleur	123
B.6 Abréviations	124

CHAPITRE 1.

INTRODUCTION.

La structure d'arborescences constitue la plus importante structure non-linéaire utilisée dans les algorithmes sur ordinateur. Une des maintes applications employant cette structure consiste en la traduction automatique de langues écrites. Le système proposé dans cette thèse fut conçu spécialement pour manipuler les structures de données et les règles de transformation impliquées dans la traduction automatique. Il est basé sur le SYSTEME-Q (Ref. 3), qui consiste en un ensemble de règles de transformation s'appliquant bien à la traduction automatique, mais peut servir à de nombreuses applications nécessitant des structures de graphes et d'arborescences.

La machine proposé consiste donc en un jeu d'environ 40 instructions permettant la manipulation de chaînes de caractères, la création et le parcours de structures d'arborescences ainsi que la comparaison d'éléments ayant de telles structures. De plus, il permet des appels récursifs aux sous-programmes. Le jeu d'instructions fut implanté sous forme de micro-programmes sur un ordinateur MICRODATA 1600.

CHAPITRE 2. SYSTEME-Q.

Un système-q est un ensemble de règles permettant de faire certaines transformations sur des graphes orientés. Chaque flèche d'un tel graphe est surmontée d'une expression parenthésée représentant tout un arbre de sorte que le graphe représente un ensemble de suites d'arbres. Une transformation effectuée par un système-q sur un graphe consiste à remplacer une suite d'arbres par une autre.

2.1 Eléments de base.

La chaîne de caractères, l'arbre et la liste constituent les éléments de base d'un système-q. Toute chaîne de caractères (ou étiquette) est représentée par une suite de caractères dont le premier seulement peut être quelconque. Les autres caractères de la chaîne doivent être des chiffres ou des lettres.

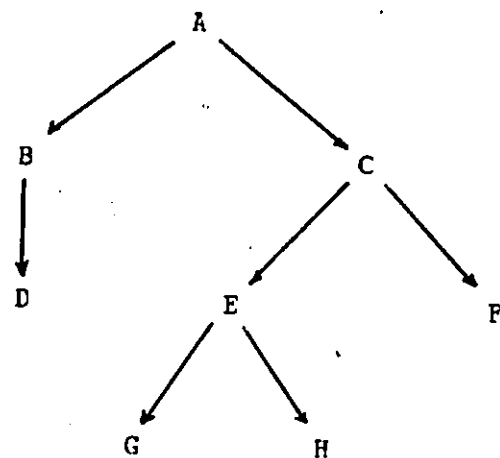
Exemples:

A -PL1 ETIQUETTE /CCDE ,B

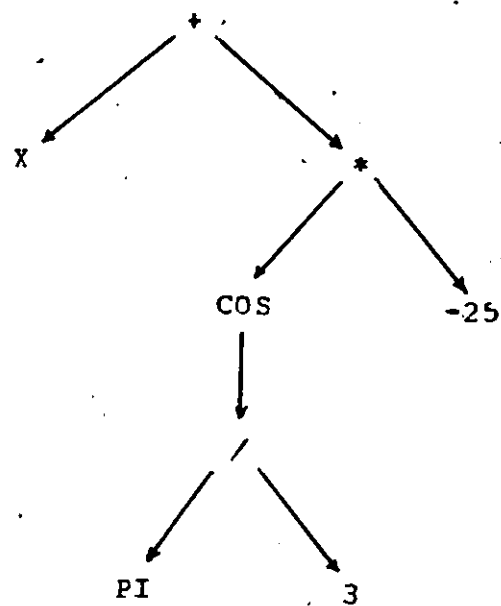
L'arbre est une représentation des relations entre les différentes étiquettes. Tout arbre peut correspondre à un schéma du genre de ceux donnés dans les exemples qui suivent.

Exemples:

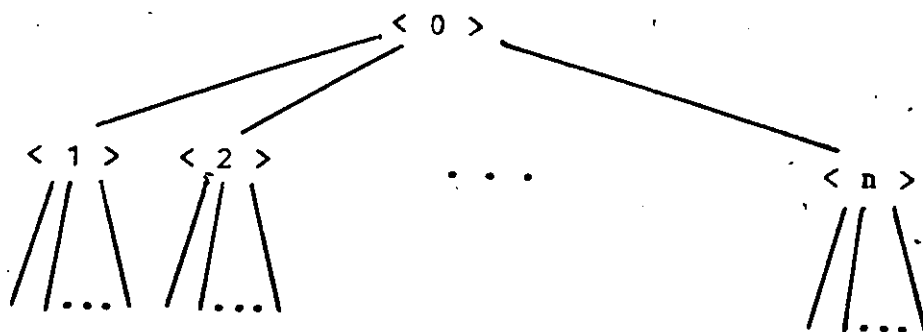
(a)



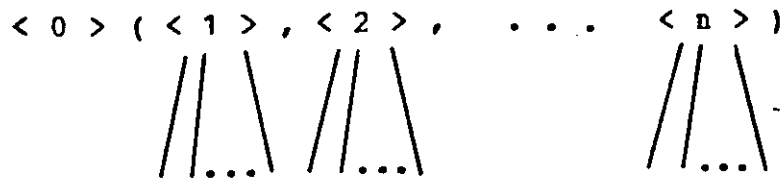
(b)



Tout arbre est représenté par une expression formée d'étiquettes, de parenthèses et de virgules. Une telle expression se construit à partir d'un schéma d'arbre en remplaçant autant de fois que nécessaire la configuration



par la configuration



Les schémas de l'exemple précédent correspondent donc aux arbres

(a) $A (B (D) , C (E (G , H) , F))$

(b) $+ (X , * (\cos (/ (\pi , 3)) , -25))$

Enfin, une suite d'arbres séparés par des virgules constitue une liste. De plus, le symbole -NUL- indique la liste vide.

Exemples:

$+ (X , Y) , \sin (* (\pi , .5))$

$A , B (C , D (E)) , F (G)$

$X (Y) , Z , -NUL-$

2.2 Désintégrateur d'étiquette.

Lorsqu'une liste est composée de sous-arbres qui se réduisent à un seul caractère, il existe une autre façon de la représenter. Il s'agit d'utiliser l'opérateur \$\$, appelé désintégrateur, qui transforme une étiquette en une liste de caractères. Par exemple, la liste

A , B , C , D , E

peut être remplacée par la représentation équivalente

$$$ABCDE$

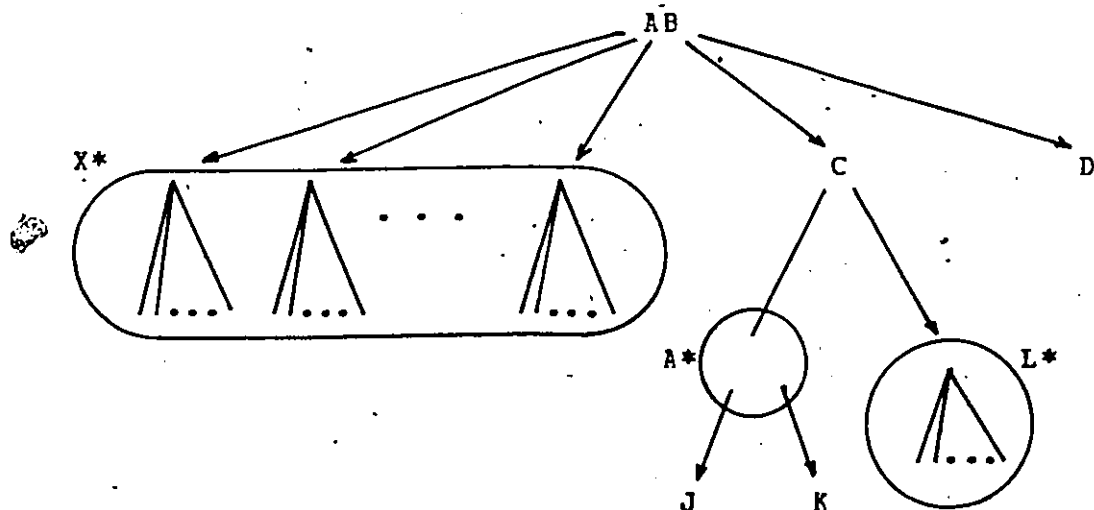
2.3 Arbres et listes généralisés.

Un arbre généralisé est un arbre dans lequel apparaissent certaines variables qui représentent des étiquettes, des arbres ou des listes encore inconnus. Ces variables se composent d'une lettre suivie d'un astérisque qui peut être suivi ou non d'un chiffre. Si la lettre est comprise entre A et F, la variable représente une étiquette. Si elle est comprise entre I et L, la variable représente un arbre. Enfin, si la lettre est comprise entre U et Z, la variable représente une liste. Il existe donc trois types de variables, soit les variables étiquettes, les variables arbres et les variables listes.

Par exemple, l'arbre généralisé

$AB (X^*, C (A^* (J, K), L^*), D)$

peut être représenté par le schéma



Si on substitue respectivement aux variables X^* , A^* et L^* les expressions -NUL-, INV et $F (X, Y (Z))$, on obtient l'arbre

AB (C (INV (J, K), F (X, Y (Z))), D) .

2.4 Relations entre des listes.

On définit quatre relations binaires entre des listes, soit les relations =, $\neq$, -DANS- et -HORS-. Les relations = et $\neq$ correspondent respectivement à l'égalité et à l'inégalité entre deux listes. Par exemple, les relations

I, J, K, L, M, N = \$\$IJKLMN
A (-NUL-) = A
AB $\neq$ AB (X)

ont toutes la valeur VRAI, alors que la relation

X (C, D, E) = X (C, I*)

a la valeur VRAI seulement si I* représente la liste D, E. Il est à noter que les relations = et $\neq$ dépendent de l'ordre et de la répétition des éléments. Par exemple, les relations

A, B = B, A
X, Y(Z), X = Y(Z), X, X

ont toutes deux la valeur FAUX.

Les relations -DANS- et -HORS- sont plus complexes. Si on utilise e1 et e2 pour représenter deux étiquettes quelconques, a1 et a2 pour représenter deux arbres quelconques, l1 et l2 pour représenter deux listes quelconques et le symbole = pour représenter l'équivalence, la relation -DANS- se définit récursivement par

-NUL- -DANS- 12 = VRAI
 11 -DANS- -NUL- = 11 = -NUL-
 a1, 11 -DANS- 12 = a1 -DANS- 12 et 11 -DANS- 12
 a1 -DANS- a2, 12 = a1 -DANS- a2 ou a1 -DANS- 12
 e1(11) -DANS- e2(12) = e1 = e2 et 11 -DANS- 12

alors que la relation -HORS- se définit récursivement par

-NUL- -HORS- 12 = FAUX
 11 -HORS- -NUL- = FAUX
 a1, 11 -HORS- 12 = a1 -HORS- 12 et
 (11 -HORS- 12 ou 11 = -NUL-)
 a1 -HORS- a2, 12 = a1 -HORS- a2 et
 (a1 -HORS- 12 ou 12 = -NUL-)
 e1(11) -HORS- e2(12) = e1 ≠ e2 ou 11 -HORS- 12

Notez que la relation -HORS- n'est pas l'exacte négation de la relation -DANS-. Par exemple, les relations

A, B(A, B), C -DANS- B(E(A), A), A(C) et
 A, B(A, B), C -HORS- B(B(A), A), A(C)

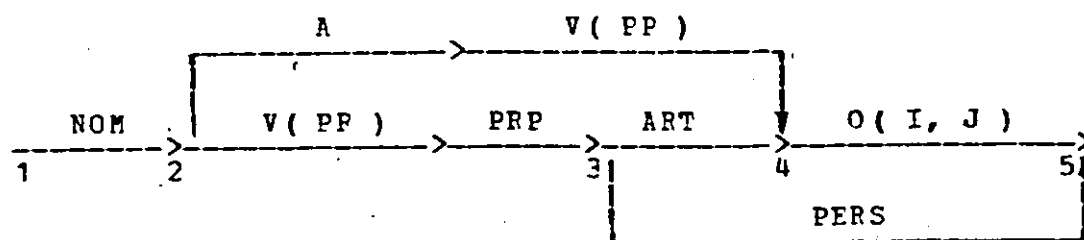
ont toutes deux la valeur FAUX. Voici maintenant des exemples dans lesquels les relations -DANS- et -HORS- sont vérifiées:

A, D, B, A -DANS- A, B, C, D, E, F, G
 A, B(A, B), A -DANS- E(B(A), A), A(C)
 A, B(A, A), A -HORS- E(E(A), B), B(C)

Contrairement aux relations = et ≠, les relations -DANS- et -HORS- sont indépendantes de l'ordre et de la répétition des éléments.

2.5 Graphes d'arborescences.

Un graphe d'arborescences constitue une façon économique de représenter un ensemble de suite d'arbres. Un tel graphe est composé d'un ensemble de flèches, chacune surmontée d'un arbre, qui relie entre eux un ensemble de sommets, de sorte qu'il ne contient pas de circuit et que toute flèche du graphe figure dans un chemin allant d'un unique sommet d'entrée à un unique sommet de sortie. Voici un exemple d'un tel graphe:



Ce graphe représente l'ensemble des suites d'arbres

NOM + A + V(PP) + O(I, J)

NOM + V(PR) + PRP + ART + O(I, J)

NOM + V(PR) + PRP + PERS

dans lesquels tous les arbres d'une même suite sont séparés par le signe plus (+). Formellement, ce graphe peut être représenté par

-1- NCM -2-

-2- V(PR) + PRP -3-

-2- A + V(PP) -4-

-3- ART -4-

-3- PERS -5-

-4- O(I, J) -5- .

Notez que seuls les sommets d'entrée et de sortie ainsi que

ceux qui se situent à l'intersection de plus de deux flèches sont représentés par des entiers positifs.

2.6 Règles.

Un système-q se compose d'un ensemble de règles. Dans leur forme simplifiée, ces règles s'écrivent

$$a_1 + a_2 + \dots + a_M == b_1 + b_2 + \dots + b_N .$$

Les a_i et les b_i représentent des arbres quelconques et les suites $a_1 + a_2 + \dots + a_M$ et $b_1 + b_2 + \dots + b_N$ sont respectivement le membre gauche et le membre droit de la règle. Le symbole $==$ peut se traduire par la phrase "se transforme en".

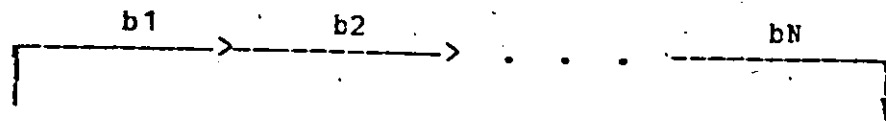
L'ensemble des règles qui s'appliquent sur un graphe le transforme en un nouveau graphe en deux phases distinctes. Dans la première phase, on considère toute suite d'arbres contenue dans le graphe. Pour toute suite d'arbres dans le graphe qui correspond au membre gauche d'une règle dans le système-q considéré, on y ajoute en parallèle la suite d'arbres représentée par le membre droit de cette règle. Par exemple, étant donnée la suite

$$\xrightarrow{a_1} \xrightarrow{a_2} \dots \xrightarrow{a_M}$$

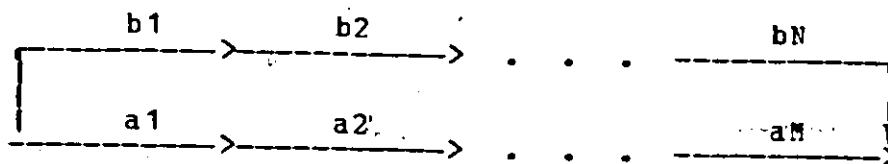
appartenant au graphe, si la règle

$$a_1 + a_2 + \dots + a_M == b_1 + b_2 + \dots + b_N .$$

figure dans le système-q considéré, on l'applique sur ce chemin en y ajoutant de son entrée à sa sortie la nouvelle suite



pour obtenir



Il est à noter que la même règle ne s'applique qu'une seule fois sur le même chemin, de sorte que le processus se poursuit jusqu'à ce qu'il n'y ait plus de règles à appliquer. Il est cependant possible que le processus ne converge pas et dans ce cas, le résultat est indéfini. Autrement, le résultat est indépendant de l'ordre dans lequel les règles sont appliquées.

Dans la deuxième phase, on supprime d'abord toute flèche figurant sur un chemin ayant servi à l'application d'au moins une règle. On supprime ensuite toute flèche qui ne figure plus dans un chemin allant de l'entrée à la sortie du graphe.

Soit par exemple le système²q

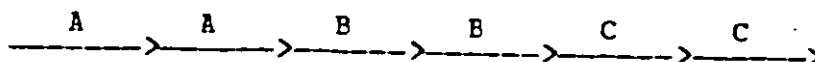
$$A + B + C == S .$$

$$A + S + B(*) + C == S .$$

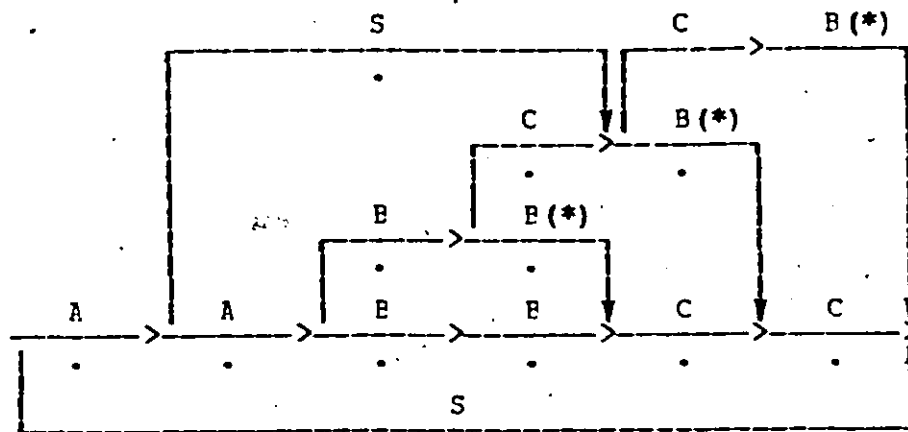
$$B(*) + C == C + B(*) .$$

$$E + B == B + B(*) .$$

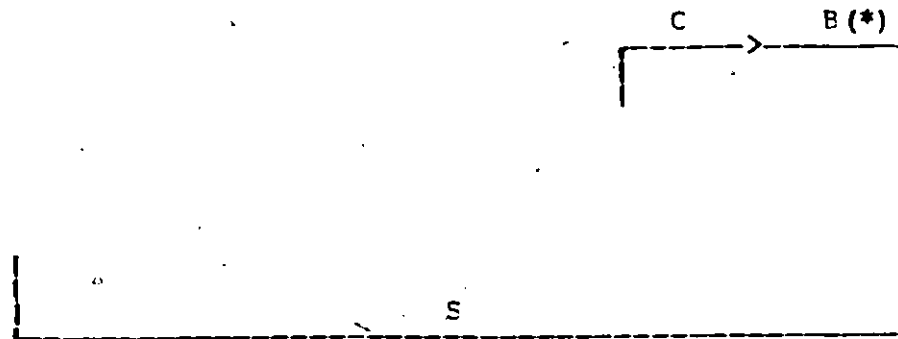
à être appliqué sur le graphe



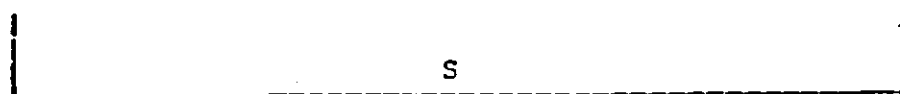
Après la première phase, on obtient



Les points servent à indiquer les flèches ayant servi à l'application d'une règle. Au milieu de la deuxième phase, on obtient



pour enfin obtenir le graphe résultant



Dans leur forme générale, les règles d'un système-q peuvent être suivies d'une condition sous forme d'une expression booléenne. Les opérateurs permis dans cette expression sont les opérateurs booléens classiques -OU-, -ET- et -NON- ainsi que les relations =, $\neq$, -DANS- et -HORS- définis précédemment. Une règle suivie d'une telle condition ne peut s'appliquer que si la condition est satisfaite, c'est-à-dire lorsque l'expression booléenne a la

valeur VRAI.. Par exemple, la règle

$$A* + B == A*(B) / A* -DANS- A, B, C .$$

correspond aux règles simplifiées

$$A + B == A(B) .$$

$$B + B == B(B) .$$

$$C + B == C(B) .$$

puisque'elle ne s'applique que si la variable étiquette A* représente l'étiquette A, l'étiquette B ou l'étiquette C.

2.7 Traitement-q.

Il est souvent souhaitable de transformer un graphe en plusieurs phases, où chaque phase consiste à appliquer un système-q différent. Il est donc possible d'enchaîner plusieurs systèmes-q en une suite de phases; la première s'applique sur le graphe de départ et chaque autre phase s'applique sur le résultat de la phase précédente. Une telle suite de phases est appelée traitement-q.

Les différentes phases d'un traitement-q sont exécutées dans l'ordre dans lequel elles apparaissent et le mot -REQ- est utilisé pour les séparer. L'ensemble des règles d'un traitement-q est généralement appelé une grammaire, alors qu'une sous-grammaire fait référence à l'ensemble des règles d'un système-q particulier.

Ce chapitre décrit une implantation particulière des systèmes-q sur ordinateur. Dans cette implantation, le traitement des systèmes-q se fait en deux étapes distinctes, soit l'étape de compilation et l'étape d'interprétation. Dans la première, les règles de chaque sous-grammaire sont codées en mémoire sous forme d'arborescences. Dans la seconde, le ou les graphes initiaux sont d'abord codés et pour chaque phase, les règles de la sous-grammaire sont appliquées sur le résultat de la phase précédente.

3.1 Compilateur.

Le compilateur accepte en entrée une grammaire complète, composée d'une ou plusieurs sous-grammaires, qui peut être définie formellement comme suit:

$\langle \text{traitement-q} \rangle ::= \langle \text{phase} \rangle \mid \langle \text{phase} \rangle \text{-REQ-}$
 $\langle \text{traitement-q} \rangle$

$\langle \text{phase} \rangle ::= \langle \text{détails} \rangle \langle \text{inversion} \rangle \langle \text{système-q} \rangle$

$\langle \text{détails} \rangle ::= \mid \text{-DET-}$

$\langle \text{inversion.} \rangle ::= \mid \text{-INV-}$

$\langle \text{système-q} \rangle ::= \mid \langle \text{système-q} \rangle \langle \text{règle} \rangle \mid$
 $\langle \text{système-q} \rangle \langle \text{commentaire} \rangle$

$\langle \text{commentaire} \rangle ::= \langle \text{texte} \rangle .$

< texte > ::= ** | < texte > < caractère autre
que point et égal >

< caractère autre que point et égal > ::= + | - |
* | / | (|) | \$ | % | , | < lettre > |
< chiffre > | < signe non significatif >

Il produit en sortie un fichier contenant la grammaire codée sous forme d'arborescences ainsi qu'une liste imprimée de toutes les règles de la grammaire. Si la grammaire contient des erreurs, certains messages d'erreurs seront imprimés en plus. La FIGURE 3.1a illustre cette étape de compilation.

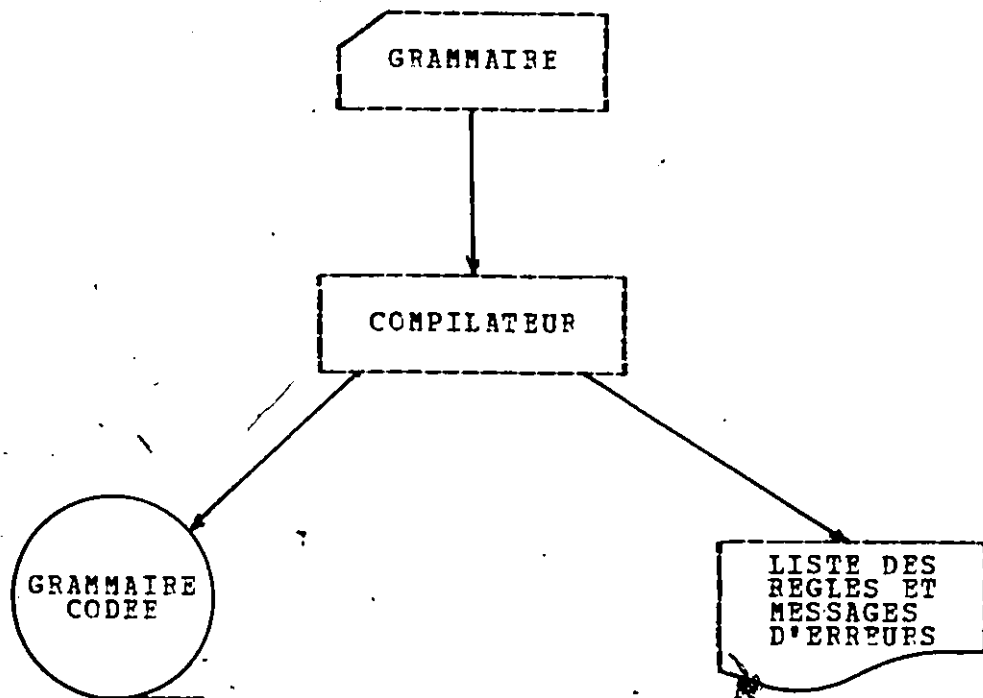


FIGURE 3.1a

En mémoire, les règles de chaque sous-grammaire sont classées sous forme de liste à chaînage simple. L'étiquette racine du dernier arbre du membre gauche de la règle constitue le critère de classement. Cette organisation des

-REQ-

$A + A^* == B(A^*) / A^* -DANS- A, B, C .$

$C + D == CD .$

$E + F == EF .$

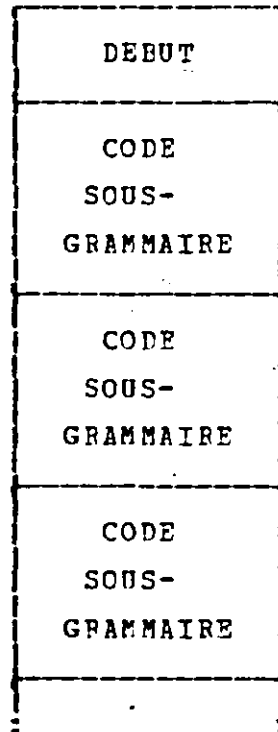
$G + I^* == G(I^*) .$

$D + F(G) == DFG .$

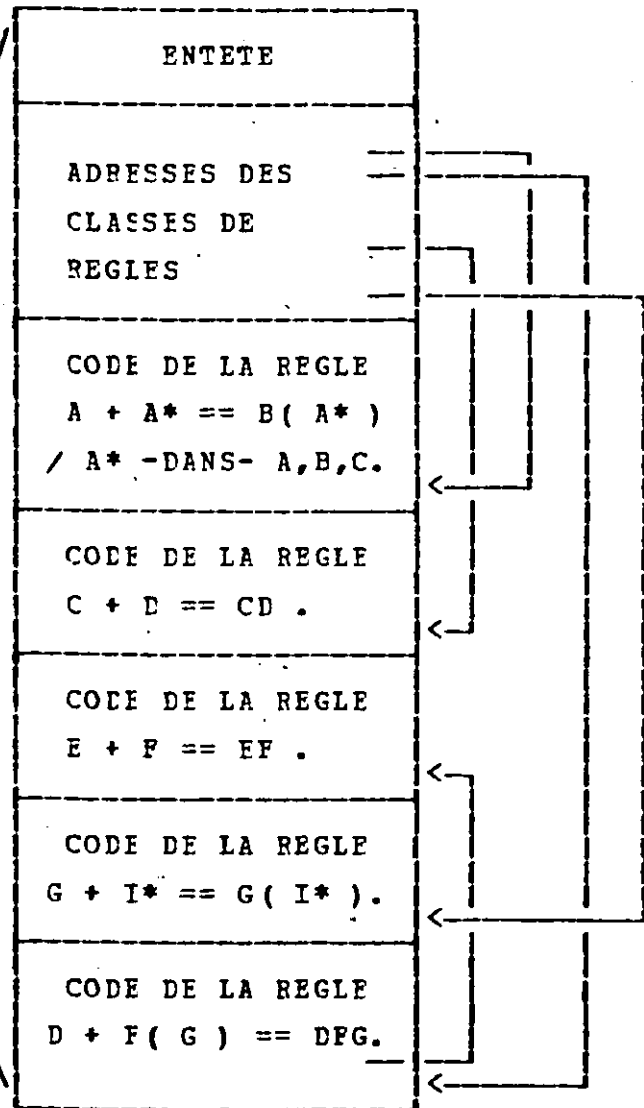
-REQ-

:

* GRAMMAIRE *



* CODE GRAMMAIRE *



* CODE SOUS-GRAMMAIRE *

FIGURE 3.1b

règles d'une même sous-grammaire facilite leur consultation puisqu'elle limite le nombre de règles à vérifier pour chaque application. En effet, chaque règle peut appartenir à une seule parmi 257 classes, soit 256 classes ordinaires et une classe spéciale pour les règles dont le dernier arbre du membre gauche est représenté par une variable. Le code de chaque sous-grammaire comprend donc une zone de 257 pointeurs, associés à chacune des 257 classes, qui indiquent les adresses de la première règle de chaque classe. S'il n'existe aucune règle d'une telle classe dans la sous-grammaire, le pointeur associé à cette classe contient l'adresse NULLE, qui représente un pointeur NIL.

La FIGURE 3.1b illustre cette organisation du code d'une sous-grammaire. Cette sous-grammaire comprend cinq règles. Les règles $E + F == EF$. et $D + F(G) == DFG$. appartiennent à la même classe puisque les étiquettes racines du dernier arbre de leur membre gauche sont identiques (soit F). Le pointeur associé à cette classe indique donc l'adresse de la règle $D + F(G) == DFG$. alors que l'adresse de la règle $E + F == EF$. est indiquée par le pointeur associé à la règle $D + F(G) == DFG$. . S'il existait une autre règle de cette même classe, elle serait insérée au début de la liste des règles de cette classe et enchaînée à la règle $D + F(G) == DFG$. . Enfin, la règle $G + I^* == G(I^*)$. appartient à la classe spéciale puisque le dernier arbre de son membre gauche est représenté par une variable arbre (soit I^*).

Il est à noter que si la sous-grammaire illustrée dans la FIGURE 3.1b était précédée du mot -INV-, chacune de ses règles serait inversée, le membre gauche devenant le membre droit et vice versa. Dans ce cas, les critères de classement des règles deviendraient les étiquettes racines du dernier arbre du membre droit, soit B, CD, EF, G et DFG.

3.2 Interpréteur.

L'interpréteur accepte en entrée le code grammaire généré par le compilateur ainsi que les graphes initiaux, qui peuvent être définis formellement comme suit:

$$\begin{aligned} < \text{données d'un traitement-q} > ::= < \text{graphe} \\ & \text{d'arborescences} > \mid < \text{graphe d'arborescences} > \\ & / < \text{données d'un traitement-q} > \end{aligned}$$

Il produit en sortie un fichier contenant le graphe résultant sous forme codé en plus d'une liste imprimée de ce graphe ainsi que du détail d'exécution, s'il est demandé. Le détail d'exécution n'est imprimé que si la sous-grammaire est précédée du mot -DET-. La FIGURE 3.2 illustre cette étape d'interprétation.

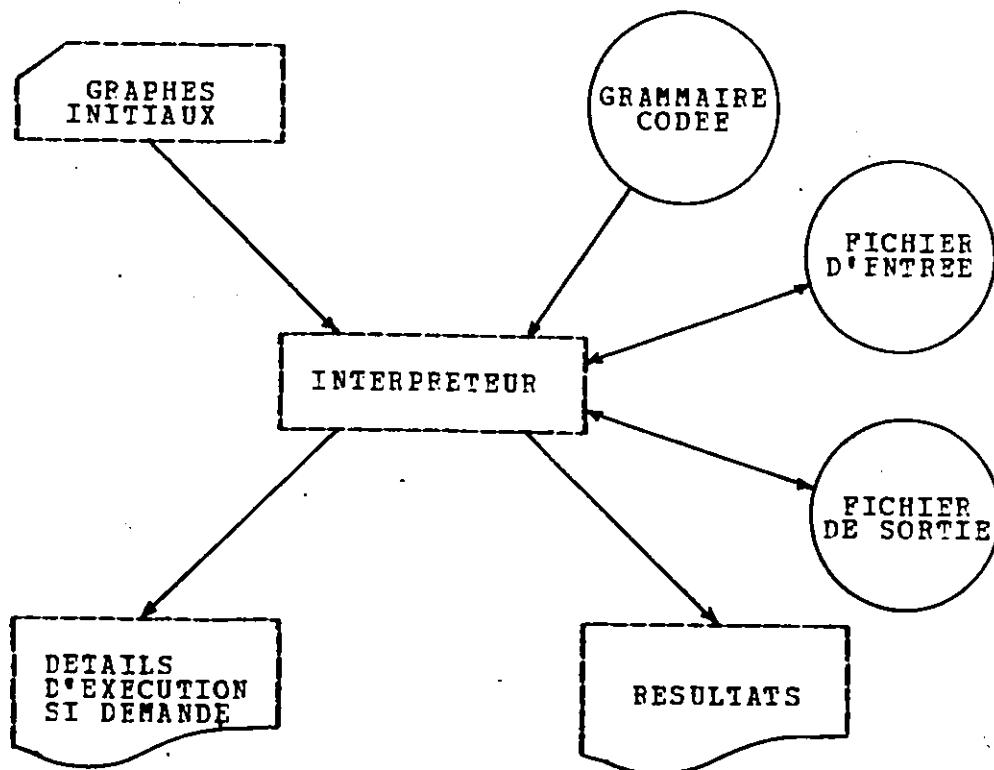


FIGURE 3.2

L'étape d'interprétation est répétée autant de fois qu'il y a de sous-grammaires dans le traitement-q. Initialement, les graphes d'entrée sont sur cartes. Avant l'exécution de la première phase, ils sont transférés sur le fichier d'entrée. Les règles de la première sous-grammaire leur sont ensuite appliquées et les graphes résultants sont placés sur le fichier de sortie. S'il existe une autre sous-grammaire, les fichiers d'entrée et de sortie sont interchangés de sorte que les graphes de sortie de la phase précédente deviennent les graphes d'entrée de la phase courante. Ce processus se poursuit jusqu'à ce que les règles de la dernière sous-grammaire aient été appliquées.

L'application de règles sur un graphe d'entrée constitue la phase la plus complexe du traitement des systèmes-q par ordinateur. Le parcours du graphe doit se faire d'une manière systématique afin d'assurer la vérification de toutes les possibilités tout en évitant l'application multiple d'une même règle sur le même chemin du graphe. A cette fin, le parcours du graphe s'effectue de droite à gauche et le sommet de sortie constitue le point de départ. Une pile est utilisée pour retenir les éléments du trajet parcouru qui n'ont pas encore été examinés. Les éléments examinés sont retenus à l'aide d'un indicateur spécial associé à chacun d'eux.

Initialement donc, le graphe est traversé à partir du sommet de sortie, suivant le parcours le plus direct, et tous les éléments du trajet parcouru sont placés sur la pile. L'application même des règles n'est effectuée que lorsque le parcours aboutit soit au sommet d'entrée, soit à un élément du graphe déjà examiné. Lorsqu'un de ces cas se présente, le dernier élément placé sur la pile en est retiré et la vérification de l'application possible d'une ou plusieurs règles sur tout chemin du graphe qui se termine

par cet élément est effectuée. L'élément retiré de la pile sert donc à déterminer la classe des règles à vérifier.

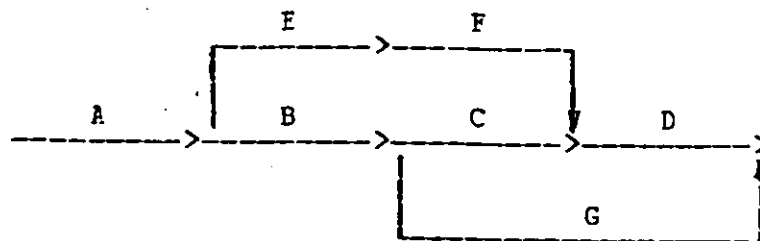
Une fois l'application des règles terminée, le parcours du graphe se poursuit en partant de l'alternant du dernier élément examiné, si cet alternant existe. L'alternant d'un élément constitue soit un chemin du graphe initial, soit un chemin ajouté par l'application d'une règle, qui aboutit au même sommet que l'élément. Si le dernier élément examiné ne possède pas d'alternant, l'élément au sommet de la pile en est retiré et le même procédé est répété. Le parcours du graphe et l'application des règles se poursuivent jusqu'à ce que le dernier élément soit retiré de la pile et qu'il ne possède pas d'alternant.

Soit par exemple le système-q simplifié

$$A + E == I + J .$$

$$B + D == H .$$

et le graphe



Le parcours de ce graphe et l'application des règles s'effectue comme suit:

- Les éléments D, C, B et A sont d'abord affectés à la pile.

- Le parcours aboutissant au sommet d'entrée, l'élément A est retiré du sommet de la pile et l'application possible de règles est vérifiée. Ici, aucune règle ne s'applique.

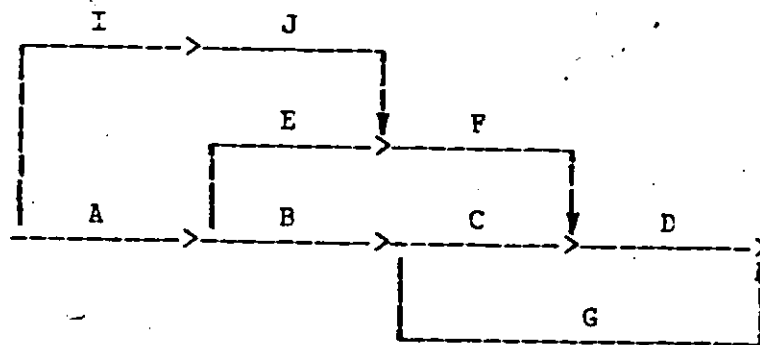
- Puisque l'élément A ne possède pas d'alternant, l'élément B est retiré du sommet de la pile. L'application possible de règles à ce stage est vérifiée et encore aucune règle ne s'applique.

- Comme auparavant, puisque l'élément B ne possède pas d'alternant, l'élément C est retiré du sommet de la pile. Encore aucune règle ne s'applique à ce stage.

- Cette fois, l'élément C possède un alternant et ce chemin est parcouru, les éléments F et E étant affectés au sommet de la pile.

- Le parcours aboutissant à un élément déjà examiné, soit A, l'élément E est retiré du sommet de la pile.

- À ce stage, la règle $A + E == I + J$ peut être appliquée, donnant comme résultat le graphe



- L'élément E possède maintenant un alternant et ce nouveau chemin est parcouru, les éléments J et I étant affectés au sommet de la pile.

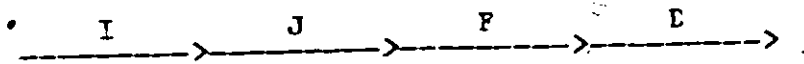
- Le parcours aboutissant au sommet d'entrée, les éléments I, puis J, puis F et enfin D sont retirés dans l'ordre du sommet de la pile et examinés, sans qu'aucune règle ne s'applique sur leurs trajets.

- La pile est maintenant vide mais comme l'élément D possède un alternant, le parcours du graphe n'est pas terminé. Le chemin alternatif est suivi et seul l'élément G est affecté au sommet de la pile.

- Le parcours aboutissant à un élément déjà examiné, soit B, l'élément G est retiré du sommet de la pile.

- A ce stage, aucune règle ne s'applique, la pile est vide et le dernier élément vérifié, soit G, ne possède pas d'alternant. Le parcours du graphe est alors terminé.

Une fois le parcours du graphe terminé, la phase d'épuration est exécutée. Le graphe est parcouru à nouveau et cette fois les pointeurs sont modifiés pour éliminer du graphe les éléments qui ont servi à l'application d'au moins une règle. Ces éléments ont été retenus lors du premier parcours à l'aide d'un indicateur spécial associé à chaque élément du graphe. Enfin, les éléments n'apparaissant plus sur un chemin allant de l'entrée à la sortie du graphe sont aussi éliminés. Pour l'exemple précédent, le graphe résultant devient donc



Ce chapitre décrit les différentes structures de données utilisées dans la représentation interne des graphes et des règles, et donne une définition formelle de chacun des éléments qui constituent un système-q.

4.1 Représentation interne d'une règle.

Chaque règle comprend un membre gauche et un membre droit suivis ou non d'une condition. La forme générale d'une règle est la suivante:

$$\langle \text{r\grave{e}gle} \rangle ::= \langle \text{membre gauche} \rangle == \\ \langle \text{membre droit} \rangle \langle \text{condition} \rangle .$$

$$\langle \text{membre gauche} \rangle ::= \langle \text{suite g\^eneralis\^ee} \rangle \mid --$$

$$\langle \text{membre droit} \rangle ::= \langle \text{suite g\^eneralis\^ee} \rangle \mid --$$

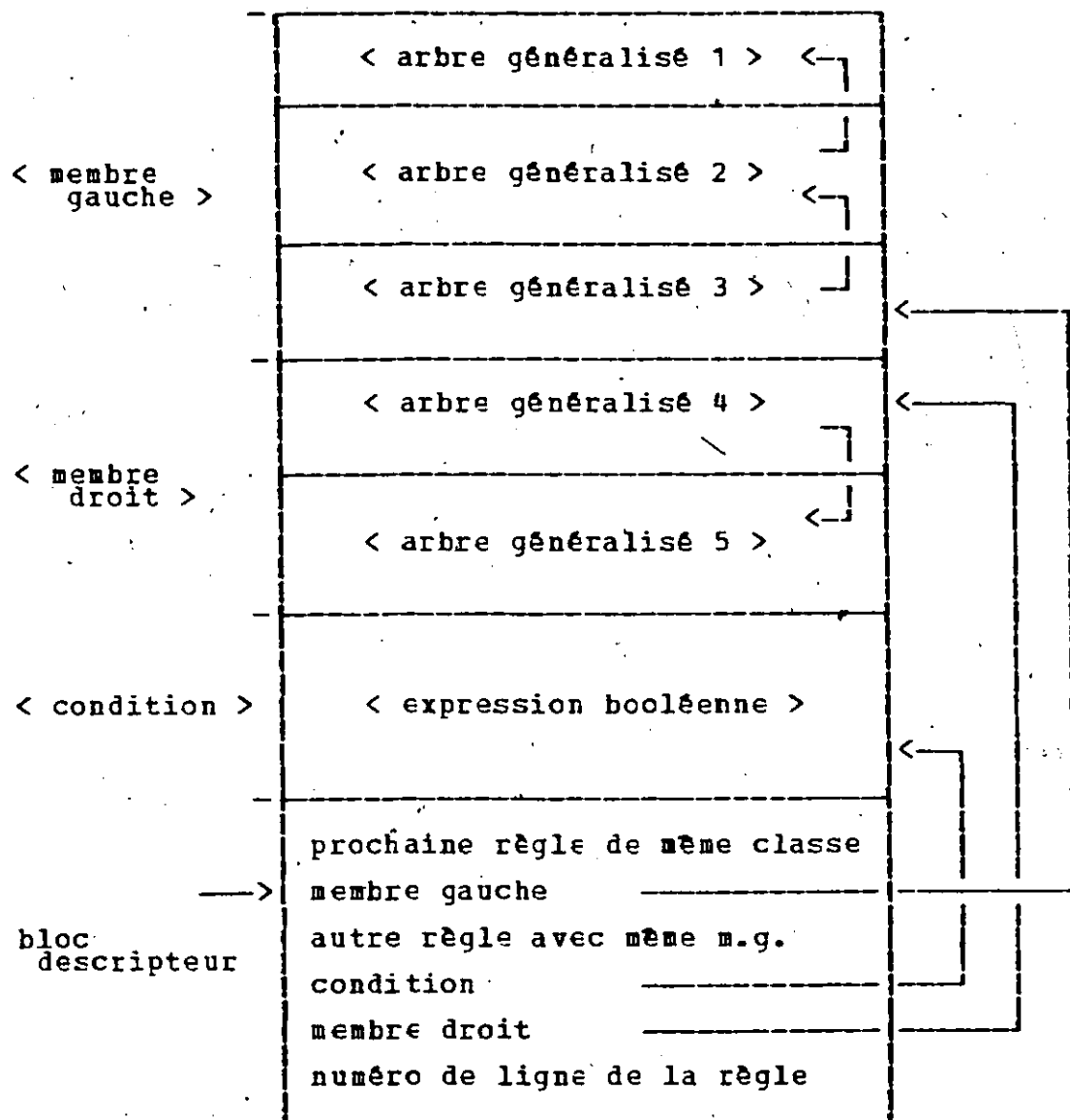
$$\langle \text{suite g\^eneralis\^ee} \rangle ::= \langle \text{arbre g\^eneralis\^e} \rangle \mid$$

$$\langle \text{arbre g\^eneralis\^e} \rangle + \langle \text{suite g\^eneralis\^ee} \rangle$$

$$\langle \text{condition} \rangle ::= \langle \text{expression bool\^eenne} \rangle$$

Lorsque le symbole -- apparaît à la place du membre gauche d'une règle, il indique que ce membre est identique au membre gauche de la règle précédente. Il en est de même pour le membre droit.

Une fois compilée, une règle codée en mémoire prend la forme représentée par la FIGURE 4.1. Chaque membre est composé d'une suite de blocs enchaînés les uns aux autres,



< ag1 > + < ag2 > + < ag3 > == < ag4 > + < ag5 >
 / < expression booléenne >.

FIGURE 4.1

représentant chacun un arbre généralisé. Les éléments du membre gauche sont enchaînés de droite à gauche alors que ceux du membre droit le sont de gauche à droite. La condition, une expression booléenne, est représentée d'une façon quelque peu différente, puisqu'elle exprime des relations entre listes. Enfin, le bloc descripteur est composé d'une série de pointeurs, donnant entre autre l'adresse en mémoire du membre gauche, l'adresse du membre droit ainsi que celle de la condition.

4.1.1 Descripteurs.

Le bloc descripteur, qui apparaît à la fin de la règle codée, contient les adresses en mémoire des différentes parties de la règle ainsi que les pointeurs servant à l'enchaînement des différentes règles. Ce bloc, représenté par la FIGURE 4.1.1, comprend cinq adresses ainsi qu'un code particulier à la règle. L'adresse du deuxième élément du bloc descripteur, soit celui qui indique l'emplacement du membre gauche, représente l'adresse de la règle elle-même.

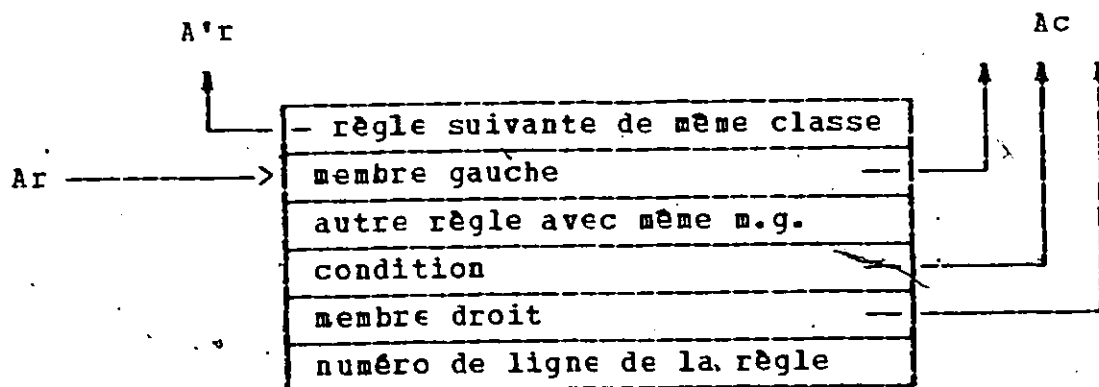


FIGURE 4.1.1

- prochaine règle de même classe.

Ce pointeur donne l'adresse de la règle suivante de même classe que la règle courante. Si cette règle existe, ce pointeur indique l'adresse du deuxième élément de son bloc descripteur.

- membre gauche.

Ce pointeur donne l'adresse du code étiquette ou du code variable de la racine du dernier arbre généralisé du membre gauche.

- autre règle avec même membre gauche.

Ce pointeur donne l'adresse d'une autre règle possédant le même membre gauche que la règle courante. Si cette règle existe, ce pointeur donne l'adresse du deuxième élément de son bloc descripteur.

- condition.

Ce pointeur donne l'adresse du code de l'opération ou celle du code de la relation du premier élément dans l'expression booléenne qui n'a pas d'ancêtres.

- membre droit.

Ce pointeur donne l'adresse du code étiquette ou du code variable de la racine du premier arbre généralisé du membre droit.

- numéro de ligne de la règle.

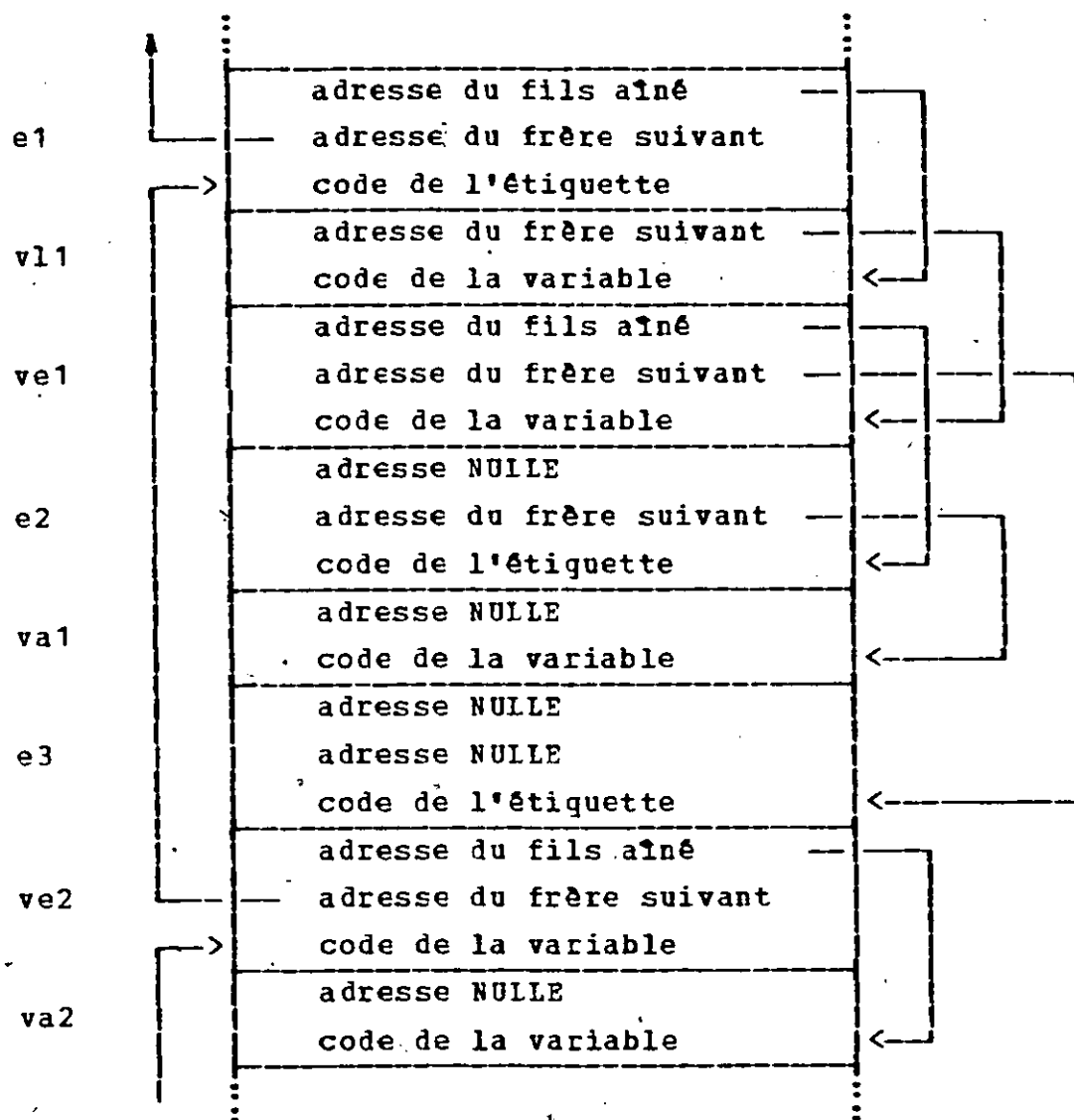
Ce code, propre à chaque règle, représente la position de la règle dans la grammaire.

4.1.2 Arbres généralisés.

Comme il a déjà été mentionné, chaque membre d'une règle est composé d'une suite d'arbres généralisés. Un arbre généralisé est un arbre dans lequel apparaissent éventuellement certaines variables qui représentent des étiquettes, des arbres ou des listes encore inconnues. La définition précise d'un arbre généralisé est:

```
< arbre généralisé > ::= < variable arbre > |  
    < étiquette généralisée > |  
    < étiquette généralisée > (< liste généralisée >)  
  
< liste généralisée > ::= < liste simple généralisée > |  
    < liste simple généralisée >, < liste généralisée >  
  
< liste simple généralisée > ::= < variable liste > |  
    -NUL- | < arbre généralisé > |  
    $$< étiquette généralisée >  
  
< étiquette généralisée > ::= < variable étiquette > |  
    < étiquette >
```

En mémoire, chaque élément d'un arbre ou d'une liste est représenté par un code ainsi que des adresses indiquant l'emplacement du fils aîné et du frère suivant de l'élément. La FIGURE 4.1.2 représente une section du membre gauche d'une règle. Cette section illustre deux arbres généralisés, soit $e1(v11, ve1(e2, va1), e3)$ et $ve2(va2)$, qui

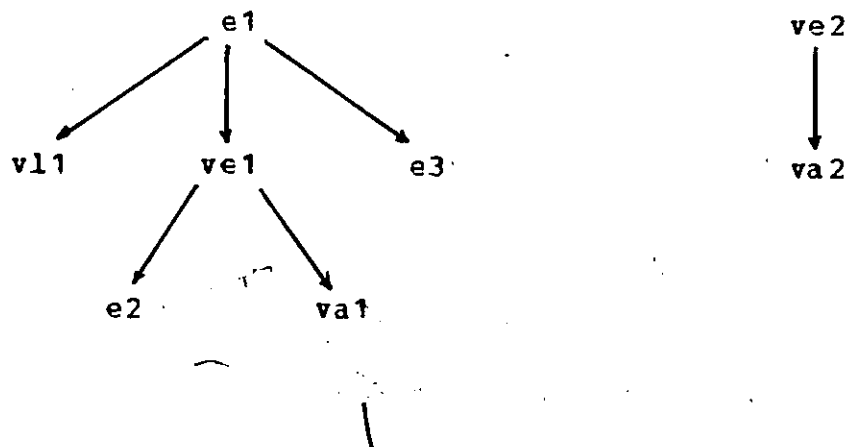


e ::= < étiquette >
 vl ::= < variable liste >
 ve ::= < variable étiquette >
 va ::= < variable arbre >

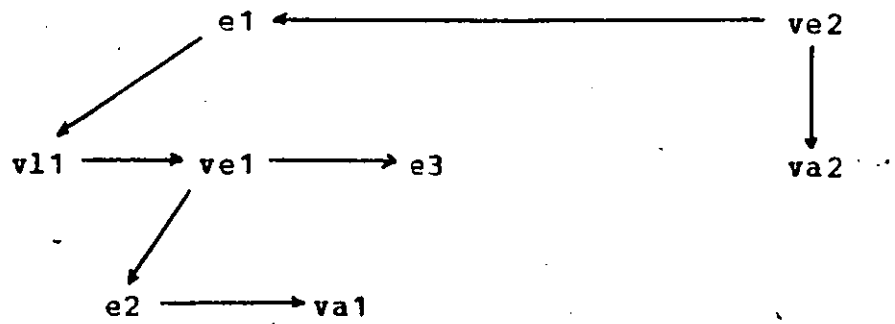
... + e1(vl1, ve1(e2, va1), e3) + ve2(va2) + ...

FIGURE 4.1.2

correspondent au schéma suivant:



Ce schéma nous indique les relations entre les différents éléments des deux arbres. Par exemple, il nous indique que vl1, ve1 et e3 sont les fils de e1 alors que e2 et va1 sont les fils de ve1. Toutefois, pour représenter un arbre en mémoire, on utilise des relations un peu différentes. Au lieu de représenter vl1, ve1 et e3 comme fils de e1, on représente seulement le premier fils, soit vl1, comme fils aîné de e1. Les deux autres fils sont représentés comme frères suivants du fils aîné. On indique donc ve1 comme frère suivant de vl1 et e3 comme frère suivant de ve1. Le schéma suivant indique ces relations, telles qu'utilisées dans la représentation interne des arbres. Dans ce schéma, une flèche verticale ou oblique indique la relation fils aîné alors qu'une flèche horizontale indique la relation frère suivant. Notez que s'il s'agissait d'une section du membre droit d'une règle, la flèche entre e1 et ve2 serait dirigée dans l'autre sens.



Voici maintenant, de façon plus détaillée, comment chaque élément d'un arbre généralisé est représenté en mémoire:

< étiquette >

adresse du fils aîné
adresse du frère suivant
code de l'étiquette

< variable étiquette >

adresse du fils aîné
adresse du frère suivant
code de la variable

< variable arbre > et < variable liste >

adresse du frère suivant
code de la variable

\$\$ < étiquette >

adresse du frère suivant
code de \$\$
code de l'étiquette

\$\$ < variable étiquette >

adresse du frère suivant
code de \$\$
code de la variable

Remarquez que les variables arbrés et les variables listes ainsi que les éléments précédés du désintegrateur \$\$ ne peuvent avoir de fils. De plus, la représentation de l'élément -NUL- n'est pas indiquée puisqu'il est simplement ignoré.

4.1.3. Expression booléenne.

La partie condition d'une règle, représentée par une expression booléenne, exprime des relations entre plusieurs listes. Elle se définit comme suit:

< expression booléenne > ::= < secondaire booléen > |
 < secondaire booléen > -OU- < expression booléen-
 ne >

< secondaire booléen > ::= < primaire booléen > |
 < primaire booléen > -ET- < secondaire booléen >

```

< primaire booléen > ::= -- |
    -NON- < primaire booléen > |
    (. < expression booléenne > .) |
    < liste généralisée > < relation > < liste
    généralisée >

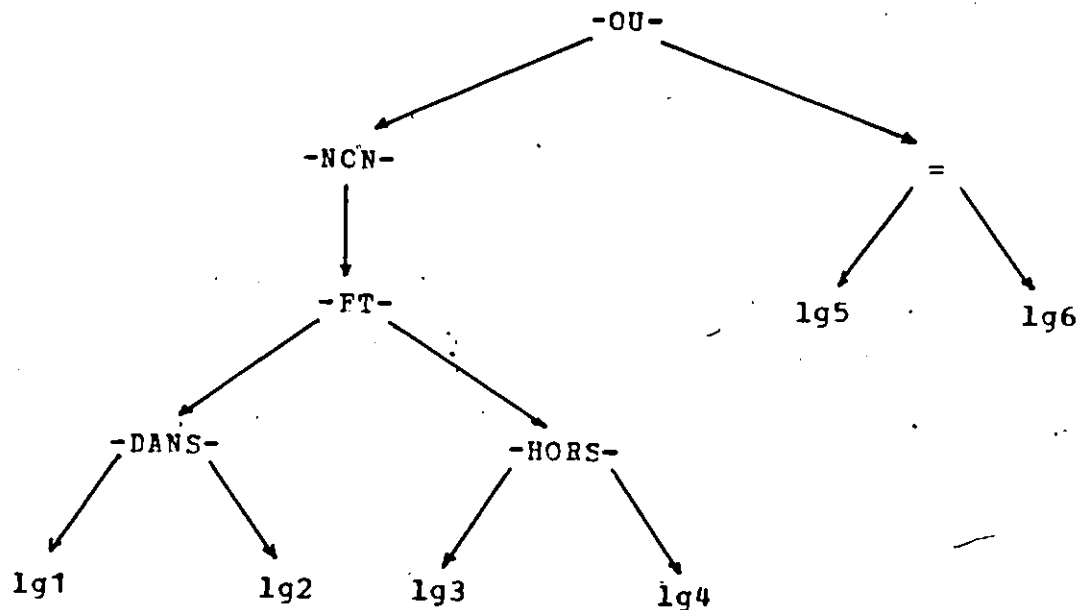
```

```

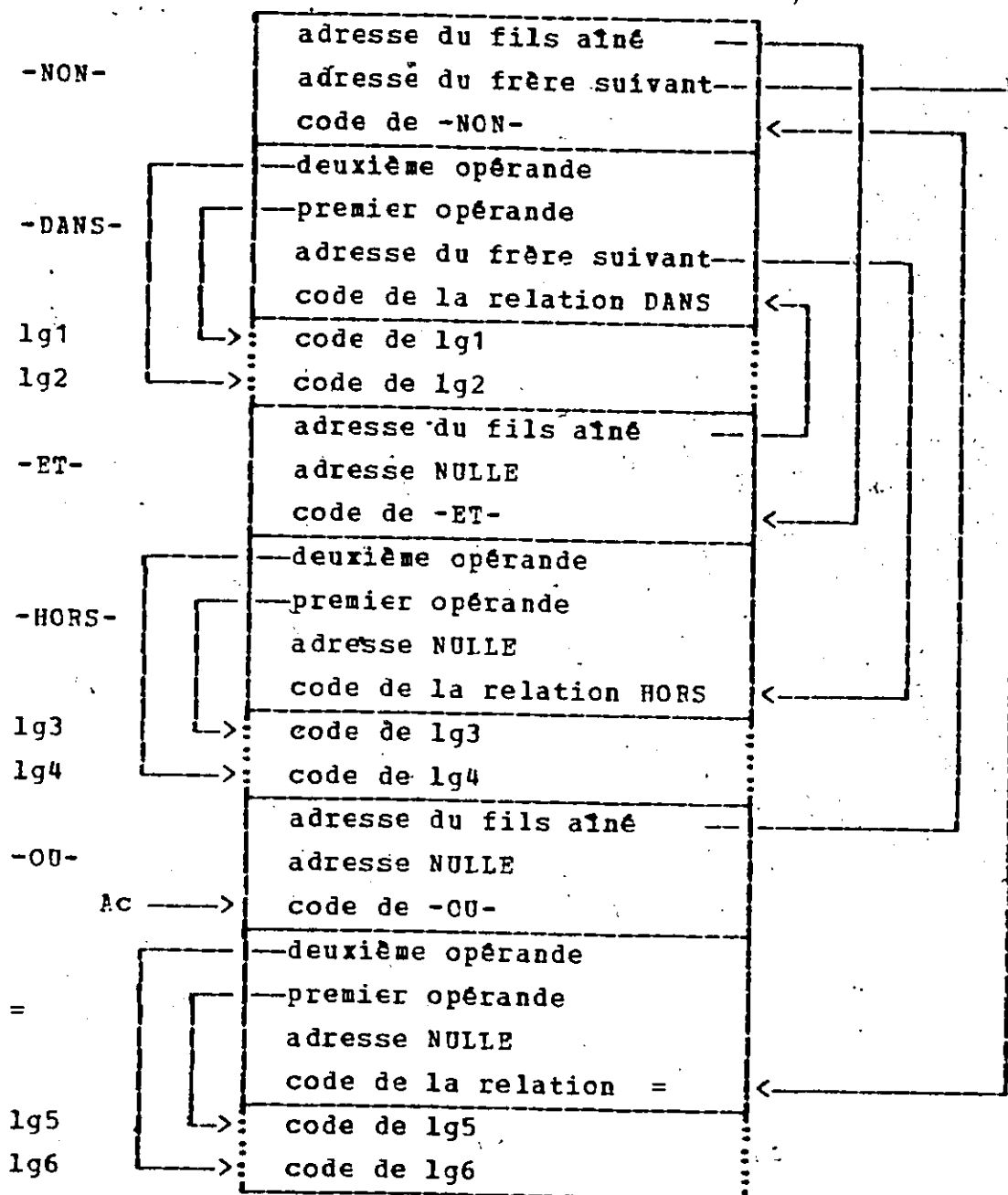
< relation > ::= = | ≠ | -DANS- | -HORS-

```

La FIGURE 4.1.3 représente la condition -NON-(. lg1 -DANS- lg2 -ET- lg3 -HORS- lg4 .) -OU- lg5 = lg6. Cette expression peut être représentée par l'arborescence:



Comme pour les arbres généralisés, la représentation interne d'une expression booléenne utilise les relations de fils aîné et de frère suivant. Cependant, les opérateurs =, ≠, -DANS- et -HORS- font directement référence à leurs deux opérands (soit des listes généralisées, puisqu'il y en a

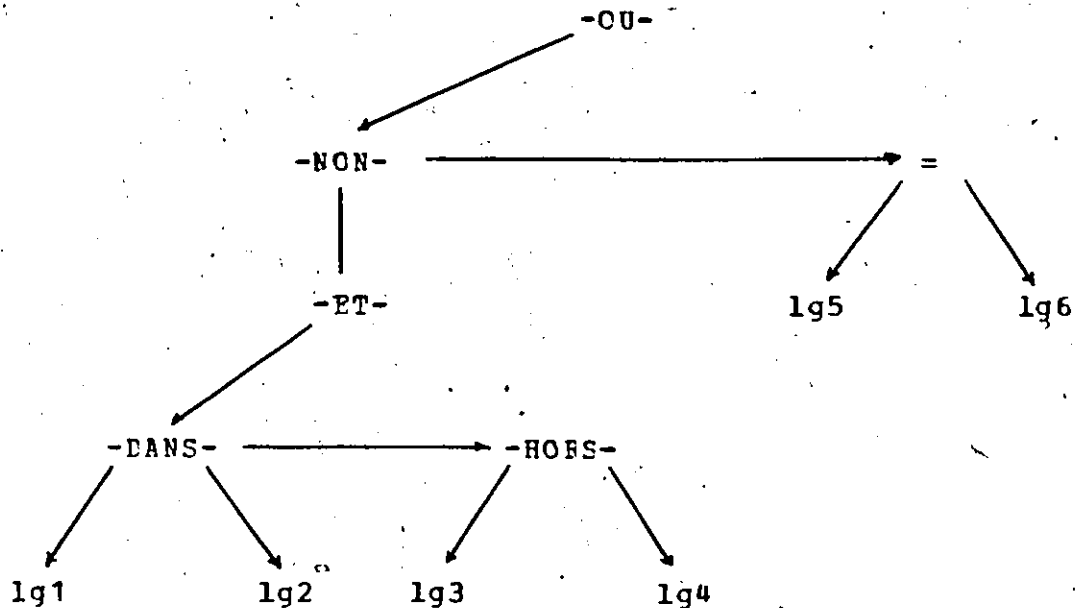


lg ::= < liste généralisée >

/ -NON- (.lg1 -DANS- lg2 -ET- lg3 -HORS- lg4.) -OU- lg5 =
lg6.

FIGURE 4.1.3

toujours seulement deux). Le schéma suivant indique donc les relations entre les éléments telles qu'utilisées dans la représentation interne d'une expression booléenne. Comme auparavant, une flèche verticale ou oblique représente la relation de fils aîné alors qu'une flèche horizontale représente celle de frère suivant.



Voici maintenant, de façon plus détaillée, la représentation interne de chaque élément d'une expression booléenne:

-NON-, -OU-, -ET-

adresse du fils aîné
adresse du frère suivant
code de l'opérateur

< relation >

adresse du 2e opérande
adresse du 1er opérande
adresse du frère suivant
code de la relation

< liste généralisée >

La représentation interne des listes généralisées est identique à celle qui est décrite dans la section sur les arbres généralisés.

4.2 Représentation interne d'un graphe.

Un graphe d'arborescences peut être représenté formellement par

< graphe d'arborescences > ::= < chemin sans
carrefour > | < chemin sans carrefour >
< graphe d'arborescences >

< chemin sans carrefour > ::= < sommet > < suite
d'arbres > < sommet >

< suite d'arbres > ::= < arbre > | < arbre > +
< suite d'arbres >

< sommet > ::= - < entier > -

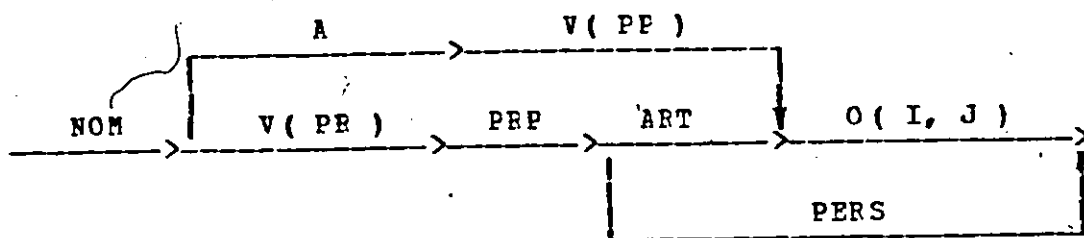
< entier > ::= < chiffre > | < chiffre > < entier >

La représentation des arbres dans un graphe d'arborescences est identique à celle utilisée pour les

règles, sauf qu'il n'y a pas de variable et l'utilisation de la liste vide (-NUL-) et du désintégrateur d'étiquettes (\$\$) n'est pas permise.

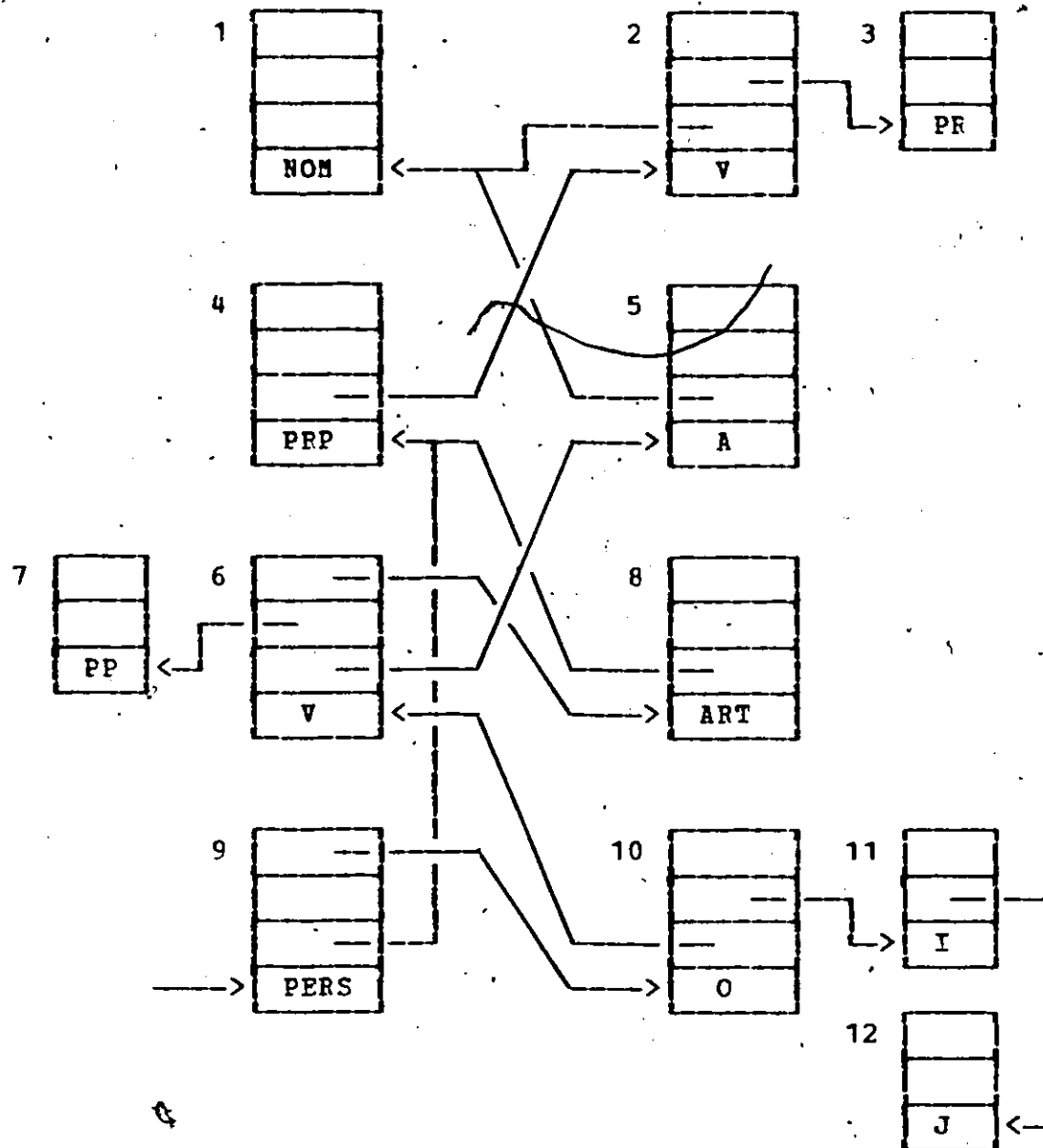
< arbre > ::= < étiquette > | < étiquette > (< liste >)
 < liste > ::= < arbre > | < arbre > , < liste >

La FIGURE 4.2 illustre la représentation interne du graphe donné en exemple dans le chapitre sur les systèmes-q. Ce graphe correspond au schéma



On note tout d'abord que les arbres du graphe sont enchaînés de droite à gauche, comme pour le membre gauche d'une règle, le sommet d'entrée devenant la sortie et le sommet de sortie devenant l'entrée. Les chiffres au coin gauche de chaque bloc indiquent l'ordre physique des blocs en mémoire. Chaque élément du graphe est représenté par un de ces blocs qui contiennent un code étiquette ainsi que les adresses indiquant l'emplacement du fils aîné et du frère suivant de l'élément. De plus, les blocs représentant la racine de chacun des arbres du graphe contiennent une adresse additionnelle qui indique l'emplacement d'un alternant. Cet alternant représente un autre frère suivant de l'élément qui a comme premier frère suivant l'élément dans lequel se trouve l'adresse de l'alternant.

Dans la FIGURE 4.2, l'arbre V(PP) est le frère suivant de O(I, J). Cependant, l'arbre ART est aussi frère



- 1 - NOM - 2 -
- 2 - V (PR) + PRP - 3 -
- 2 - A + V (PP) - 4 -
- 3 - ART - 4 -
- 3 - PERS - 5 -
- 4 - O (I, J) - 5 -

FIGURE 4.2

suisant de $O(I, J)$. Comme un élément ne peut avoir plus d'un frère suisant, l'arbre ART devient l'alternant du premier frère suisant de $O(I, J)$, soit l'arbre $V(PP)$. Si $O(I, J)$ possédait un autre frère suisant, ce dernier deviendrait l'alternant de l'arbre ART.

Voici comment chaque élément d'un graphe est représenté en mémoire:

< étiquette racine >

adresse de l'alternant
adresse du fils aîné
adresse du frère suisant
code de l'étiquette

< étiquette >

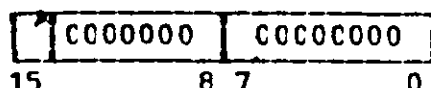
adresse du fils aîné
adresse du frère suisant
code de l'étiquette

4.3 Adresses.

Pour la représentation interne d'une règle, la mémoire est subdivisée en mots de 16 bits. Chaque mot peut donc contenir soit une adresse de 16 bits, soit deux caractères de 8 bits. Toute adresse en mémoire prend la forme

	A	D	R	E	S	S	E	
15					8	7		0

et le dernier bit, qui n'est pas utilisé pour l'adresse elle-même, sert d'indicateur spécial. Dans les systèmes-q, cet indicateur peut servir soit à retenir le trajet déjà parcouru dans un graphe, soit à retenir les arbres du graphe ayant servi à l'application d'au moins une règle. Enfin, une valeur de 0 dans les bits 0 à 14 indique une adresse NULL.



4.4 Étiquettes.

Toute étiquette est représentée par une suite de caractères. Le premier caractère peut être quelconque mais les autres doivent être des chiffres ou des lettres. Il n'y a aucune limite au nombre de caractères qui peuvent former une étiquette, mais plus l'étiquette est longue, plus elle prendra de place en mémoire. Une étiquette se définit comme suit:

< étiquette > ::= < caractère > |

< étiquette > < lettre > |

< étiquette > < chiffre >

< caractère > ::= < lettre > | < chiffre > | < signe significatif > | < signe non significatif >

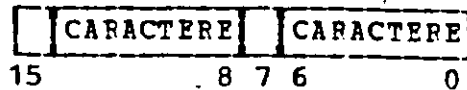
< lettre > ::= A | B | C | D | E | F | G | H | I | J |
K | L | M | N | O | P | Q | R | S | T | U | V | W
| X | Y | Z

< chiffre > ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

< signe significatif > ::= + | - | * | / | (|) | \$ |
= | % | . | ,

< signe non significatif > ::= # | : | ; | < | > | ?

En mémoire, les caractères sont représentés de la façon suivante:



Chaque caractère est représenté par un code ASCII de 7 bits, les bits 15 et 7 servant d'indicateurs. Le bit 15 du premier mot de la chaîne de caractères sert d'indicateur spécial, comme décrit plus haut. De plus, une valeur de 1 dans le bit 7 du mot indique que le mot contient le dernier caractère de la chaîne. Si le nombre de caractères de la chaîne est impair, le dernier octet contient un espace.



4.5 Variables.

Une variable se définit comme suit:

< variable étiquette > ::= < abcdef > * < indice >

< variable arbre > ::= < ijkl > * < indice >

< variable liste > ::= < uvwxyz > * < indice >

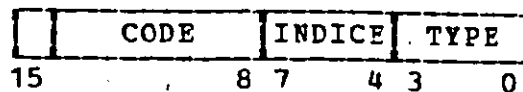
< abcdef > ::= A | B | C | D | E | F

< ijkl > ::= I | J | K | L

< uvwxyz > ::= U | V | W | X | Y | Z

< indice > ::= | < chiffre >

En mémoire, toute variable est représentée par un seul mot.



Le premier octet (bits 15 à 8) contient un code spécial, soit X'20' (espace en ASCII), qui indique qu'il s'agit d'une variable. Comme pour les étiquettes, le bit 15 sert d'indicateur spécial. Enfin, le dernier octet (bits 7 à 0) est divisé en 2 parties, la première représentant l'indice, la seconde indiquant le type de variable.

- code variable.

Le code des variables est X'20'. Lorsque ce code apparaît dans le premier octet (bits 14 à 8), il indique que le mot représente une variable.

- indice.

Cette partie du code variable est constituée de la représentation binaire de l'indice. Notez que si la variable ne possède pas d'indice, on lui attribue l'indice 0.

- type.

Le code type dépend du caractère utilisé dans la définition de la variable.

<u>LETTRE</u>	<u>CODE</u>	<u>TYPE</u>	<u>LETTRE</u>	<u>CODE</u>	<u>TYPE</u>	<u>LETTRE</u>	<u>CODE</u>	<u>TYPE</u>
A	0000		I	0110		U	1010	
B	0001		J	0111		V	1011	
C	0010		K	1000		W	1100	
D	0011		L	1001		X	1101	
E	0100					Y	1110	
F	0101					Z	1111	

Par exemple, le code de J*2 est défini comme suit:

	0100000	0010	0111	
15		8 7	4 3	0

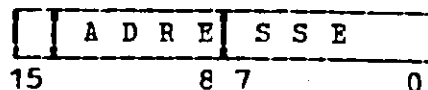
CHAPITRE 5.

DESCRIPTION DU SYSTEME.

Cette section donne une description détaillée de l'organisation du système proposée pour le traitement de structures de graphes et d'arborescences. Le système comporte environ une quarantaine d'instructions machine ainsi qu'une dizaine de modes d'adressage, tous orientés vers le traitement de structures d'arborescences binaires. De plus, il permet l'accès à deux fichiers d'entrée/sortie sur disque ainsi qu'à une imprimante. Enfin, il met une pile interne et quatre registres à la disposition de l'utilisateur.

5.1 Mémoire.

La mémoire du système est composée de mots de 16 bits. Chaque mot possède une adresse de 15 bits de sorte qu'il est possible d'adresser jusqu'à 32,768 mots de mémoire. En mémoire, les adresses prennent la forme suivante,



et le dernier bit, n'étant pas utilisé pour l'adresse elle-même, peut servir d'indicateur quelconque. Enfin, l'adresse composée de 15 zéros représente l'adresse NULLE qui ne réfère à aucun mot en mémoire.

De plus, la mémoire du système est subdivisée en trois zones:

- zone privilégiée.

(0000 à 03FF)

Cette zone est utilisée par le système et n'est pas accessible à l'utilisateur. Elle contient entre autre la pile du système ainsi que les vecteurs d'entrée et de sortie.

- zone de programmation.

(0400 à 3FFF)

Cette zone est accessible à l'utilisateur et sert à retenir les programmes et certaines données. Les modes d'adressage DIRECT, RELATIF, INDIRECT et REGISTRE INDEXE permettent l'accès à cette zone de mémoire.

- zone de travail.

(4000 à 7FFF)

Cette zone est aussi accessible à l'utilisateur et sert à retenir le code des structures de graphes et d'arborescences. Seuls les modes d'adressage VECTEUR INDEXE REGISTRE 0, VECTEUR INDEXE, VECTEUR INDEXE MOINS 1 et VECTEUR INDEXE MOINS 2 permettent l'accès à cette zone de travail.

5.2 Compteur d'instructions.

Le compteur d'instructions est un registre de 16 bits qui indique l'adresse de la prochaine instruction à exécuter. Ce registre est accessible indirectement par l'intermédiaire des instructions de contrôle. Il est à noter que l'adresse dans le compteur d'instructions réfère toujours à un octet de 8 bits, ce qui permet des instructions de longueurs variables. Il n'est donc pas nécessaire que chaque instruction débute sur le premier octet d'un mot. Cependant, la première instruction des sous-programmes ainsi

que toutes les données référencées par une adresse doivent être alignées sur un mot.

5.3 Pile.

Le système comporte une pile interne de 768 mots située dans la zone privilégiée de la mémoire. Cette pile est accessible à l'utilisateur par l'intermédiaire des instructions PSH et POP. Ces dernières permettent d'ajouter et de retirer des éléments de la pile ainsi que de la réinitialiser. De plus, les instructions JSR, JSP et RTS utilisent cette pile pour conserver des données ainsi que des adresses de retour.

L'adresse du sommet de la pile est indiquée par un registre de 16 bits qui n'est pas directement accessible à l'utilisateur. Il est à noter cependant que les instructions PSH et POP permettent à l'utilisateur de créer et de maintenir ses propres piles, indépendamment de la pile du système.

5.4 Registres.

Quatre registres de 16 bits, notés R0 à R3, sont disponibles à l'utilisateur. Ces registres peuvent contenir soit des adresses, soit données. Le contenu de chaque registre est directement accessible à l'aide du mode d'adressage REGISTRE. De plus, le mode d'adressage REGISTRE INDEXE utilise le contenu d'un registre dans le calcul d'une adresse.

Enfin, le registre R0 occupe une fonction un peu spéciale. Son contenu sert dans le calcul d'adresses dans

la zone de travail pour les instructions CDL et CDP ainsi que pour le mode d'adressage VECTEUR INDEXE REGISTRE 0. De plus, il peut être modifié indirectement par l'intermédiaire de l'instruction CDL. De cette façon, le registre R0 peut servir d'indicateur de la position courante du code des graphes et des arborescences dans la zone de travail.

5.5 Indicateurs de présence des variables.

Il existe quatre vecteurs, notés V0 à V3, qui peuvent servir à l'enregistrement de la présence des variables dans les différentes parties d'une règle. Ces vecteurs ont une longueur de 20 octets (ou 160 bits), chaque bit de ces octets étant associé à une des 160 variables disponibles. Les vecteurs V0 et V3 servent à retenir la présence des variables dans la condition d'une règle, alors que le vecteur V1 est associé au membre gauche de la règle et le vecteur V2 à son membre droit. Seules les instructions CDP, APP, CPP et TFP peuvent modifier ou faire référence au contenu de ces vecteurs.

5.6 Code conditionnel.

Le résultat de certaines opérations est reflété par des indicateurs dans le registre du code conditionnel. Ce registre est constitué de deux bits. Le premier de ces bits sert à indiquer la condition zéro (ou l'égalité) alors que le second sert à indiquer la condition négative.

- bit zéro. (Z)

Lorsque le bit zéro du code conditionnel est 1, il indique que les éléments comparés lors de la dernière

opération sont égaux, identiques ou de même type, dans le cas des instructions CMP, CCA, CCC, CLB, CLT, CTR et TPA. Cependant, ce bit sert aussi aux instructions TST, DEC, TTN, TTX et TPP, et peut représenter des conditions variées, suivant l'opération.

- bit négatif. (N)

Lorsque le bit négatif du code conditionnel est 1, il indique que le bit de gauche du résultat de la dernière opération est 1. Ce bit ne peut être mis à 1 que par l'instruction CMP.

Lorsqu'une opération affecte le code conditionnel, les bits zéro et négatif du code sont d'abord remis à 0 afin d'éviter la possibilité que les deux bits soient à 1 simultanément.

Seule l'instruction BRA sert à vérifier l'état du code conditionnel. Le tableau qui suit indique les différentes conditions qui peuvent être vérifiées ainsi que les états du code conditionnel qui satisfont à ces conditions.

<u>CONDITION</u>	<u>ETATS SATISFAISANTS</u>
égal à	Z = 1
plus petit que	N = 1
plus grand que	N = 0 et Z = 0
non égal à	Z = 0
plus petit ou égal à	N = 1 ou Z = 1
plus grand ou égal à	N = 0

5.7 Interruptions.

Le système est pourvu d'un mécanisme de traitement d'interruptions qui permet l'interception de certaines conditions inhabituelles. Lorsqu'une de ces conditions se présente, le bit associé à cette condition dans le registre d'état des interruptions est mis à 1 pour signaler l'interruption. A ce moment, immédiatement après l'exécution de l'instruction ayant causé l'interruption, le système effectue un appel de sous-programme à l'adresse contenue dans le vecteur d'interruption associé à la condition signalée et commence l'exécution du sous-programme de traitement de l'interruption à cette adresse. Ce sous-programme doit être fourni par l'utilisateur et doit se terminer par l'instruction RTS afin d'assurer le retour à l'instruction suivant celle qui a causé l'interruption.

Le registre d'état des interruptions, qui n'est pas accessible à l'utilisateur, est constitué de trois bits et permet le signalement de trois types d'interruptions. Ces types d'interruptions se composent de l'interruption d'adressage, l'interruption de dépassement de capacité et l'interruption de débordement de pile. De plus, une adresse fixe, connue sous le nom de vecteur d'interruption, est associée à chaque type d'interruption et doit contenir l'adresse du sous-programme de traitement de l'interruption.

- interruption d'adressage. (A)

Le bit d'interruption d'adressage est mis à 1 lorsque l'adresse d'un opérande de l'instruction exécutée fait référence à un mot dans la zone privilégiée (0000 à 03FF). Ce bit est aussi mis à 1 lorsque l'adresse d'un opérande fait référence à un mot dans la zone de travail (4000 à 7FFF) et que le mode d'adressage utilisé

est soit le mode DIRECT, RELATIF, INDIRECT ou REGISTRE INDEXE. Enfin, il est à noter que les instructions PSH et POP n'affectent pas ce bit lorsqu'ils font référence à la pile du système, qui se situe dans la zone privilégiée.

- interruption de dépassement de capacité. (O)

Le bit d'interruption de dépassement de capacité est mis à 1 lorsque le calcul de l'adresse d'un opérande de l'instruction exécutée cause un débordement.

- interruption de débordement de pile. (S)

Le bit d'interruption de débordement de pile est mis à 1 lors de l'exécution de l'instruction PSH lorsque la pile du système est remplie et lors de l'exécution de l'instruction POP lorsque la pile du système est vide.

<u>TYPE D'INTERRUPTION</u>	<u>ADRESSE DU VECTEUR D'INTERRUPTION</u>
----------------------------	--

adressage	3FFD
dépassement de capacité.	3FFE
débordement de pile	3FFF

5.8 Entrées / Sorties.

FICHER D'ENTREE:

Le fichier d'entrée se compose d'une série d'images de cartes de 80 caractères chacune. Lors de la lecture, chaque nouvelle carte est affectée à une liste circulaire de 128 caractères. La première carte est donc affectée aux 80 premiers éléments de cette liste. Le pointeur au prochain caractère en entrée indique alors le premier caractère dans la liste. Chaque fois qu'un caractère est lu, ce pointeur est modifié pour indiquer la position du caractère suivant dans la liste. Chaque fois que ce pointeur est augmenté de 80, c'est-à-dire chaque fois que tous les caractères d'une carte ont été lus, les 80 caractères de la carte suivante sont affectés à la liste, à la suite des caractères de la carte précédente. Comme la liste est circulaire, lorsqu'un caractère est affecté au 128ème élément de la liste, le prochain est alors affecté au début de la liste. De cette façon, les 48 derniers caractères de la carte précédente sont toujours conservés dans la liste, permettant un retour en arrière dans le cas de la comparaison d'une chaîne de caractères se poursuivant sur la carte suivante. Enfin, lorsque la fin du fichier est rencontrée, un caractère spécial de code X'03' (fin de transmission) est inséré comme prochain caractère.

VECTEUR DE SORTIE:

Le vecteur de sortie peut contenir jusqu'à 128 caractères et ses dimensions peuvent être modifiées. Tous les caractères à sortir sur l'imprimante passent d'abord par le vecteur de sortie. Ce vecteur contient initialement des

espaces et le pointeur au prochain caractère en sortie indique le début du vecteur. Lorsqu'un caractère doit être imprimé, il est affecté au vecteur de sortie et le pointeur est incrémenté de 1. Les caractères du vecteur ne sont réellement imprimés que lorsque le pointeur au prochain caractère en sortie devient plus grand que la limite supérieure du vecteur, c'est-à-dire lorsque le vecteur est rempli. Dans ce cas, les caractères du vecteur sont imprimés et le vecteur lui-même est réinitialisé à blanc. Il est aussi possible d'imprimer le contenu du vecteur même si la limite supérieure n'a pas été atteinte; dans ce cas le vecteur est également réinitialisé à blanc. Lorsque la limite inférieure du vecteur de sortie est supérieure à 1, elle indique la position sur la ligne à partir de laquelle les caractères doivent être imprimés. C'est donc dire que tous les caractères à gauche de cette position seront des espaces.

FICHER DE SORTIE:

Le fichier de sortie se compose également d'une série d'images de cartes de 80 caractères chacune. Lors de l'écriture sur ce fichier, chaque caractère passe d'abord par un vecteur de 80 éléments et les caractères ne sont affectés au fichier lui-même que lorsque ce vecteur est rempli.

5.9 Modes d'adressage.

Toute instruction faisant référence à un opérande en mémoire possède un champ ADR de 4 bits qui indique le mode de l'ADRESSE qui suit l'instruction. Si l'instruction nécessite deux opérandes, le premier champ ADR réfère toujours à la première ADRESSE alors que le second réfère à la deuxième ADRESSE.

IMMEDIAT:



L'opérande suit l'instruction.

DIRECT:



L'opérande est à l'adresse qui suit l'instruction.

RELATIF:



L'adresse de l'opérande est calculée en ajoutant le déplacement qui suit l'instruction au compteur d'instructions.

INDIRECT:

	0011
DEPLACEMENT	

L'adresse absolue de l'opérande se trouve à l'adresse calculée en ajoutant le déplacement qui suit l'instruction au compteur d'instructions.

REGISTRE:

	01xx
--	------

L'opérande est dans le registre xx.

VECTEUR INDEXE REGISTRE 0:

	1000
--	------

L'opérande est dans la zone de travail, à l'adresse indiquée par le contenu du registre 0. Notez que toute adresse dans la zone de travail consiste en un index et que par conséquent l'adresse zéro indique le premier mot de cette zone.

VECTEUR INDEXE:

	1001
DEPLACEMENT	

L'opérande est dans la zone de travail, à l'adresse indiquée par le contenu du mot à l'adresse calculée en ajoutant le déplacement qui suit l'instruction au compteur d'instructions.

VECTEUR INDEXE MOINS 1:



L'opérande est dans la zone de travail, à l'adresse indiquée par le contenu du mot à l'adresse calculée en ajoutant le déplacement qui suit l'instruction au compteur d'instructions, moins un.

VECTEUR INDEXE MOINS 2:



L'opérande est dans la zone de travail, à l'adresse indiquée par le contenu du mot à l'adresse calculée en ajoutant le déplacement qui suit l'instruction au compteur d'instructions, moins deux.

REGISTRE INDEXE:



L'adresse de l'opérande est calculée en ajoutant au compteur d'instructions le déplacement qui suit l'instruction ainsi que le contenu du registre xx.

5.10 Adressage relatif.

L'adressage relatif est utilisé pour référer à un opérande en mémoire ainsi que pour indiquer une adresse de branchement. Dans les deux cas, l'adresse est représentée par un DEPLACEMENT qui suit l'instruction. Ce DEPLACEMENT représente le nombre d'octets, sous forme de complément à deux, entre l'adresse qui suit le DEPLACEMENT dans l'instruction et l'adresse à laquelle l'instruction fait référence. Cette dernière est donc calculée en ajoutant le DEPLACEMENT au contenu du compteur d'instruction. Lorsque le DEPLACEMENT réfère à une adresse de branchement, le résultat de ce calcul est affecté au compteur d'instructions. D'autre part, lorsque le DEPLACEMENT réfère à un opérande, le résultat donne l'adresse absolue de cet opérande en mémoire et doit donc indiquer le premier octet du mot.

5.11 Jeu d'instructions.

Cette section décrit les opérations effectuées par chacune des instructions du système. Ces instructions sont subdivisées en cinq classes distinctes:

- Instructions de contrôle.

Les instructions de cette classe servent principalement à modifier le compteur d'instructions afin d'effectuer un branchement au cours de l'exécution du programme.

- Instructions de comparaison.

Cette classe comprend toute instruction qui

modifie les bits zéro ou négatif du code conditionnel suivant la relation entre deux éléments.

- Instructions de gestion des pointeurs.

Toutes les instructions de cette classe sont utilisées pour manipuler le contenu d'adresses servant surtout à retenir des index ou des pointeurs.

- Instructions de gestion des variables.

Cette classe comprend les trois instructions servant à la manipulation des indicateurs dans les vecteurs de présence des variables.

- Instructions d'entrées/sorties.

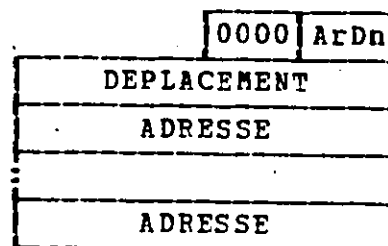
Toutes les instructions de cette classe servent à la lecture ou à l'écriture de caractères sur un des trois unités d'entrées/sorties, soit, le fichier d'entrée, le fichier de sortie ou le vecteur de sortie sur l'imprimante.

Une description complète de chaque instruction ainsi que des exemples de leur utilisation sont inclus. Le langage assembleur utilisé dans les exemples est décrit dans l'Appendice B.

5.11.1 Instructions de contrôle.

JSR	-	Jump to subroutine.
JSP	-	Jump to subfunction.
RTS	-	Return from subroutine.
BRA	-	Branch on condition.
BDX	-	Branch on index.
JMP	-	Unconditionnal jump.
NOP	-	No operation.
HLT	-	Halt.

JSR - Jump to subroutine.



CODE OPERATION : 0000

MODE D'ADRESSAGE : relatif

OPERATION :

Le DEPLACEMENT, qui suit le premier octet de l'instruction, est additionné au "compteur d'instructions pour donner l'adresse du sous-programme. Les deux bits du champ Ar indiquent le nombre d'arguments à passer au sous-programme. Ces arguments, dont les ADRESSES relatives suivent immédiatement le DEPLACEMENT dans l'instruction, sont affectés aux adresses qui précèdent la première instruction du sous-programme. Les deux bits du champ Dn indiquent le nombre de données locales à être conservées. Ces données, dont les ADRESSES relatives suivent celles des arguments dans l'instruction, sont placées sur la pile avec l'adresse de retour.

EXEMPLES D'UTILISATION :

JSR 0,0,MEMDRO

- le nombre zéro est affecté à la pile pour indiquer qu'aucune donnée locale n'a été placée sur la pile.
- l'adresse de l'instruction suivante est affectée au sommet de la pile.
- l'adresse de MEMDRO est affectée au compteur d'instructions.

JSR 1,2,RECU,TEMP,DALTER,COUGRA

- l'argument à l'adresse TEMP est affecté à l'adresse RECU-1.
- les mots aux adresses DALTER et COUGRA sont affectés au sommet de la pile.
- le nombre 2 est affecté à la pile pour indiquer que deux données locales ont été placées sur la pile.
- l'adresse de l'instruction suivante est affectée au sommet de la pile.
- l'adresse de RECU est affectée au compteur d'instruction.

L'instruction de retour qui termine le sous-programme RECU doit être suivie de deux adresses indiquant l'emplacement en mémoire des données locales sur la pile. Il est donc nécessaire que tout appel à ce sous-programme place deux données sur la pile, même si dans certains cas il n'y a aucune donnée à conserver. Dans ces cas, l'instruction qui appelle le sous-programme doit être suivie d'adresses NULLES (ou zéro).

JSP - Jump to subfunction.

0001	ArDn
DEPLACEMENT	
ADRESSE	
ADRESSE	

CODE OPERATION : 0001

MODE D'ADRESSAGE : relatif

OPERATION :

Le DEPLACEMENT, qui suit le premier octet de l'instruction, est additionné au compteur d'instructions pour donner l'adresse du sous-programme. L'ADRESSE de l'argument à retourner au programme appelant, qui apparaît dans l'instruction sous forme relative, immédiatement après le DEPLACEMENT, est transformée en adresse réelle et affectée à l'adresse qui précède la première instruction du sous-programme. Les deux bits du champ Ar indiquent le nombre d'arguments à passer au sous-programme. Ces arguments, dont les ADRESSES relatives suivent immédiatement l'ADRESSE de l'argument à retourner, dans l'instruction, sont affectés aux adresses qui précèdent l'emplacement de l'adresse de l'argument à retourner, avant la première instruction du sous-programme. Les deux bits du champ Dn indiquent le nombre de données locales à être conservées. Ces données, dont les ADRESSES relatives suivent celles des arguments dans l'instruction, sont placées sur la pile avec l'adresse de retour.

EXEMPLE D'UTILISATION :

JSP 1,1,IARBRE,FLISTG,BIDON,NUL

- l'adresse de l'argument FLISTG est affectée à l'adresse IARBRE-1.
- l'argument à l'adresse BIDON est affecté à l'adresse IARBRE-2.
- l'adresse de la donnée locale étant NULLE, le nombre zéro est placé sur la pile.
- le nombre 1 est affecté à la pile pour indiquer qu'une donnée locale a été placée sur la pile.
- l'adresse de l'instruction suivante est affectée au sommet de la pile.
- l'adresse de IARBRE est affectée au compteur d'instructions.

RTS - Return from subroutine.

101111XX
ADRESSE
ADRESSE

CODE OPERATION : 101111

MODE D'ADRESSAGE : relatif

OPERATION :

L'adresse de retour est retirée du début de la pile et placée dans le compteur d'instruction. Le second élément sur la pile indique le nombre de variables locales conservées dans la pile. Ces variables sont alors retirées de la pile et affectées aux ADRESSES relatives qui suivent le premier octet de l'instruction. Notez que les éléments de la pile sont retirés dans l'ordre inverse de celui dans lequel ils y ont été placés.

EXEMPLES D'UTILISATION :

MEMDRO:

:

:

RTS

- l'adresse de retour est retirée du sommet de la pile et affectée au compteur d'instructions.
- l'élément suivant, qui devrait être 0 pour indiquer qu'aucune donnée locale ne fut empilée lors de l'appel au sous-programme MEMDRO, est aussi retiré de la pile.

RECU:

:

:

:

RTS COUGRA, DALTER

- l'adresse de retour est retirée du sommet de la pile et affectée au compteur d'instructions.
- le nombre de données locales est retiré de la pile.
(Ce nombre devrait être de 2 dans ce cas, puisque l'appel au sous-programme RECU place deux données locales sur la pile.)
- les deux données locales sont retirées du sommet de la pile et affectées dans l'ordre aux adresses COUGRA et DALTER.

BRA - Branch on condition.

01001	CDE
DEPLACEMENT	

CODE OPERATION : 01001

OPERATION :

Le DEPLACEMENT qui suit le premier octet de l'instruction est additionné au compteur d'instruction pour donner l'adresse de branchement, si la condition est satisfaite. La condition en effet au moment de l'exécution de l'instruction est déterminée par les indicateurs négatif et zéro du code conditionnel. La condition à vérifier pour effectuer le branchement est indiquée par les trois bits du champ CODE, dont la signification est représentée dans le tableau qui suit.

<u>CODE</u>	<u>CONDITION A VERIFIER</u>
001 (BEQ)	égal à
010 (BLT)	plus petit que
000 (BGT)	plus grand que
101 (BNE)	non égal à
100 (BLE)	plus petit ou égal à
110 (BGE)	plus grand ou égal à

EXEMPLE D'UTILISATION :

BLE SUITE

- l'adresse de SUITE est affectée au compteur d'instructions si l'indicateur zéro ou l'indicateur négatif du code conditionnel est 1.

BDX - Branch on index.

r ADR r	
011000	XX
ADRESSE	
DEPLACEMENT	

CODE OPERATION : 011000

MODES D'ADRESSAGE : direct, relatif, indirect

OPERATION :

Le contenu du mot dont l'ADRESSE suit le premier octet de l'instruction est décrémente de 1 et tant que le résultat n'est pas négatif, le DEPLACEMENT qui apparaît dans l'instruction est additionné au compteur d'instruction pour donner l'adresse de branchement.

EXEMPLE D'UTILISATION :

BOUCLE:

:
:
:
:

BDX I2,BOUCLE

- le mot d'adresse I2 est décrémente de 1.
- si le résultat est positif, l'adresse de BOUCLE est affectée au compteur d'instructions.

JMP - Unconditionnal jump.

011110XX

DEPLACEMENT

CODE OPERATION : 011110

OPERATION :

Le DEPLACEMENT qui suit le premier octet de l'instruction est additionné au compteur d'instruction pour donner l'adresse de branchement.

EXEMPLE D'UTILISATION :

JMP ETIEG

- l'adresse de ETIEG est affectée au compteur d'instructions.

NOP - No operation.

5
111011XX

CODE OPERATION : 111011

OPERATION :

Cette instruction n'exécute aucune
opération.

EXEMPLE D'UTILISATION :

NOP

- passage à l'instruction suivante.

HLT - Halt.

111111XX

CODE OPERATION : 111111

OPERATION :

Cette instruction termine
l'exécution.

EXEMPLE D'UTILISATION :

HLT

- arrêt de l'exécution.

5.11.2 Instructions de comparaison.

CCA	-	Compare characters.
CCC	-	Compare character code.
CMP	-	Compare.
TST	-	Test indicator.
TTN	-	Test inclusion.
TTX	-	Test exclusion.
CLB	-	Compare labels.
CLT	-	Compare lists.
CTR	-	Compare trees.
TPA	-	Test parameter.

CCA - Compare characters.

	01000	CAR
CARACTERE		CARACTERE

CODE OPERATION : 01000

OPERATION :

La chaîne de CARACTERES qui suit le premier octet de l'instruction est comparée aux caractères en entrée. Si les deux chaînes sont identiques, l'indicateur zéro du code conditionnel est mis à 1, et le pointeur au prochain caractère en entrée est incrémenté pour indiquer le caractère qui suit immédiatement la chaîne de caractères venant d'être comparée. De plus, les caractères lus, incluant les espaces, sont affectés au vecteur de sortie. Autrement, si les deux chaînes sont différentes, l'indicateur zéro du code conditionnel est mis à 0, et le pointeur au prochain caractère en entrée n'est pas modifié. Les trois bits du champ CAR indiquent le nombre de caractères dans la chaîne. Si CAR contient 0, la comparaison se fait sur un caractère, mais le pointeur au prochain caractère en entrée n'est pas modifié, même si les caractères comparés sont identiques.

EXEMPLES D'UTILISATION :

CCA A'-NUL-'

BEQ RISNUL

Si les cinq caractères suivants en entrée sont respectivement -, N, U, L et -,

- l'indicateur zéro du code conditionnel est mis à 1.
- l'indicateur du prochain caractère en entrée est modifié pour indiquer le caractère qui suit le

dernier trait d'union (-).

- l'adresse de IISNUL est affectée au compteur d'instructions.

sinon,

- l'indicateur zéro du code conditionnel est remis à 0.
- l'indicateur du prochain caractère en entrée n'est pas modifié.
- l'adresse de l'instruction suivante est affectée au compteur d'instructions.

CCA* A'('

BNE FILS

Si le prochain caractère en entrée est le caractère (,

- l'indicateur zéro du code conditionnel est mis à 1.
- l'adresse de FILS est affectée au compteur d'instructions.

autrement,

- l'indicateur zéro du code conditionnel est remis à 0.
- l'adresse de l'instruction suivante est affectée au compteur d'instructions.

Notez que dans le dernier exemple, l'indicateur du prochain caractère en entrée n'est jamais modifié puisque le symbole * suivant l'instruction CCA indique que le champ CAR de l'instruction doit contenir 0.

CCC - Compare character code.

01101	CDE
-------	-----

CODE OPERATION : 01101

OPERATION :

Si le caractère en entrée est du même type que celui représenté par le champ CODE de l'instruction, l'indicateur zéro du code conditionnel est mis à 1. Autrement, l'indicateur est remis à 0.

<u>CODE</u>	<u>TYPE DE CARACTERE</u>
000	caractère nul
001	chiffre (0 - 9)
010	lettre (A - Z)
011	chiffre ou lettre
100	caractère spécial
101	variable étiquette
110	variable liste
111	variable arbre

Dans le cas des variables, deux caractères sont vérifiés. Notez que cette instruction ne modifie pas le pointeur au prochain caractère en entrée.

EXEMPLES D'UTILISATION :

CCC 6
BNE VARLIS

- l'indicateur zéro du code conditionnel est remis à 0.
- si les deux prochains caractères en entrée représentent une variable liste, l'indicateur zéro du code conditionnel est mis à 1 et dans ce cas l'instruction suivante n'affectera pas de branchement à l'adresse VARLIS.

CCC 1
BEQ CHIFFR

- l'indicateur zéro du code conditionnel est remis à 0.
- si le prochain caractère en entrée est un chiffre entre 0 et 9, l'indicateur zéro du code conditionnel est mis à 1 et l'instruction suivante provoquera un branchement à l'adresse CHIFFR.

CMP - Compare.

100101XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 100101
MODES D'ADRESSAGE : tous
OPERATION :

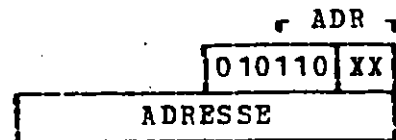
Le contenu du mot à la deuxième ADRESSE apparaissant dans l'instruction est retranché du contenu du mot à la première ADRESSE. Si le résultat est zéro, l'indicateur zéro du code conditionnel est mis à 1, autrement il est remis à 0. Si le résultat est négatif, l'indicateur négatif du code conditionnel est mis à 1, autrement il est remis à 0. Aucun des mots adressés par cette instruction n'est modifié.

EXEMPLE D'UTILISATION :

CMP LGSGCR, LGSGMX

- les indicateurs zéro et négatif du code conditionnel sont remis à 0.
- le contenu du mot d'adresse LGSGMX est retranché du mot d'adresse LGSGCR.
- si le résultat de la soustraction est zéro, l'indicateur zéro du code conditionnel est mis à 1.
- si le résultat de la soustraction est négatif, l'indicateur négatif du code conditionnel est mis à 1.

TST - Test indicator.



CODE OPERATION : 010110

MODE D'ADRESSAGE : vecteur indexé

OPERATION :

Vérifie si le bit de gauche du mot dont l'ADRESSE suit le premier octet de l'instruction est 0 ou 1. Si ce bit est 1, l'indicateur zéro du code conditionnel est mis à 0. Autrement, l'indicateur est mis à 1.

EXEMPLE D'UTILISATION :

TST V(DALTER-2)

BEQ IISARB

- l'indicateur zéro du code conditionnel est remis à 0.
- si le bit de gauche du mot d'adresse V(DALTER-2) est 0, l'indicateur zéro du code conditionnel est mis à 1 et l'adresse de IISARB sera affectée au compteur d'instructions.

TTN - Test inclusion.

100110XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 100110

MODES D'ADRESSAGE : tous

OPERATION :

Cette instruction vérifie la relation -DANS- entre deux listes. Le mot à la première ADRESSE apparaissant dans l'instruction est utilisé comme pointeur dans la zone de travail du début de la première liste alors que le mot à la deuxième ADRESSE est utilisé comme pointeur dans la zone de travail du début de la seconde liste. Si la relation -DANS- entre les deux listes est satisfaite, l'indicateur zéro du code conditionnel est mis à 1, autrement il est remis à 0.

EXEMPLE D'UTILISATION :

TTN V(DLIST1-1),V(DLIST2-1)

BEQ DNSLIS

- L'indicateur zéro du code conditionnel est remis à 0.
- si la relation -DANS- entre les deux listes dont les adresses dans la zone de travail sont indiquées par le contenu des mots aux adresses V(DLIST1-1) et V(DLIST2-1) est satisfaite, l'indicateur zéro du code conditionnel est mis à 1 et l'instruction suivante provoquera un branchement à l'adresse DNSLIS.

TTX - Test exclusion.

100111XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 100111

MODES D'ADRESSAGE : tous

OPERATION :

Cette instruction vérifie la relation -HORS- entre deux listes. Le mot à la première ADRESSE apparaissant dans l'instruction est utilisé comme pointeur dans la zone de travail du début de la première liste alors que le mot à la deuxième ADRESSE est utilisé comme pointeur dans la zone de travail du début de la seconde liste. Si la relation -HORS- entre les deux listes est satisfaite, l'indicateur zéro du code conditionnel est mis à 1, autrement il est remis à 0.

EXEMPLE D'UTILISATION :

TTX V(DLIST1-1),V(DLIST2-1)
BNE LISLIS

- L'indicateur zéro du code conditionnel est remis à 0.
- si la relation -HORS- entre les deux listes dont les adresses dans la zone de travail sont indiquées par le contenu des mots aux adresses V(DLIST1-1) et V(DLIST2-1) est satisfaite, l'indicateur zéro du code conditionnel est mis à 1 et dans ce cas, l'instruction suivante ne provoquera pas de branchement à l'adresse LISLIS.

CLB -- Compare labels.

101000XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 101000

MODES D'ADRESSAGE : tous

OPERATION :

Cette instruction vérifie l'égalité entre deux étiquettes. Le mot à la première ADRESSE apparaissant dans l'instruction est utilisé comme pointeur dans la zone de travail du début de la première étiquette alors que le mot à la deuxième ADRESSE est utilisé comme pointeur dans la zone de travail du début de la seconde étiquette. Si les deux étiquettes sont identiques, l'indicateur zéro du code conditionnel est mis à 1, autrement il est remis à 0.

EXEMPLE D'UTILISATION :

CLB COUGRA, COUREG

BIF ETIEG

- l'indicateur zéro du code conditionnel est remis à 0.
- si les deux étiquettes dont les adresses dans la zone de travail sont indiquées par le contenu des mots aux adresses COUGRA et COUREG sont identiques, l'indicateur zéro du code conditionnel est mis à 1 et l'instruction suivante provoquera un branchement à l'adresse ETIEG.

CLT - Compare lists.

101001XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 101001
MODES D'ADRESSAGE : tous
OPERATION :

Cette instruction vérifie l'égalité entre deux listes. Le mot à la première ADRESSE apparaissant dans l'instruction est utilisé comme pointeur dans la zone de travail du début de la première liste alors que le mot à la deuxième ADRESSE est utilisé comme pointeur dans la zone de travail du début de la seconde liste. Si les deux listes sont identiques, l'indicateur zéro du code conditionnel est mis à 1, autrement il est remis à 0.

EXEMPLE D'UTILISATION :

CLT V(DLIST-1),V(GLIST-1)
BEQ LISTEG

- l'indicateur zéro du code conditionnel est remis à 0.
- si les deux listes dont les adresses dans la zone de travail sont indiquées par le contenu des mots aux adresses V(DLIST-1) et V(GLIST-1) sont identiques, l'indicateur zéro du code conditionnel est mis à 1 et l'instruction suivante provoquera un branchement à l'adresse LISTEG.

CTR - Compare trees.

101010XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 101010

MODES D'ADRESSAGE : tous

OPERATION :

Cette instruction vérifie l'égalité entre deux arbres. Le mot à la première ADRESSE apparaissant dans l'instruction est utilisé comme pointeur dans la zone de travail du début du premier arbre alors que le mot à la deuxième ADRESSE est utilisé comme pointeur dans la zone de travail du début du second arbre. Si les deux arbres sont identiques, l'indicateur zéro du code conditionnel est mis à 1, autrement il est remis à 0.

EXEMPLE D'UTILISATION :

CTR COUGRA, DVPLIS
BEQ ARBEG

- l'indicateur zéro du code conditionnel est remis à 0.
- si les deux arbres dont les adresses dans la zone de travail sont indiquées par le contenu des mots aux adresses COUGRA et DVPLIS sont identiques, l'indicateur zéro du code conditionnel est mis à 1 et l'instruction suivante provoquera un branchement à l'adresse ARBEG.

TPA - Test parameter.

101101	CD	XXXX	ADR
ADRESSE			

CODE OPERATION : 101101

MODES D'ADRESSAGE : tous

OPERATION :

Si la variable codée à l'ADRESSE qui suit le premier octet de l'instruction est du même type que celui représenté par le champ Code de l'instruction, l'indicateur zéro du code conditionnel est mis à 1. Autrement, l'indicateur est remis à 0.

<u>CODE</u>	<u>TYPE DE CARACTERE</u>
00	aucun (étiquette)
01	variable étiquette
10	variable arbre
11	variable liste

EXEMPLES D'UTILISATION :

TPA 2,V(COUREG)

- l'indicateur zéro du code conditionnel est remis à 0.
- si le code à l'adresse V(COUREG) représente une variable arbre, le bit zéro du code conditionnel est mis à 1.

TPA 0,V(COUGRA)

- l'indicateur zéro du code conditionnel est remis à 0.
- si le code à l'adresse V(COUGRA) représente une étiquette, le bit zéro du code conditionnel est mis à 1.

5.11.3 Instructions de gestion des pointeurs.

CLR - Clear.

MOV - Move.

ADD - Addition.

SUB - Subtraction.

PSH - Push stack.

POP - Pop stack.

IPT - Increment pointer.

SPT - Set pointer.

SET - Set indicator.

CLI - Clear indicator.

CRI - Classify rule.

CLR - Clear.

0010	ADR
ADRESSE	

CODE OPERATION : 0010

MODES D'ADRESSAGE : tous

OPERATION :

Remet à zéro tous les bits du mot à l'ADRESSE qui suit le premier octet de l'instruction.

EXEMPLES D'UTILISATION :

CLR IARBRE

- le mot d'adresse IARERE est mis à 0.

CLR V(PTCALC)

- le pointeur à l'adresse V(PTCALC) est mis à 0 pour indiquer une adresse NULLE.

MOV - Move.

011101XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 011101
MODES D'ADRESSAGE : tous
OPERATION :

Le contenu du mot à la deuxième
ADRESSE dans l'instruction est affecté au mot à la première
ADRESSE dans l'instruction.

EXEMPLES D'UTILISATION :

MOV PCONDI,3

- le nombre 3 est affecté à l'adresse PCONDI.

MOV DLISTG,DLISTD

- le contenu du mot d'adresse DLISTD est affecté à
l'adresse DLISTG.

84.

ADD - Addition.

100001XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 100001

MODES D'ADRESSAGE : tous

OPERATION :

Le contenu du mot à la deuxième ADRESSE apparaissant dans l'instruction est additionné au contenu du mot à la première ADRESSE et le résultat est affecté à cette première ADRESSE.

EXEMPLE D'UTILISATION :

ADD IGRAPH,200

- le nombre 200 est additionné au contenu du mot d'adresse IGRAPH et le résultat est affecté à cette adresse.

SUB - Subtraction.

100010XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 100010
MODES D'ADRESSAGE : tous
OPERATION :

Le contenu du mot à la deuxième ADRESSE apparaissant dans l'instruction est retranché du contenu du mot à la première ADRESSE et le résultat est affecté à cette première ADRESSE.

EXEMPLE D'UTILISATION :

SUB I2,R3

- le contenu du registre R3 est retranché du contenu du mot d'adresse I2 et le résultat est affecté à cette adresse.

PSH - Push stack.

010101XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 010101

MODES D'ADRESSAGE : immédiat,direct
relatif,indirect

OPERATION :

Le contenu du mot à la première ADRESSE apparaissant dans l'instruction est incrémenté de 1 et l'adresse ainsi calculée est utilisée comme sommet de pile sur laquelle est affecté le mot à la deuxième ADRESSE dans l'instruction. Lorsque le premier mode d'adressage est le mode IMMEDIAT (ADR = 0000), l'ADRESSE correspondante n'apparaît pas et l'instruction réfère au sommet de la pile du système. Dans ce cas, le mot à l'ADRESSE suivant l'instruction est affecté à la pile du système.

EXEMPLES D'UTILISATION :

PSH SOMPIL COUGRA

- le mot d'adresse SOMPIL est incrémenté de 1.
- le contenu du mot d'adresse COUGRA est affecté à l'adresse indiquée par le contenu du mot d'adresse SOMPIL.

PSH #DALTER

- le pointeur de sommet de la pile du système est incrémenté de 1.
- l'élément #DALTER, c'est-à-dire l'adresse de DALTER, est affecté au sommet de la pile du système.

POP - Pop stack.

110111XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 110111

MODES D'ADRESSAGE : immédiat,direct
relatif,indirect

OPERATION :

Le mot sur la pile dont le pointeur de sommet est à la première ADRESSE apparaissant dans l'instruction est transféré à la deuxième ADRESSE dans l'instruction et le contenu du mot à la première ADRESSE est ensuite décrémenté de 1. Lorsque le premier mode d'adressage est le mode IMMEDIAT (ADR = 0000), l'ADRESSE correspondante n'apparaît pas et l'instruction réfère au sommet de la pile du système. Dans ce cas, le mot au sommet de la pile du système est transféré à l'ADRESSE suivant l'instruction. Enfin, lorsque les deux modes d'adressage de l'instruction sont le mode IMMEDIAT, aucune ADRESSE ne suit l'instruction et le pointeur de sommet de la pile du système est réinitialisé au début de la pile, ce qui a comme effet d'enlever tous les éléments de cette pile.

EXEMPLES D'UTILISATION:

POP SOMPIL,RSOUSA

- le mot à l'adresse indiquée par le contenu de SOMPIL est affecté à l'adresse RSOUSA.
- le mot d'adresse SOMPIL est décrémenté de 1.

POP R2

- l'élément au sommet de la pile du système est transféré au registre R2.
- le pointeur de sommet de la pile du système est décrémenté de 1.

POP

- le pointeur de sommet de la pile du système est réinitialisé au début de la pile.

IPT - Increment pointer.

0011	ADR	ENTIER
ADRESSE		

CODE OPERATION : 0011

MODES D'ADRESSAGE : TOUS

OPERATION :

Le bit de poids fort de l'ENTIER qui apparaît dans l'instruction est propagé vers la gauche pour former un nombre en complément à deux de 16 bits. Ce nombre est ensuite additionné au contenu du mot dont l'ADRESSE suit l'ENTIER dans l'instruction et le résultat est stocké à cette même ADRESSE.

EXEMPLES D'UTILISATION :

IPT 2, LGCD SG

- le mot d'adresse LGCD SG est incrémenté de 2.

IPT -1, SOMPIL

- le mot d'adresse SOMPIL est décrémenté de 1.

SPT - Set pointer.

1000 11XX		
ENTIER	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 100011

MODES D'ADRESSAGE : tous

OPERATION :

Le bit de poids fort de l'ENTIER qui apparaît dans l'instruction est propagé vers la gauche pour former un nombre en complément à deux de 16 bits. Ce nombre est ensuite additionné au contenu du mot à la deuxième ADRESSE apparaissant dans l'instruction et le résultat est affecté à la première ADRESSE apparaissant dans l'instruction.

EXEMPLE D'UTILISATION :

SPT -2,PTCALC,DRETIQ

- le nombre 2 est retranché du contenu du mot d'adresse DRETIQ et le résultat est affecté à l'adresse PTCALC.

SET - Set indicator.

r. ADR	
110100	XX
ADRESSE	

CODE OPERATION : 110100

MODE D'ADRESSAGE : vecteur indexé

OPERATION :

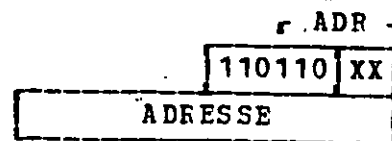
Le bit de gauche du mot à l'ADRESSE
qui suit le premier octet de l'instruction est mis à 1.

• EXEMPLE D'UTILISATION :

SET V (COUGRA-2) :

- le bit de gauche du mot d'adresse V (COUGRA-2) est mis
à 1.

CLI - Clear indicator.



CODE OPERATION : 110110

MODE D'ADRESSAGE : vecteur indexé

OPERATION :

Le bit de gauche du mot à l'ADRESSE
qui suit le premier octet de l'instruction est mis à 0.

EXEMPLE D'UTILISATION :

CLI V(PTSORT)

- le bit de gauche du mot d'adresse V(PTSORT) est mis à
0.

CRL - Classify rule.

010111XX	ADR	ADR
ADRESSE		
ADRESSE		

CODE OPERATION : 010111

MODES D'ADRESSAGE : tous

OPERATION :

Le code de 16 bits à la deuxième ADRESSE apparaissant dans l'instruction est décomposé pour former un pointeur de 8 bits qui pourra servir d'adresse dans une table. Ce pointeur de 8 bits est affecté à la première ADRESSE et les 8 bits de gauche sont mis à zéro. Si le code représente un variable, le premier octet du mot à la première ADRESSE suivant l'instruction est mis à 1 alors que le deuxième octet de ce mot est mis à 0.

EXEMPLE D'UTILISATION :

CRL CLERGC,V(DLISTG)

Si le code à l'adresse V(DLISTG) représente une étiquette quelconque,

- le premier octet du mot d'adresse CLERGC est mis à 0.
- les cinq bits de droite du code du deuxième caractère à l'adresse V(DLISTG) et les trois bits de droite du code du premier caractère à cette même adresse sont combinés pour former un pointeur de 8 bits, les cinq bits étant placés à gauche et les trois autres à droite, et ce pointeur est affecté au deuxième octet du mot d'adresse CLERGC.

autrement, s'il représente une variable quelconque,

- le premier octet du mot d'adresse CLERGC est mis à 1.
- le deuxième octet du mot d'adresse CLERGC est mis à 0.

5.11.4 Instructions de gestion des variables.

APP - Add present parameters.
CPP - Clear present parameters.
TPP - Test present parameters.

APP - Add present parameter.

r ADR r	
101100	XX
ADRESSE	

CODE OPERATION : 101100

MODES D'ADRESSAGE : immédiat,direct
relatif,indirect

OPERATION :

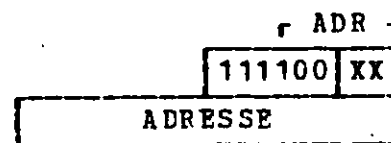
Les indicateurs des variables apparaissant dans la condition de la règle précédente sont transférés au vecteur qui enregistre la présence des variables dans la condition présente. Le code (0 ou 3) à l'ADRESSE qui suit le premier octet de l'instruction doit indiquer le vecteur d'utilisation des variables dans la condition présente.

EXEMPLE D'UTILISATION :

APP PARTIE

- le OU logique entre les indicateurs du vecteur qui retient la présence des variables dans la condition de la règle précédente et ceux du vecteur condition présente est effectué et le résultat est affecté au vecteur condition présente (V0 ou V3) indiqué par le code à l'adresse PARTIE.

CPP - Clear present parameter.



CODE OPERATION : 111100

MODES D'ADRESSAGE : immédiat, direct
relatif, indirect

OPERATION :

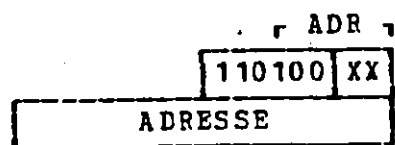
Le vecteur qui enregistre la présence des variables, soit dans le membre gauche, le membre droit ou la condition, est réinitialisé à zéro, indiquant l'absence de variables dans ce vecteur. Un code de 0 ou 3 à l'ADRESSE qui suit le premier octet de l'instruction indique qu'un des vecteurs condition doit être réinitialisé, alors qu'un code de 1 indique le vecteur membre gauche et un code de 2 le vecteur membre droit.

EXEMPLE D'UTILISATION :

CPP PARTIE

- les indicateurs du vecteur représenté par le code à l'adresse PARTIE sont remis à 0.

TPP - Test present parameter.



CODE OPERATION : 110100

MODES D'ADRESSAGE : immédiat,direct
relatif,indirect

OPERATION :

Si toutes les variables dont la présence est enregistrée dans le vecteur de présence des variables indiqué par le code à l'ADRESSE qui suit le premier octet de l'instruction sont aussi présentes dans le vecteur qui enregistre la présence des variables dans le membre gauche, l'indicateur zéro du code conditionnel est mis à 1. Autrement, si une de ces variables n'apparaît pas dans le vecteur membre gauche, cet indicateur est remis à 0. Un code de 0 ou 3 à l'ADRESSE qui suit le premier octet de l'instruction indique qu'un des vecteurs condition doit être utilisé alors qu'un code de 2 indique le vecteur membre droit.

EXEMPLE D'UTILISATION :

TPP SECTN

- l'indicateur zéro du code conditionnel est remis à 0.
- si, pour tout bit de 1 dans le vecteur indiqué par le code à l'adresse SECTN, le bit correspondant du vecteur membre gauche est aussi 1, l'indicateur zéro du code conditionnel est mis à 1.

5.11.5 Instructions d'entrées/sorties.

RDC	-	Read character.
WRC	-	Write character.
PRT	-	Print characters.
SPM	-	Set printer mode.
SIO	-	Swap input output.
DOT	-	Decode and output tree.
BIN	-	Code binary.
DEC	-	Code decimal.
CDL	-	Code label.
CDP	-	Code parameter.

RDC - Read character.

110000	*	+
--------	---	---

CODE OPERATION : 110000

OPERATION :

Le pointeur au prochain caractère en entrée est modifié pour indiquer le prochain caractère autre qu'un espace. Si la fin du fichier est rencontrée, un caractère spécial, de code X'03', est inséré comme prochain caractère. De plus, les caractères lus sont insérés dans le vecteur de sortie si l'un des champs * ou + contient 1. Si le champ + contient 1, les espaces rencontrés sont aussi insérés dans le vecteur de sortie.

EXEMPLES D'UTILISATION :

RDC

- l'indicateur du prochain caractère en entrée est modifié pour indiquer le prochain caractère non blanc en entrée.

RDC*

- l'indicateur du prochain caractère en entrée est modifié pour indiquer le prochain caractère non blanc en entrée.
- le prochain caractère en entrée est affecté au vecteur de sortie sur l'imprimante.

RDC+

- l'indicateur du prochain caractère en entrée est modifié pour indiquer le prochain caractère non blanc en entrée.
- les espaces rencontrés sont affectés au vecteur de sortie sur l'imprimante.
- le prochain caractère en entrée est affecté au vecteur de sortie sur l'imprimante.

WRC - Write character.

110001XX	CARACTERE
----------	-----------

CODE OPERATION : 110001

OPERATION :

Le caractère qui suit le premier octet de l'instruction est affecté au fichier de sortie. Si ce caractère est le caractère ETX (Fin de Transmission), de code 03, le fichier de sortie est fermé.

EXEMPLES D'UTILISATION :

WRC A'/'

- le code ASCII du caractère '/' est affecté au fichier de sortie.

WRC X'03'

- le fichier de sortie est fermé.

PRT - Print characters.

r ADR r	
100100	XX
ADRESSE	

CODE OPERATION : 100100

MODES D'ADRESSAGE : immédiat,direct
relatif,indirect

OPERATION :

La chaîne de caractères à l'ADRESSE qui suit le premier octet de l'instruction est affectée au vecteur de sortie et le pointeur au prochain caractère à la sortie est modifié pour indiquer la position suivant le dernier caractère dans le vecteur. Si l'indicateur dépasse la limite des dimensions du vecteur, ce dernier est imprimé et réinitialisé à blanc, et la suite de la chaîne de caractères est affectée au début du vecteur réinitialisé. Le premier élément de la chaîne de caractères doit toujours indiquer le nombre de caractères dans la chaîne. De plus, si ADR indique le mode d'adressage immédiat, le caractère affecté au vecteur de sortie est le prochain caractère en entrée et dans ce cas l'instruction n'occupe qu'un octet.

EXEMPLES D'UTILISATION :

PRT. CHAINE

:

:

CHAINE: DC 28,A' ERREUR. SOMMET TROP GRAND. '

- la chaîne de 28 caractères à l'adresse CHAINE est affectée au vecteur de sortie sur l'imprimante.

PRT

- le prochain caractère en entrée est affecté au vecteur de sortie sur l'imprimante.

SPM - Set print mode.

	110101	CD
DEBLIGN	FINLIGN	

CODE OPERATION : 110101

OPERATION :

Cette instruction sert à indiquer le mode du vecteur de sortie, suivant le contenu du champ CoDe.

Si le champ CoDe contient 00, les deux octets suivant le premier octet de l'instruction déterminent les dimensions du vecteur de sortie. Le premier des deux octets indique la position du premier caractère à être imprimé sur une nouvelle ligne alors que le second indique la position du dernier caractère à être imprimé avant de passer à la ligne suivante. La limite inférieure est de 1 alors que la limite supérieure est de 128. Donc, si le contenu du champ DEBLIGN est zéro ou supérieur au contenu du champ FINLIGN, le premier caractère sera affecté au début du vecteur de sortie et la fin sera indiquée par le contenu du champ FINLIGN.

Si le champ CoDe contient 11 (*), l'opération est identique à celle effectuée par le code 00, sauf que le contenu du vecteur de sortie est imprimé avant que ses dimensions soient modifiées.

Si le champ CoDe contient 01 (/), le contenu du vecteur de sortie est imprimé immédiatement et réinitialisé à blanc.

Enfin, si le champ CoDe contient 10 (*), le contenu du vecteur de sortie sera imprimé sur la même ligne que le précédent.

Notez que dans le cas des codes 01 et 10, les deux octets DEBLIGN et FINLIGN n'apparaissent pas dans l'instruction.

EXEMPLES D'UTILISATION :

SPM* 1,120

- le contenu du vecteur de sortie est imprimé et le vecteur lui-même est réinitialisé à blanc.
- la limite inférieure du vecteur de sortie est mise à 1 et sa limite supérieure est mise à 120.

SPM/

- le contenu du vecteur de sortie est imprimé et le vecteur lui-même est réinitialisé à blanc.

SPM+

- lorsque le contenu du vecteur de sortie sera imprimé, les caractères apparaîtront sur la même ligne que le précédent.

SIO - Swap input/output.

111001XX

CODE OPERATION : 111001

OPERATION :

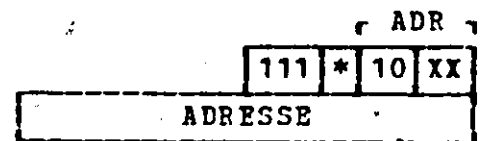
Les fichiers d'entrée et de sortie sont interchangeés, le fichier d'entrée devenant le fichier de sortie et vice-versa, et les pointeurs sont réinitialisés pour indiquer le début de chaque fichier.

EXEMPLE D'UTILISATION :

SIO

- les fichiers d'entrée et de sortie sont fermés.
- le fichier de sortie devient le fichier d'entrée et vice-versa.
- les pointeurs sont réinitialisés au début de chaque fichier.

DOT - Decode and output tree.



CODE OPERATION : 111x10

MODE D'ADRESSAGE : vecteur indexé

OPERATION :

L'arbre, dont le code interne se situe dans la zone de travail, débutant à l'ADRESSE suivant le premier octet de l'instruction, est décodé et affecté à la sortie sous forme de caractères. Si le bit du champ * est 0, l'arbre décodé est affecté à l'imprimante. Si ce bit est 1, l'arbre est affecté au fichier de sortie.

EXEMPLE D'UTILISATION :

DOT* V(COURAN)

- l'arbre dont la racine est à l'adresse V(COURAN) est décodé et affecté à l'imprimante sous forme d'une suite d'étiquettes, de parenthèses et de virgules.

BIN - Code binary.



CODE OPERATION : 100000

MODES D'ADRESSAGE : direct, relatif, indirect

OPERATION :

La suite de chiffres en entrée est codée sous forme binaire et le résultat est affecté à l'ADRESSE qui suit le premier octet de l'instruction. Ce codage se poursuit tant que le caractère en entrée est un chiffre. Le pointeur au prochain caractère en entrée est alors modifié pour indiquer ce caractère. De plus, les caractères lus, sauf les espaces, sont affectés au vecteur de sortie. Enfin, si le nombre représenté par la suite de chiffres en entrée est supérieur à 9999, l'indicateur zéro du code conditionnel est mis à 1 et le contenu du mot à l'adresse qui suit l'instruction n'est pas modifié.

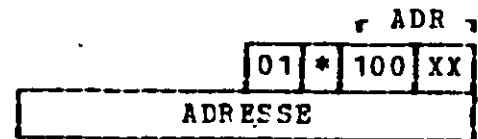
EXEMPLE D'UTILISATION :

BIN NOSOME

Soit la chaîne 712 - en entrée, le prochain caractère en entrée étant le 7,

- le nombre X'02C8' est affecté à l'adresse NOSOME.
- l'indicateur du prochain caractère en entrée est incrémenté pour indiquer le caractère suivant le chiffre 2, soit -.
- les caractères 1, 2 ainsi que le prochain caractère en entrée (soit '-') sont affectés au vecteur de sortie sur l'imprimante.

DEC - Code decimal.



CODE OPERATION : 01x100

MODES D'ADRESSAGE : tous

OPERATION :

Le nombre binaire à l'ADRESSE apparaissant dans l'instruction est transformé en nombre décimal et chaque chiffre (entre 0 et 9) est codé sous forme de caractère. Ces caractères sont affectés au vecteur de sortie si le bit du champ * est 0. Autrement, si le bit du champ * est 1, ces caractères sont affectés au fichier de sortie.

EXEMPLE D'UTILISATION :

DEC* NSOMMT

- le nombre binaire à l'adresse NSOMMT est transformé en nombre décimal et le code ASCII de chaque chiffre généré est affecté au fichier de sortie.

CDL - Code label.

101011XX

CODE OPERATION : 101011

OPERATION :

L'étiquette en entrée est codée dans sa forme interne et déposé dans la zone de travail, à l'adresse indiquée par le contenu du registre 0. Si le codage requiert plus de deux octets (un mot), le contenu du registre 0 est incrémenté automatiquement pour indiquer le dernier mot ayant servi à retenir le code. De plus, le pointeur au prochain caractère en entrée est modifié pour indiquer le prochain caractère suivant l'étiquette et tous les caractères lus, incluant les espaces, sont affectés au vecteur de sortie.

EXEMPLE D'UTILISATION :

CDL

Soit la chaîne LABEL) en entrée, le prochain caractère en entrée étant le premier L,

- le code X'4C41' ('LA') est affecté à la zone de travail, à l'adresse indiquée par le contenu du registre R0.
- le contenu du registre R0 est incrémenté de 1 et le résultat est affecté à ce registre.
- le code X'4245' ('BE') est affecté à la zone de travail, à l'adresse indiquée par le contenu du registre R0.
- le contenu du registre R0 est incrémenté de 1 et le résultat est affecté à ce registre.
- le code X'4CA0' ('L ') est affecté à la zone de travail, à l'adresse indiquée par le contenu du

registre R0. Notez que dans ce cas le deuxième octet du mot contient X'A0' alors que le code ASCII d'un espace est X'20'. Le bit de gauche du deuxième octet a été mis à 1 pour indiquer que ce mot contient le dernier caractère de la chaîne.

- l'indicateur du prochain caractère en entrée est modifié pour indiquer le caractère suivant le dernier L de LABEL, soit).
- les caractères A, B, E, L ainsi que les espaces suivant ces caractères et le prochain caractère en entrée (soit ') ') sont affectés au vecteur de sortie sur l'imprimante.

CDP - Code parameter.

ADR	
111000	XX
ADRESSE	

CODE OPERATION : 111000

MODES D'ADRESSAGE : immédiat, direct
relatif, indirect

OPERATION :

La variable en entrée est codée dans sa forme interne et affecté à la zone de travail, à l'adresse indiquée par le contenu du registre 0. De plus, sa présence est indiquée dans le vecteur qui enregistre la présence des variables, soit dans le membre gauche, le membre droit ou la condition. Un code de 0 ou 3 à l'ADRESSE suivant le premier octet de l'instruction indique que l'un des deux vecteurs condition doit être utilisé, alors qu'un code de 1 indique le vecteur membre gauche et un code de 2 indique le vecteur membre droit. Enfin, le pointeur au prochain caractère en entrée est modifié pour indiquer le prochain caractère suivant la variable et tous les caractères lus, incluant les espaces, sont affectés au vecteur de sortie.

EXEMPLE D'UTILISATION :

CDP PARTIE

Soit la chaîne J*3 ,A) en entrée, le prochain caractère en entrée étant le J,

- le code X'2037' est affecté à la zone de travail, à l'adresse indiquée par le contenu du registre R0.
- la présence de cette variable arbre est indiquée dans le vecteur représenté par le code à l'adresse

PARTIE, en mettant l'indicateur associé à cette variable à 1.

- l'indicateur du prochain caractère en entrée est modifié pour indiquer le caractère suivant le chiffre 3, soit ..
- les caractères *, 3 ainsi que les espaces suivant ces caractères et le prochain caractère en entrée (soit ' , ') sont affectés au vecteur de sortie sur l'imprimante.

CHAPITRE 6.

CONCLUSION.

Le jeu d'instructions proposé s'est avéré efficace pour l'exécution du système pour lequel il a été développé, soit le système-q, résultant non seulement en une amélioration de la vitesse d'exécution mais surtout en une réduction de la capacité de mémoire requise par le système.

En plus, dans un projet réalisé en dehors de cette thèse, il a été démontré que la machine proposée s'apprête bien à des applications autres que les SYSTEMES-Q (Ref. 3), mais nécessitant le même type de structures de données. Cependant, certaines instructions sont encore trop spécialisées et pourraient subir des modifications afin de les rendre utile à un plus grand nombre d'applications.

Enfin, au point de vue amélioration, une instruction servant à copier tout un arbre (ou sous-arbre) d'une zone de mémoire à une autre, s'occupant de modifier les pointeurs automatiquement, serait une addition fort utile. Elle pourrait servir à la création de nouveaux arbres à partir de sous-arbres déjà existant ainsi qu'à la gestion de la mémoire utilisée pour conserver les structures d'arborescences.

APPENDICE A.LISTE DU MICROPROGRAMME.

Cet appendice contient la liste imprimée du microprogramme qui émule le jeu d'instructions présenté au chapitre 5. Ce programme fut réalisé sur un MICRODATA 1600 et occupe près de 2K (2048 mots de 16 bits) de mémoire fixe. L'introduction de cette liste explique l'utilisation des registres internes ainsi que de certaines zones de mémoire réservées au système. De plus, la plupart des instructions sont suivies de commentaires expliquant leur fonction exacte.

Le programme lui-même est subdivisé en cinq parties comme suit:

0000	TABLE DE BRANCHEMENT DES INSTRUCTIONS.
0041	INITIALISATIONS.
006D	VERIFICATION DES INTERRUPTIONS ET APPEL DES INSTRUCTIONS.
0097	SOUS-PROGRAMMES GENERAUX SERVANT A L'EXECUTION DE PLUSIEURS INSTRUCTIONS.
0278	ROUTINES D'EXECUTION DES INSTRUCTIONS INDIVIDUELLES.
077F	

EMULATEUR DU SYSTEME-0.

UTILISATION DES REGISTRES :

REGISTRES PRIMAIRES (PF).

1 REGISTRE D'INSTRUCTION : CODE OPERATION.
 2 REGISTRE D'INSTRUCTION : MCDES D'ADRESSAGE.
 3,4 DEPLACEMENT OU ENTIER.
 5 REGISTRE DU CODE CONDITIONNEL.
 6,7 ADRESSE DU PREMIER OPERANDE.
 8,9 PREMIER OPERANDE.
 10,11 DEUXIEME OPERANDE.
 14,15 COMPTEUR D'INSTRUCTIONS.

REGISTRES SECONDAIRES (SF).

1,2 INDICATEUR DU SOMMET DE LA PILE.
 10 INDICATEUR DU PROCHAIN CARACTERE A L'ENTREE.
 11 COMPTEUR DES CARACTERES A L'ENTREE.
 12 INDICATEUR DU PROCHAIN CARACTERE A L'IMPRIMANTE.
 13 LIMITE INFERIEURE DU VECTEUR IMPRIMANTE.
 14 LIMITE SUPERIEURE DU VECTEUR IMPRIMANTE.
 15 COMPTEUR DES CARACTERES A LA SORTIE.

UTILISATION DE LA MEMOIRE :

ADRESSES.

0010-0017 REGISTRES R0 A R3.
 0018-001F CONSTANTES.
 0020-006F VECTEURS PARAMETRES.
 0070-00BF VECTEUR FICHIER DE SORTIE.
 0100-017F VECTEUR FICHIER D'ENTREE.
 0180-01FF VECTEUR DE SORTIE SUR L'IMPRIMANTE.
 0200-07FF PILE DU SYSTEME.
 7FFA-7FFF VECTEURS D'INTERRUPTION.
 8000-FFFF ZONE DE TRAVAIL (VECTEUR GRAMMAIRE).

TYPES D'INSTRUCTIONS :

STRUCTURE

TYPE	CCOE	OP	ADR	OP
0	15	1	7	0
1	1	1	1	1
	15	8	7	0
2	1	1	1	1
	15	8	7	0

7

* * * TABLE DE BRANCHEMENT DES INSTRUCTIONS.

0000 1441	JP	INIT.	* INITIALISE LE SYSTEME.
0001 1C78	JP	JSB	* JUMP TO SUBROUTINE.
0002 1C78	JP	JSB	*
0003 1C78	JP	JSB	*
0004 1C78	JP	JSB	* JUMP TO SUBFUNCTION.
0005 1C78	JP	JSB	*
0006 1C78	JP	JSB	*
0007 1C78	JP	JSB	*
0008 1C8B	JP	JSB	* CLEAR.
0009 1C8B	JP	JSB	*
000A 1C8B	JP	JSB	*
000B 1C8B	JP	JSB	*
000C 1C8B	JP	JSB	* INCREMENT POINTER.
000D 1C8B	JP	JSB	*
000E 1C8B	JP	JSB	*
000F 1C8B	JP	JSB	*
0010 1C8B	JP	JSB	* COMPARE CHARACTERS.
0011 1C8B	JP	JSB	*
0012 1D05	JP	BRA	* BRANCH ON CONDITION.
0013 1D05	JP	BRA	*
0014 0684	JE	DEC	* CODE DECIMAL.
0015 1D13	JP	PSH	* PUSH STACK.
0016 1D2B	JP	TST	* TEST INDICATOR.
0017 1D32	JP	CRL	* CLASSIFY RULE.
0018 1D01	JP	BOX	* BRANCH ON INDEX.
0019 143F	JP	HLT	*
001A 1D75	JP	CCC	* COMPARE CHARACTER CODE.
001B 1D75	JP	CCC	*
001C 0684	JE	DEC	* CODE DECIMAL.
001D 1D0D	JP	MOV	* MOVE.
001E 1D05	JP	JMP	* UNCONDITIONAL JUMP.
001F 143F	JP	HLT	*
0020 0400	JE	BIN	* CODE BINARY.
0021 0454	JE	ADS	* ADD.
0022 0454	JE	ADS	* SUBTRACT.
0023 045F	JE	SPT	* SET POINTER.
0024 0469	JE	PRT	* PRINT CHARACTERS.
0025 0478	JE	CMP	* COMPARE.
0026 048B	JE	UNS	* INCLUDES.
0027 048C	JE	HRS	* EXCLUDES.
0028 04ED	JE	CLB	* COMPARE LABEL.
0029 0529	JE	CLT	* COMPARE LIST.
002A 0556	JE	CAR	* COMPARE TREE.
002B 057A	JE	CDL	* CODE LABEL.
002C 05CB	JE	APP	* ADD PRESENT PARAMETERS.
002D 05E3	JE	TPA	* TEST PARAMETER.
002E 143F	JP	HLT	*
002F 05FB	JE	RTS	* RETURN FROM SUBPROGRAM.
0030 0613	JE	RDC	* READ CHARACTER.
0031 0620	JE	WRC	* WRITE CHARACTER.
0032 062D	JE	SET	* SET INDICATOR.
0033 143F	JP	HLT	*
0034 0636	JE	TPP	* TEST PRESENT PARAMETERS.

0035	0651	JE	SPM	* SET PRINTER MODE.
0036	0620	JE	SET	* CLEAR INDICATOR.
0037	0348	JE	POP	* RCP STACK.
0038	06AE	JE	CDP	* CODE PARAMETER.
0039	143F	JP	HLT	
003A	0720	JE	DOT	* DECODE AND OUTPUT TREE.
003B	1460	JP	FETCH	* NO OPERATION.
003C	0712	JE	CPP	* CLEAR PRESENT PARAMETERS.
003D	143F	JP	HLT	
003E	0720	JE	DOT	* DECODE AND OUTPUT TREE.
003F	1780	HLT		* HALT.
0040	1460	JE	FETCH	
*** INITIALISATIONS.				
0041	BF06	INIT	ZOF 15.(U)	* LE COMPTEUR D'INSTRUCTIONS EST INITIALISE
0042	2E02	LF	14.2	* A L'ADRESSE DE DEPART.
0043	8500	ZOF	5	* LE CODE CONDITIONNEL ET LES INDICATEURS
				* D'INTERRUPTION SONT INITIALISES A 0.
0044	1200	LM	X'00.	
0045	2118	LF	1.X'18.	
0046	A113	WMF	1.(N)	* LE MOT A L'ADRESSE 0018 EST INITIALISE
0047	1127	LT	X'27.	* A '2710' (10,000).
0048	A1D3	WMF	1.1(N)	
0049	1110	LT	X'10.	
004A	A1D3	WMF	1.1(N)	* LE MOT A L'ADRESSE 001A EST INITIALISE
004B	1103	LT	X'03.	* A '03E8' (1,000).
004C	A1D3	WMF	1.1(N)	
004D	11E8	LT	X'E8.	
004E	A1D3	WMF	1.1(N)	* LE MOT A L'ADRESSE 001C EST INITIALISE
004F	1100	LT	X'00.	* A 'C064' (100).
0050	A1D3	WMF	1.1(N)	
0051	1164	LT	X'64.	
0052	A1D3	WMF	1.1(N)	* LE MOT A L'ADRESSE 001E EST INITIALISE
0053	1100	LT	X'00.	* A '000A' (10).
0054	A1D3	WMF	1.1(N)	
0055	110A	LT	X'0A.	
0056	1469	JP	RSP	*** TEMPERAIRE ***
0057	1080	SSF		
0058	2AFF	LF	10.X'FF.	* L'INDICATEUR DU PROCHAIN CARACTERE A
				* L'ENTREE EST INITIALISEE A -1.
0059	2B50	LF	11.X'50.	* LE COMPTEUR DES CARACTERES A L'ENTREE
				* EST INITIALISEE A 80.
005A	2C00	LF	12.X'00.	* L'INDICATEUR DU PROCHAIN CARACTERE A LA
				* SORTIE SUR L'IMPRIMANTE EST INITIALISEE A 0.
005B	2D00	LF	13.X'00.	* LA LIMITE INFERIEURE DU VECTEUR IMPRIMANTE
				* EST INITIALISEE A 0.
005C	2E7F	LF	14.X'7F.	* LA LIMITE SUPERIEURE DU VECTEUR IMPRIMANTE
				* EST INITIALISEE A 127.
005D	2F00	LF	15.X'00.	* LE COMPTEUR DES CARACTERES A LA SORTIE
				* EST INITIALISEE A 0.
005E	1120	LT	X'20.	
005F	A1200	LM	X'00.	* LE VECTEUR DU FICHIER DE SORTIE EST INITIALISE
0060	216F	LF	1.X'6F.	* A DES ESPACES.
0061	A1D3	WMF	1.1(N)	
0062	6141	CP	1.X'41.	

0063	1461	JP	*-2	*	
0064	1201	LM	X'01'	*	LE VECTEUR DE SORTIE SUR L'IMPRIMANTE EST
0065	217F	LF	1.X'7F'	*	INITIALISE A DES ESPACES.
0066	A103	WMF	1.1(N)	*	
0067	6101	CP	1.X'01'	*	
0068	1466	JP	*-2	*	

* * * EXECUTION DE L'INSTRUCTION RSP.

0069	1080	RSP	SSF		
006A	22FE	LF	2.X'FE'	*	LE SOMMET DE LA PILE EST INITIALISE.
006B	2101	LF	1.X'01'	*	
006C	1040	SPF			

EJE

* SOUS-PROGRAMME QUI ADDITIONNE LE CONTENU DES REGISTRES PF10,11
* AU CONTENU DES REGISTRES PF3,9.

0097 CB01	ADDER	MOV	11.(T)	
0098 8930	ADD	3.TC		* LE REGISTRE ETAT PFO EST MIS A JOUR,
0099 CA01	MOV	10.(T)		DEPENDANT DU RESULTAT DE L'ADDITION.
009A 8880	ADD	8.LTC		
009B 1020	RTN			

* SOUS-PROGRAMME QUI CALCULE LES ADRESSES DES OPERANDES DEPENDANT DU
* MODE D'ADRESSAGE, ET AFFECTE L'ADRESSE DU PREMIER OPERANDE AUX
* REGISTRES PF6,7. LE CONTENU DU PREMIER OPERANDE AUX REGISTRES
* PF8,9 ET LE CONTENU DU SECOND OPERANDE AUX REGISTRES PF10,11.

* LE REGISTRE PF12 DOIT CONTENIR LE CODE DU TYPE D'INSTRUCTION.
* TYPE 0 : MODE D'ADRESSAGE DANS LE MEME OCTET QUE LE CODE
* OPERATION.

* TYPE 1 & 2 : MODE(S) D'ADRESSAGE DANS L'OCTET SUIVANT CELUI
* DU CODE OPERATION.

* N.B. : LES INSTRUCTIONS DE TYPES 0 ET 1 NE REQUIERENT QU'UN
* OPERANDE ALORS QUE CEUX DU TYPE 2 NECESSITENT DEUX
* OPERANDES.

009C 4CFF	ADRES	TZ	12.X'FF'	* INSTRUCTION QU TYPE 0?
009D 14A2	JP	AD1		

* INSTRUCTION TYPE 0.

009E C101	MOV	1.(T)		* LE MODE D'ADRESSAGE SE TROUVANT DANS LE
009F B220	CPY	2.T		* REGISTRE PF1 EST TRANSFERE AU REGISTRE PF2.
00A0 8C40	INC	12		* LE TYPE EST MIS A 1. INDICANT QUE
00A1 148B	JP	AD2		L'INSTRUCTION NECESSITE UN OPERANDE.

* INSTRUCTION TYPE 1 OU 2.

00A2 CE02	MOV	14.(M)		* L'OCTET SUIVANT CELUI DU CODE OPERATION EST
00A3 AF03	RMF	15.(N)		* LD ET AFFECTE AU REGISTRE PF2.
00A4 8F40	INC	15		* DE PLUS, LE COMPTEUR D'INSTRUCTIONS EST MIS
00A5 B240	CPY	2.T		* A JOUR.
00A6 8E30	ADD	14.L		

00A7 5C02	TN	12.X'02'		* INSTRUCTION DU TYPE 2?
00A8 148B	JP	AD2		* NON: BRANCHEMENT AU TRAITEMENT DE L'OPERANDE.

* INSTRUCTION TYPE 2.

00A9 4209	TZ	2.X'0B'		
00AA 14AF	JP	*+5		
00AB 5204	TN	2.X'0A'		* MODE REGISTRE?
00AC 14B1	JP	*+5		
00AD 2C06	LF	12.6		
00AE 148B	JP	AD2		* SI UN DES 2 MODES EST LE MODE REGISTRE OU GRAMMAIRE INDEXEE REG 0, LE TYPE EST MIS A 6.

00AF 5207	TN	2.X'07'	* MCDE GRAMMAIRE INDEXEE REG 0?
00B0 14AD	JP	*-3	
00B1 4280	TZ	2.X'80'	
00B2 1486	JP	*+4	
00B3 5240	TN	2.X'40'	* MCDE REGISTRE?
00B4 1483	JP	*+4	
00B5 14AD	JP	*-9	
00B6 5270	TN	2.X'70'	* MCDE GRAMMAIRE INDEXEE REG 0?
00B7 14AD	JP	*-10	
00B8 1102	LT	2	* SI AUCUN DES 2 MODES N'EST LE MODE REGISTRE
00B9 8F20	ADD	15.T	* OU GRAMMAIRE INDEXEE REG 0, LE COMPTEUR
00BA 8E80	ADD	14.L	* D'INSTRUCTIONS EST INCREMENTE DE 2 POUR
			* INDICUER LE MOT CONTENANT L'ADRESSE DU
			* SECOND OPERANDE.
00BB 4204	TZ	2.X'04'	* MCDE REGISTRE OU REGISTRE INDEXE?
00BC 14C1	JP	REG	
00BD 5208	TN	2.X'08'	
00BE 14D0	JP	AD4	
00BF 4207	TZ	2.X'07'	* MCDE GRAMMAIRE INDEXEE REG 0?
00C0 14D0	JP	AD4	
			* TRAITEMENT DES MODES D'ADRESSAGE NECESSITANT UN REGISTRE.
			* *
00C1 B600	REG		* CALCULE L'ADRESSE DU REGISTRE DEPENDANT DES
00C2 2703	ZOF	6	* 2 DERNIERS BITS DU MODE D'ADRESSAGE ET
00C3 C201	MOV	2.(T)	* L'ADRESSE AUX REGISTRES PF6.7.
00C4 E720	AND	7.T	
00C5 F700	SFL	7	
00C6 3710	AF	7.X'10'	
00C7 C602	MOV	6.(M)	* LE CONTENU DU REGISTRE EST LU ET AFFECTE AUX
00C8 A703	RMF	7.(N)	* REGISTRES PF8.9.
00C9 B920	CPY	8.T	
00CA A7C8	RMF*	7.I(N)	
00CB B920	CPY	9.T	
00CC 5208	TN	2.X'08'	* MCDE REGISTRE?
00CD 1531	JP	ADE	* OUI: LE TRAVAIL EST DEJA FAIT.
00CE 5204	TN	2.X'04'	* MCDE GRAMMAIRE INDEXEE REG 0?
00CF 1507	JP	GRI+5	* OUI: BRANCHEMENT AU MODE GRAMMAIRE.
00D0 CE02	MOV	14.(M)	* LE MOT SUIVANT L'INSTRUCTION CONTENANT
00D1 AF03	RMF	15.(N)	* L'ADRESSE, LE DEPLACEMENT OU L'OPERANDE
00D2 8F43	INC	15.(N)	* EST LU ET AFFECTE AUX REGISTRES PF6.7.
00D3 B620	CPY	6.T	
00D4 AE82	RMF	14.L(M)	
00D5 8F40	INC	15	* DE PLUS, LE COMPTEUR D'INSTRUCTIONS EST MIS
00D6 B720	CPY	7.T	* A JOUR.
00D7 8E80	ADD	14.L	
			* TRAITEMENT DU MODE IMMEDIAT.
			* *
			* *
00D8 420F	TZ	2.X'0F'	* MCDE IMMEDIAT?
00D9 14E1	JP	DIR	
00DA C601	MOV	6.(T)	* L'OPERANDE VENANT D'ETRE AFFECTE AUX REGISTRES
00DB 8820	CPY	8.T	* PF6.7 EST TRANSFERE AUX REGISTRES PF8.9.
00DC C701	MOV	7.(T)	
00DD B920	CPY	9.T	
00DE B600	ZOF	6	* L'ADRESSE DANS PF6.7 EST MISE A 0 POUR
00DF B700	ZOF	7	* INDICUER UNE ADRESSE NULLE.

LOCN	CODE	FLAGS	LABELS	OP	*	OPERANDS	COMMENTS	PAGE	8
00E0	1531			JP		ADE			
* TRAITEMENT DU MODE DIRECT.									
00E1	420E			TZ		2.X'0E'	* MODE DIRECT?		
00E2	14E6			JP		REL			
00E3	F700			SFL		7	* L'ADRESSE DANS LES REGISTRES PF6,7 EST DECALEE		
00E4	F683			SFL		6.L	* D'UN BIT VERS LA GAUCHE.		
00E5	1522			JP		OPR			
* TRAITEMENT DES MODES NECESSITANT UN DEPLACEMENT.									
00E6	CF01			REL					
00E7	8720			MOV		15.(T)	* LE CONTENU DU COMPTEUR D'INSTRUCTIONS EST		
00E8	CE01			ADD		7.T	* AJOUTE AU DEPLACEMENT DANS LES REGISTRES		
00E9	8680			MOV		14.(T)	* PF6,7.		
00EA	5001			ADD		6.LTC			
00EB	14EF			TN		0.X'01'	* DEPASSEMENT DE CAPACITE?		
00EC	1120			JP		*+4			
00ED	C520			LT		X'20'	* CUI: LE BIT DE DEPASSEMENT DE CAPACITE		
00EE	152A			LOR		5.T	* DANS LE REGISTRE PF5 EST MIS A 1.		
00EF	F720			JP		CPR+8			
00F0	F700			SFR		7	* LE BIT DE DROITE DE L'ADRESSE EST MIS A 0		
00F1	520D			SFL		7	* POUR REFERER A UNE ADRESSE PAIRE.		
00F2	1522			TN		2.X'0D'	* MCDE RELATIF?		
				JP		OPR	* OUI: L'ADRESSE VIENT D'ETRE CALCULEE.		
* TRAITEMENT DU MODE INDIRECT.									
00F3	420C			TZ		2.X'0C'	* MODE INDIRECT?		
00F4	14FB			JP		REI			
00F5	C602			MOV		6.(M)	* LE MOT A L'ADRESSE CONTENUE DANS LES REGISTRES		
00F6	A703			RMF		7.(N)	* PF6,7 EST LU ET AFFECTE AUX REGISTRES PF6,7.		
00F7	8620			CPY		6.T			
00F8	A7CB			RMF*		7.I(N)			
00F9	B720			CPY		7.T			
00FA	14E3			JP		DIR+2	* BRANCHEMENT AU TRAITEMENT DU MODE DIRECT.		
* TRAITEMENT DU MODE REGISTRE INDEXE.									
00FB	5208			REI					
00FC	1502			TN		2.X'08'			
00FD	5204			JP		GRI			
00FE	1502			TN		2.X'04'	* MODE REGISTRE INDEXE?		
00FF	F900			JP		GRI			
0100	F880			SFL		9	* L'INDEX DANS LES REGISTRES PF8,9 EST DECALE		
0101	1515			SFL		8.L	* D'UN BIT VERS LA GAUCHE.		
				JP		IDX			
* TRAITEMENT DU MODE GRAMMAIRE INDEXEE.									
0102	C602			GRI					
0103	A703			MOV		6.(M)	* LE MOT A L'ADRESSE CCNTENUE DANS LES REGISTRES		
0104	B820			RMF		7.(N)	* PF6,7 EST LU ET AFFECTE AUX REGISTRES PF8,9.		
0105	A7CB			CPY		8.T			
0106	B920			RMF*		7.I(N)			
0107	F900			CPY		9.T			
0108	F880			SFL		9	* L'INDEX DANS PF8,9 EST DECALE D'UN BIT VERS		
0109	5202			SFL		8.L	* LA GAUCHE.		
				TN		2.X'02'	* MODE GRAMMAIRE INDEXEE MOINS 1 OU MOINS 2?		

LUCN	CODE	FLAGS	LABELS	OP	* OPERANDS	COMMENTS	PAGE	9
010A	1513		JP	+	9			
010B	1102		LT	2		* CUI: L'ADRESSE DANS PF8.9 EST DECREMENTE DE 2.		
010C	9920		SBT	9,T		*		
010D	5980		SBT	8,L		* MODE GRAMMAIRE INDEXEE MCINS 2?		
010E	5201		JTN	2,X*01		*		
010F	1513		JP	+	4			
0110	1102		LT	2		* CUI: L'ADRESSE DANS PF8.9 EST ENCORE DECREMENTE DE 2.		
0111	9920		SBT	9,T		*		
0112	9890		SBT	8,L		* L'ADRESSE DE LA GRAMMAIRE CODEE EST AFFECTEE AUX REGISTRES PF6.7.		
0113	2603		LF	6,X*03		*		
0114	2700		LF	7,X*00		*		
* TRAITEMENT DE L'INDEX POUR LES MODES GRAMMAIRE INDEXEE ET								
* REGISTRE INDEXE.								
* INDEX								
0115	C901		MOV	9,(T)		* L'INDEX DANS LES REGISTRES PF8.9 EST AJOUTE		
0116	8720		ADD	7,T		* A L'ADRESSE DANS LES REGISTRES PF6.7.		
0117	C801		MOV	8,(T)		*		
0118	8680		ADD	6,LTC		* DEPASSEMENT DE CAPACITE?		
0119	5001		TN	0,X*01		*		
011A	151E		JP	+	4			
011B	1120		LT	X*20		* CUI: LE BIT DE DEPASSEMENT DE CAPACITE		
011C	C520		LOR	5,T		* DANS LE REGISTRE PF5 EST MIS A 1.		
011D	152A		JP	OPR+8		*		
011E	5208		TN	2,X*08		* MODE GRAMMAIRE INDEXEE?		
011F	1522		JP	OPR		*		
0120	5204		TN	2,X*04		*		
0121	152C		JP	OPR+10		* CUI.		
* AFFECTATION DE L'OPERANDE.								
* OPR								
0122	6680		CP	6,X*80		* L'ADRESSE DANS LES REGISTRES PF6.7 REFERE A LA ZONE DE TRAVAIL?		
0123	1525		JP	+	2			
0124	1528		JP	+	4	* CUI.		
0125	66F8		CP	6,X*F8		* L'ADRESSE DANS LES REGISTRES PF6.7 REFERE A LA ZONE PRIVILEGEE.		
0126	1529		JP	+	2			
0127	152C		JP	+	5			
0128	1140		LT	X*40		* CUI: LE BIT D'ADRESSAGE DANS LE REGISTRE PF5 EST MIS A 1.		
0129	C520		LOR	5,T		*		
012A	B600		ZOF	6		* L'ADRESSE DANS LES REGISTRES PF6.7 EST MISE A 0.		
012B	B700		ZOF	7		*		
012C	C602		MOV	6,(M)		* LE MOT A L'ADRESSE CONTENUE DANS LES REGISTRES PF6.7 EST LU ET AFFECTE AUX REGISTRES PF8.9.		
012D	A703		RMF	7,(N)		*		
012E	B820		CPY	8,T		*		
012F	A7CB		RMF*	7,I(N)		*		
0130	B920		CPY	9,T		*		
* FIN DU TRAITEMENT.								
* ADE								
0131	4C01		TZ	12,X*01		* TYPE = 1?		
0132	1020		RTN			* CUI: LE TRAITEMENT DES ADRESSES EST TERMINE.		
0133	4C02		TZ	12,X*02		* TYPE = 0?		
0134	1539		JP	ADF		*		
0135	1102		LT	2		* CUI: LE COMPTEUR D'INSTRUCTIONS EST INCREMENTE DE 2 POUR INDIOUER L'ADRESSE DE L'INSTRUCTION SUIVANTE.		
0136	8F20		ADD	15,T		*		
0137	8E80		ADD	14,L		*		
0138	1020		RTN			*		

* * TRAITEMENT DE L'AUTRE OPERANDE.

* ADF
0139 C801 MOV 8.(T) * L'OPERANDE DANS LES REGISTRES PF8.9 EST
013A BA20 CPY 10.T * TRANSFERE DANS LES REGISTRES PF10.11.
013B C901 MOV 9.(T) *
013C BB20 CPY 11.T *
013D 7220 SRF 2 * LE MODE DE LA PREMIERE ADRESSE EST PLACE
* * DANS LES 4 PREMIERS BITS DU REGISTRE PF2.
013E 5C04 TN 12,X.04 * TYPE = 62
013F 1542 JP *+3
0140 2C01 LF 12,1 * CUI: LE TYPE DEVIENT 1.
0141 14BB JP AD2 * RETOURNE EXTRAIRE LE PREMIER OPERANDE.
0142 BC00 ZOF 12 * NON: LE TYPE DEVIENT 0.
0143 1104 LT 4 * LE COMPTEUR D'INSTRUCTIONS EST DECREMENTE DE
0144 9F20 SBT 15.T * 4 POUR INDiquer LE MOT CONTENANT L'ADRESSE
0145 9E80 SBT 14.L * DU PREMIER OPERANDE.
0146 14BB JP AD2 * RETOURNE EXTRAIRE LE PREMIER OPERANDE.

* * SOUS-PROGRAMME QUI CALCULE L'ADRESSE ABSOLUE A PARTIR DU
* * DEPLACEMENT ET L'AFECTE AUX REGISTRES PF3.4.

* DEPL
0147 CE02 MOV 14.(M) * LE DEPLACEMENT SUIVANT L'INSTRUCTION EST LU
0148 AF03 RMF 15.(N) * ET AFFECTE AUX REGISTRES PF3.4.
0149 8F43 INC 15.(N) *
014A B320 CPY 3.T *
014B AE82 RMF 14.L(M) *
014C 8F40 INC 15 * DE PLUS, LE COMPTEUR D'INSTRUCTIONS EST
014D B420 CPY 4.T * MIS A JOUR.
014E 8E80 ADD 14.L *
014F CF01 MOV 15.(T) * L'ADRESSE ABSOLUE EST CALCULEE EN AJOUTANT
0150 8420 ADD 4.T * LE CCNTENU DU COMPTEUR D'INSTRUCTIONS AU
0151 CE01 MOV 14.(T) * DEPLACEMENT DANS LES REGISTRES PF3.4.
0152 83A0 ADD 3.LT *
0153 1020 RTN

* * SOUS-PROGRAMME QUI EXTRAIT L'ELEMENT AU SOMMET DE LA PILE ET
* * L'AFECTE AUX REGISTRES PF8.9.

* DPILE SSF
0154 1080 CP 1,X'FE' * DEBOREMENT DE PILE?
0155 61FE JP *+2
0156 1558 JP *+7
0157 155E JP *
0158 1040 SPF 8 * QUI: LE CCNTENU DES REGISTRES PF8.9 EST
0159 B800 ZOF 9 * MIS A 0.
015A B900 ZOF 9 *
015B 1110 LT X.10' * L'INDICATEUR DE DEBOREMENT DE PILE DANS
015C C520 LDR 5.T * LE REGISTRE PF5 EST MIS A 1.
015D 1020 RTN
015E C102 MOV 1.(M) * LE MOT A L'ADRESSE DANS LES REGISTRES SF1.2
015F A203 RMF 2.(N) * (LE SOMMET DE LA PILE) EST LU ET AFFECTE
0160 B320 CPY 3.T * AUX REGISTRES SF3.4.
0161 A2C8 RMF* 2.I(N) *
0162 B420 CPY 4.T *
0163 1102 LT 2 * LE SCMMET DE LA PILE DANS LES REGISTRES SF1.2

COMMENTS

LOCN CODE FLAGS LABELS OP * OPERANDS

0164 9220 SBT 2,T * EST DECREMENTE DE 2.
 0165 9180 SBT 1,L *
 0166 C301 MOV 3,(T) * LE CCONTENU DES REGISTRES SF3.4 EST TRANSFERE
 0167 1040 SPF 8,T * AUX REGISTRES PF8.9.
 0168 B820 CPY *
 0169 1080 SSF *
 016A C401 MOV 4,(T) *
 016B 1040 SPF *
 016C B920 CPY 9,T *
 016D 1020 RTN *

* *
 * * SOUS-PROGRAMME QUI PLACE LES PREMIERS ELEMENTS DE LA PILE, DONT
 * * LE SOMMET EST DANS LES REGISTRES PF3.4, DANS LES REGISTRES
 * * PF10,11, PF8.9 ET PF6.7.

* DPL AF 13,X'FE' * L'INDICATEUR DU NOMBRE D'ELEMENTS DANS LA
 016E 3DFE INC 4,(N) PILE EST DECREMENTE DE 2.
 016F 8443 LU X'BB'
 0170 1688 LF 6,X'BB'
 0171 2688 LF 2.4 * LE REGISTRE PF2 INDIQUE LE NOMBRE D'OCTETS
 0172 2204 * A ETRE RETIRES DE LA PILE.
 * INSTRUCTION CLT OU CAR?

0173 4120 TZ 1,X'20' *
 0174 1577 JP DPL * OUI: 2 AUTRES OCTETS A RETIRER DE LA PILE.
 0175 3202 AF 2.2 * L'INDICATEUR DU NOMBRE D'ELEMENTS DANS LA
 0176 9D40 DEC 13 * PILE EST DECREMENTE DE 1.
 * L'OCTET EST LU ET AFFECTE AU REGISTRE
 * CORRESPONDANT.

0177 A302 RMF 3,(M)
 0178 0020 EDT 0,T
 0179 9240 DEC 2 * AUTRES OCTETS A ETRE LUS?

017A 52FF TN 2,X'FF' *
 017B 1580 JP DP2 * OUI: LE REGISTRE U INDIQUE LE REGISTRE AUQUEL
 017C 9646 DEC 6,(U) * SERA EFFECTUE LE PROCHAIN OCTET DANS LA PILE.
 * LES REGISTRES PF3.4 INDICENT L'OCTET SUIVANT
 * DANS LA PILE.

017D 9443 DEC 4,(N)
 017E 9382 SBT 3,L(M)
 017F 1577 JP DPL * LE NOUVEAU SOMMET DE LA PILE EST CALCULE ET
 0180 1102 LT 2 * AFFECTE AUX REGISTRES PF3.4.
 0181 9420 SBT 4,T *
 0182 9380 SBT 3,L *
 0183 1600 LU 0
 0184 1020 RTN *

* *
 * * SOUS-PROGRAMME QUI PLACE LE CCONTENU DES REGISTRES PF6.7, PF8.9
 * * ET PF10,11 SUR LA PILE DONT LE SOMMET EST DANS LES REGISTRES
 * * PF3.4.

* EPL AF 13,3 * L'INDICATEUR DU NOMBRE D'ELEMENTS DANS LA
 0185 3D03 INC 4 PILE EST INCREMENTE DE 3.
 0186 8440 LF 2,X'CB'

0187 2206 MOV 2,(U)
 0188 C206 TN 1,X'20' * INSTRUCTION DNS OU HRS?
 0189 5120 JP EPI

018A 158E INC 2,(U) * CUI: SEULEMENT 4 OCTETS DOIVENT ETRE AFFECTES
 018B 8246 INC 2,(U) * A LA PILE.
 018C 8246 INC 2,(U) * L'INDICATEUR DU NOMBRE D'ELEMENTS DANS LA
 018D 9D40 DEC 13

LOCN	CODE	FLAGS	LABELS	OP	* OPERANDS	COMMENTS	PAGE	12
018E	8443		EPI	INC	4.(N)	* PILE EST DÉCREMENTÉE DE 1.		
018F	A392		WMF		3.L(M)	* LE CONTENU DU REGISTRE EST AFFECTÉ AU SOMMET DE LA PILE.		
0190	0001		EOT		0.(T)			
0191	8246		INC		2.(U)	* LE REGISTRE U INDIQUE LE REGISTRE DONT LE CONTENU SERA AFFECTÉ A LA PILE.		
0192	6234		CP		2.X.34	* AUTRES OCTETS A ETRE AFFECTE A LA PILE?		
0193	158E		JP		EPI			
0194	9440		DEC		4	* NCN: LE NOUVEAU SOMMET DE LA PILE EST CALCULE.		
0195	1600		LU		0			
0196	1020		RTN					
* * SOUS-PROGRAMME QUI EXTRAIT L'ENTIER DE L'INSTRUCTION ET L'AFFECTE AUX REGISTRES PF3,4.								
0197	CE02		ENT	MOV	14.(M)	* L'ENTIER SUIVANT L'INSTRUCTION EST LU ET		
0198	AF03		RMF		15.(N)	* AFFECTE AUX REGISTRES PF3,4.		
0199	8F40		INC		15	* DE PLUS, LE COMPTEUR D'INSTRUCTIONS EST		
019A	B420		CPY		4.T	* MIS A JOUR.		
019B	8E80		ADD		14.L			
019C	B300		ZOF		3			
019D	4480		YZ		4.X.80	* ENTIER NEGATIF?		
019E	23FF		LF		3.X.FF	* CUI: LE BIT DE GAUCHE DU REGISTRE PF4 EST		
019F	1020		RTN			* PROPAGE DANS LE REGISTRE PF3.		
* * SOUS-PROGRAMME QUI AFFECTE L'INDICATEUR DU FILS AINE DE L'ELEMENT PRESENT, DONT L'INDICATEUR SE TROUVE DANS LES REGISTRES PF8,9.								
01A0	F900		FILS	SFL	9	* L'ADRESSE ABSOLUE DE L'ELEMENT DONT		
01A1	F880		SFL		8.L	* L'INDICATEUR SE TROUVE DANS LES REGISTRES		
01A2	1100		LT		X.00	* PF8,9 EST CALCULEE.		
01A3	B920		ADD		9.T			
01A4	1103		LT		X.03			
01A5	88A0		ADD		8.LT			
01A6	11FC		LT		X.FC	* CETTE ADRESSE EST DECREMENTEE DE 4.		
01A7	B920		ADD		9.T			
01A8	9880		SBT		8.L			
01A9	C802		MOV		8.(M)	* L'INDICATEUR A L'ADRESSE CALCULEE EST LU		
01AA	A903		RMF		9.(N)	* ET AFFECTE AUX REGISTRES PF8,9.		
01AB	B820		CPY		8.T			
01AC	A9CB		RMF		9.I(N)			
01AD	B920		CPY		9.T			
01AE	1020		RTN					
* * SOUS-PROGRAMME QUI AFFECTE L'INDICATEUR DU FILS AINE DE L'ELEMENT PRESENT, DONT L'INDICATEUR SE TROUVE DANS LES REGISTRES PF10,11.								
01AF	FB00		F2LS	SFL	11	* L'ADRESSE ABSOLUE DE L'ELEMENT DONT		
01B0	FA80		SFL		10.L	* L'INDICATEUR SE TROUVE DANS LES REGISTRES		
01B1	1100		LT		X.00	* PF10,11 EST CALCULEE.		
01B2	8B20		ADD		11.T			
01B3	1103		LT		X.03			

01B4 8AA0 ADD 10,LT * CETTE ADRESSE EST DECREMENTEE DE 4.
01B5 11FC LT X,FC: *

01B6 8B20 ADD 11,7 *
01B7 9A80 SBT 10,L *
01B8 CA02 MOV 10,(M) * L'INDICATEUR A L'ADRESSE CALCULEE EST LU
01B9 AB03 RMF 11,(N) * ET AFFECTE AUX REGISTRES PF10,11.

01BA BA20 CPY 10,7 *
01BB ABCB RMF* 11,I(N) *
01BC EB20 CPY 11,7 *
01BD 1020 RTN

*
*
* SOUS-PROGRAMME QUI AFFECTE L'INDICATEUR DU FILS AINE DE L'ELEMENT
* PRESENT, DONT L'INDICATEUR SE TROUVE DANS LES REGISTRES PF10,11.
* DANS LES REGISTRES PF6,7.

F3LS MOV 10,(T) * L'ADRESSE ABSOLUE DE L'ELEMENT DONT
01BE CA01 CPY 6,7 * L'INDICATEUR SE TROUVE DANS LES REGISTRES
01BF CB01 MOV 11,(T) * PF10,11 EST CALCULEE ET AFFECTEE AUX
01C1 B720 CPY 7,7 * REGISTRES PF6,7.

01C2 F700 SFL 7 *
01C3 F680 SFL 6,L *
01C4 1100 LT X,00: *

01C5 8720 ADD 7,7 *
01C6 1103 LT X,03: *
01C7 86A0 ADD 6,LT * CETTE ADRESSE EST DECREMENTEE DE 4.

01C8 11FC LT X,FC: *
01C9 8720 ADD 7,7 *
01CA 9680 SBT 6,L *
01CB C602 MOV 6,(M) * L'INDICATEUR A L'ADRESSE CALCULEE EST LU

01CC A703 RMF 7,(N) * ET AFFECTE AUX REGISTRES PF6,7.
01CD B620 CPY 6,T *
01CE A7CB RMF* 7,I(N) *
01CF B720 CPY 7,7 *
01D0 1020 RTN

*
*
* SOUS-PROGRAMME QUI AFFECTE L'INDICATEUR DU FRERE SUIVANT DE
* L'ELEMENT PRESENT, DONT L'INDICATEUR EST DANS LES REGISTRES
* PF8,9, DANS LES REGISTRES PF8,9.

FIERE SFL 9 * L'ADRESSE ABSOLUE DE L'ELEMENT DONT
01D1 F900 SFL 8,L * L'INDICATEUR SE TROUVE DANS LES REGISTRES
01D2 F860 LT X,00: * PF8,9 EST CALCULEE.

01D3 1100 ADD 9,7 *
01D4 8920 LT X,03: *
01D5 1103 ADD 8,LT *
01D6 88A0 LT X,FE: * CETTE ADRESSE EST DECREMENTEE DE 2.

01D7 11FE ADD 9,7 *
01D8 8920 SBT 9,L *
01D9 9880 MOV 8,(M) * L'INDICATEUR A CETTE ADRESSE EST LU ET
01DA CB02 RMF 9,(N) * AFFECTE AUX REGISTRES PF8,9.

01DB A903 CPY 8,T *
01DC B820 RMF* 9,I(N) *
01DD A9CB CPY 9,7 *
01DE B920 RTN

*

* * SOUS-PROGRAMME QUI AFFECTE L'INDICATEUR DU FRERE SUIVANT DE
* L'ELEMENT PRESENT, DONT L'INDICATEUR EST DANS LES REGISTRES
* * PF10.11, DANS LES REGISTRES PF10.11.

01E0 FB00	F2ERE	SFL	11	* L'ADRESSE ABSOLUE DE L'ELEMENT DONT
01E1 FA80	SFL	10,L		* L'INDICATEUR SE TROUVE DANS LES REGISTRES
01E2 1100	LT	X'00'		* PF10.11 EST CALCULEE.
01E3 8B20	ADD	11,T		
01E4 1103	LT	X'03'		
01E5 8AA0	ADD	10,LT		
01E6 11FE	LT	X'FE'		* CETTE ADRESSE EST DECREMENTEE DE 2.
01E7 8B20	ADD	11,T		
01E8 9A80	SFL	10,L		
01E9 CA02	MOV	10,(M)		* L'INDICATEUR A CETTE ADRESSE EST LU ET
01EA AB03	RMF	11,(N)		* AFFECTE AUX REGISTRES PF 10.11.
01EB BA20	CPY	10,T		
01EC ABCB	RMF*	11,I(N)		
01ED BB20	CPY	11,T		
01EE 1020	RTN			

* * SOUS-PROGRAMME QUI MODIFIE L'INDICATEUR DU PROCHAIN CARACTERE
* A L'ENTREE POUR INDiquer LE PROCHAIN CARACTERE QUI N'EST PAS
* * UN ESPACE.

01EF 1080	LIRE	SSF		* L'INDICATEUR DU PROCHAIN CARACTERE A L'ENTREE
01F0 117F	LT	X'7F'		* * DANS LE REGISTRE SF10 EST INCREMENTE DE 1.
01F1 8A40	INC	10		
01F2 EA20	AND	10,T		
01F3 5B40	INC	11		* LE COMPTEUR DES CARACTERES A L'ENTREE DANS
				* LE REGISTRE SF11 EST INCREMENTE DE 1.
01F4 6BAF	CP	11,X'AF'		
01F5 15FE	JP	LR1		
01F6 BB40	POF	11		* OUI: LE COMPTEUR EST MIS A 1.
01F7 1040	SPF			
01F8 4508	TZ	5,X'08'		* LA PROCHAINE CARTE A DEJA ETE LUE?
01F9 1C05	JP	LR2		
01FA 1000	NOP			
01FB 1108	LT	X'08'		* NGN: LA PROCHAINE CARTE EST LUE.
01FC C520	LOR	5,T		* L'INDICATEUR DE LECTURE DE LA PROCHAINE
01FD 1C05	JP	LR2		* CARTE EST MIS A 1.
01FE 1128	LT	X'28'		* LE COMPTEUR DES CARACTERES A L'ENTREE DANS
01FF DB38	XOR*	11,TC		* LE REGISTRE SF11 EST EGAL A 40?
0200 5004	TN	0,X'04'		
0201 1C05	JP	LR2		
0202 1040	SPF			
0203 11F7	LT	X'F7'		* CUI: L'INDICATEUR DE LECTURE DS LA PROCHAINE
0204 E520	AND	5,T		* CARTE EST MIS A 0.
0205 1080	SSF			
0206 1201	LM	X'01'		* LE PROCHAIN CARACTERE A L'ENTREE EST LU ET
0207 AA03	RMF	10,(N)		* AFFECTE AU REGISTRE SF9.
0208 B920	CPY	9,T		
0209 495F	TZ	9,X'5F'		* LE CARACTERE DANS LE REGISTRE SF9 EST
020A 1060	RSP			* UN ESPACE?
020B 4920	TZ	9,X'20'		
020C 15EF	JP	LIRE		* CUI.

COMMENTS

OPERANDS

LOCN CODE FLAGS LABELS OP *

020D 1060 RSP

*
* SOUS-PROGRAMME QUI AFFECTE AU VECTEUR DE SORTIE TOUS LES
* CARACTERES DU VECTEUR D'ENTREE ENTRE LA POSITION DONNEE PAR
* L'INDICATEUR DANS LE REGISTRE SF3, PLUS UN, ET CELLE DONNEE
* PAR L'INDICATEUR DANS LE REGISTRE SF10.

020E 1104 NPRNT LT X'04'
020F C520 LOR 5,T
0210 1201 LM X'01'
0211 1080 SSF
0212 8340 INC 3
0213 A303 PMF 3,(N)
0214 1040 SPF
0215 B920 CPY 9,T
0216 1C2E JP PCHAR
0217 1080 NP1
0218 C301 SSF 3,(T)
0219 DA38 MOV 10,TC
XOR*
* TN 0,X'04'
JP NPRNT+4
SPF
LT X'FB'
AND 5,T
RSP

*
* SOUS-PROGRAMME QUI AFFECTE LE CARACTERE DANS LE REGISTRE PF9
* AU VECTEUR DU FICHIER DE SORTIE.

0220 1080 OUTPUT SSF
0221 1200 LM X'00'
0222 1170 LT X'70'
0223 8F2B ADD* 15,T(N)
0224 1040 SPF
0225 C901 MOV 9,(T)
0226 A010 WMF 0
0227 1080 SSF 15
0228 8F40 INC
* CP 15,X'80'
RSP
NOP
ZOF 15
RSP

*
* SOUS-PROGRAMME QUI AFFECTE LE CARACTERE DANS LE REGISTRE PF9
* AU VECTEUR DE SORTIE SUR L'IMPRIMANTE.

022E 1080 PCHAR SSF
022F 1201 LM X'01'
0230 1180 LT X'80'
0231 CC2B LOR* 12,T(N)
0232 1040 SPF

0233 C901 MOV 9.(T) * LE CARACTERE DANS LE REGISTRE PF9 EST
0234 A010 WMF 0 * AFFECTE A CETTE ADRESSE.
0235 1080 SSF
0236 8C40 INC 12 * L'INDICATEUR DU PROCHAIN CARACTERE A
0237 CC01 MOV 12.(T) * L'IMPRIMANTE EST INCREMENTE DE 1 ET
0238 9E38 SBT* 14.TC * COMPARE AVEC LA LIMITE SUPERIEURE DU
* VECTEUR DE SORTIE SUR L'IMPRIMANTE.
0239 5002 TN 0.X*02* * L'INDICATEUR DEPASSE LA LIMITE?
023A 1C3E JP PFIN * NON.
023B 1000 PUT * LE VECTEUR DE SORTIE EST IMPRIME.
023C CD01 MOV 13.(T) * L'INDICATEUR DU PROCHAIN CARACTERE A
023D BC20 CPY 12.T * L'IMPRIMANTE EST INITIALISE A LA
023E 1040 PFIN * LIMITE INFERIEURE DANS LE REGISTRE SF13.
023F 4504 TZ 5.X*04*
0240 1C17 JP NPI
0241 1020 RTN
*

* SOUS-PROGRAMME QUI AFFECTE LE CONTENU DES REGISTRES PF8,9

* AU SOMMET DE LA PILE.

0242 C801 PILE MOV 8.(T) * LE CONTENU DES REGISTRES PF8,9 EST TRANSFERE
0243 1080 SSF * AUX REGISTRES SF3,4.
0244 8320 CPY 3.T *
0245 1040 SPF *
0246 C901 MOV 9.(T) *
0247 1080 SSF *
0248 B420 CPY 4.T *
0249 1102 LT 2 * LE SOMMET DE LA PILE DANS LES REGISTRES
024A 8222 ADD 2.T(M) * SF1,2 EST INCREMENTE DE 2.
024B 8182 CP 1.L(M) *
024C 61F8 CP 1.X*F8* * DEBOREMENT DE PILE?
024D 1C52 JP *+5
024E 1040 -SPF *
024F 1110 LT X*10* * OUI: L'INDICATEUR DE DEBOREMENT DE PILE
0250 C520 LOR 5.T * DANS LE REGISTRE PF5 EST MIS A 1.
0251 1020 RTN *
0252 A213 WMF 2.(N) * LE CONTENU DES REGISTRES SF3,4 EST AFFECTE
0253 C301 MOV 3.(T) * A L'ADRESSE DANS LES REGISTRES SF1,2
0254 A208 WMF* 2.I(N) * (LE SOMMET DE LA PILE).
0255 C401 MOV 4.(T) *
0256 1060 RSP

* SOUS-PROGRAMME QUI AFFECTE LE PROCHAIN CARACTERE A L'ENTREE

* AU VECTEUR DE SORTIE SUR L'IMPRIMANTE.

0257 1080 PRINT SSF
0258 1201 LM X*01* * LE PROCHAIN CARACTERE A L'ENTREE EST LU
0259 AA03 RMF 10.(N) * ET AFFECTE AU REGISTRE PF9.
025A 1040 SPF *
025B B920 CPY 9.T *
025C 11F8 LT X*FB* *
025D 5520 AND 5.T
025E 1C2E JP PCHAR * LE CARACTERE EST IMPRIME.
*

COMMENTS

OPERANDS

LOCN CODE FLAGS LABELS OP *

* SOUS-PROGRAMME QUI PLACE L'ADRESSE DU SOMMET DE LA PILE DANS
* LES REGISTRES PF3,4.

025F 1080 SMPL SSF 1.(T) * LE CONTENU DES REGISTRES SF1,2 EST TRANSFERE
0260 C101 MOV 1.(T) * AUX REGISTRES PF3,4.
0261 1040 SPF

0262 B320 CPY 3.(T) *
0263 1080 SSF 2.(T) *
0264 C201 MOV 2.(T) *
0265 1040 SPF 4.(T) *
0266 B420 CPY 13 * LE REGISTRE PF13 INDIQUE LE NOMBRE D'ELEMENTS
0267 B000 ZOF 13 * DANS LA NOUVELLE PILE.
0268 1020 RTN

*
* SOUS-PROGRAMME QUI AFFECTE LE CONTENU DES REGISTRES PF8,9 A
* L'ADRESSE DANS LES REGISTRES PF6,7.

0269 C602 STORE MOV 6.(M)
026A A713 WMF 7.(N)
026B C801 MOV 8.(T)
026C A7DB WMF# 7.(N) * L'ADRESSE DANS LES REGISTRES PF6,7 N'EST PAS
026D C901 MOV 9.(T) MODIFIEE.
026E 1020 RTN

*
* SOUS-PROGRAMME QUI CALCULE L'ADRESSE ABSOLUE DU VECTEUR PARAMETRE
* INDIQUE PAR LE CODE DANS LE REGISTRE PF9.

026F 2600 VECPP LF 6.X'00' * L'ADRESSE DES VECTEURS PARAMETRES PRESENTS
0270 2720 LF 7.X'20' * EST AFFECTEE AUX REGISTRES PF6,7.
0271 5903 TN 9.X'03' * LE CODE EST 0?
0272 1020 RTN
0273 1114 LT 20 * NCN: L'ADRESSE EST INCREMENTEE DE 20 POUR
0274 B720 ADD 7.(T) * INDiquer LE VECTEUR SUIVANT.
0275 8680 ADD 6.L *
0276 9940 DEC 9 * LE CODE EST DECREMENTE DE 1.
0277 1C71 JP VECPP+2

EJE

EXECUTION DES INSTRUCTIONS JSR ET JSF.

0278 0147	JSB	JE	DEPL	* L'ADRESSE DU SOUS-PROGRAMME EST CALCULEE
0279 C301	MOV	3,(T)		* ET AFFECTEE AUX REGISTRES PF10,11 AINSI
027A B620	CPY	6,T		* QU'AUX REGISTRES PF6,7.
027B BA20	CPY	10,T		
027C C401	MOV	4,(T)		
027D B720	CPY	7,T		
027E B820	CPY	11,T		
027F 5110	TN	1,X'10'		* CODE OPERATION DE L'INSTRUCTION JSF?
0280 1C8B	JP	SBI		* NCN: BRANCHEMENT AU TRAITEMENT DE JSR.

TRAITEMENT DE L'ARGUMENT A RETOURNER.

0281 0147	JE	DEPL	* L'ADRESSE DE L'ARGUMENT A RETOURNER EST
0282 9743	DEC	7,(N)	* CALCULEE ET AFFECTEE A L'ADRESSE QUI
0283 9640	DEC	6	* PRECEDE LE DEBUT DU SOUS-PROGRAMME.
0284 F320	SFR	3	* (LES REGISTRES PF6,7 CONTIENNENT L'ADRESSE
0285 F4A0	SFR	4,L	* DE DEPART DU SOUS-PROGRAMME)
0286 A692	WMF	6,L(M)	
0287 C401	MOV	4,(T)	
0288 9743	DEC	7,(N)	
0289 A612	WMF	6,(M)	
028A C301	MOV	3,(T)	
028B 220F	LF	2,X'0F'	* LE NCMBRE D'ARGUMENTS A PASSER, INDIQUE PAR
028C C101	MOV	1,(T)	* LES 5IEME ET 6IEME BITS DE L'OCTET DU CODE
028D E220	AND	2,T	* OPERATION DANS LE REGISTRE PF1 EST AFFECTE
028E F220	SFR	2	* AU REGISTRE PF2.
028F F220	SFR	2	*

TRAITEMENT DES ARGUMENTS A PASSER.

0290 52FF	ARG	TN	2,X'FF'	* AUTRES ARGUMENTS?
0291 1C9E	JP	SBI		* NON: FIN DU TRAITEMENT DES ARGUMENTS.
0292 0147	JE	DEPL		* L'ADRESSE DE L'ARGUMENT A PASSER EST CALCULEE
0293 C302	MOV	3,(M)		* L'ARGUMENT EST LU ET AFFECTE A L'ADRESSE
0294 A4CB	RMF	4,I(N)		* QUI PRECEDE CELLE CONTENUE DANS LES REGISTRES
0295 9743	DEC	7,(N)		* PF6,7. (LES REGISTRES PF6,7 CONTIENNENT
0296 9640	DEC	6		* L'ADRESSE D'UN DES MOTS QUI PRECEDE LE DEBUT
0297 A692	WMF	6,L(M)		* DU SOUS-PROGRAMME)
0298 C302	MOV	3,(M)		
0299 A403	RMF	4,(N)		
029A 9743	DEC	7,(N)		
029B A612	WMF	6,(M)		
029C 9240	DEC	2		* LE NCMBRE D'ARGUMENTS EST DECREMENTE DE 1.
029D 1C90	JP	ARG		
029E 2203	LF	2,X'03'		* LE NCMBRE DE VARIABLES LOCALES, INDIQUE PAR
029F C101	MOV	1,(T)		* LES 7IEME ET 8IEME BITS DE L'OCTET DU CODE
02A0 E221	AND	2,T(T)		* OPERATION DANS LE REGISTRE PF1 EST AFFECTE
02A1 B120	CPY	1,T		* AU REGISTRE PF2 AINSI QU'AU REGISTRE PF1.

TRAITEMENT DES VARIABLES LOCALES.

02A2 52FF * VAR 2,X'FF' * AUTRES VARIABLES LOCALES?
 02A3 1CA0 TN JP SFIN * NCN: FIN DU TRAITEMENT DES VARIABLES LOCALES.
 02A4 0147 JE DEPL * L'ADRESSE DE LA VARIABLE EST CALCULEE.
 02A5 C302 MOV 3,(M) * LA VARIABLE EST LU ET AFFECTEE AUX REGISTRES
 02A6 A403 RMF 4,(N) * PF8,9.
 02A7 B820 CPY 8,T *
 02A8 A4C3 RMF 4,I(N) *
 02A9 B920 CPY 9,T *
 02AA 0242 JE PILE *
 02AB 9240 DEC 2 * LA VARIABLE EST AFFECTEE AU SOMMET DE LA PILE
 02AC 1CA2 JP VAR * LE NCMERE DE VARIABLES EST DECREMENTE DE 1.

* * FIN DU TRAITEMENT.

02AD B800 SFIN ZOF 8 * LE NCMERE DE VARIABLES DANS LE REGISTRE PF1
 02AE C101 MOV 1,(T) * EST TRANSFERE AUX REGISTRES PF8,9 ET AFFECTE
 02AF B920 CPY 9,T * AU SOMMET DE LA PILE.
 02B0 0242 JE PILE *
 02B1 CE01 MOV 14,(T) * L'ADRESSE DE RETOUR, SOIT L'ADRESSE DANS LE
 02B2 B820 CPY 8,T * CCMPTEUR D'INSTRUCTIONS, (REGS PF14,15).
 02B3 CF01 MOV 15,(T) * EST AFFECTEE AU SOMMET DE LA PILE.
 02B4 B920 CPY 9,T *
 02B5 0242 JE PILE *
 02B6 CA01 MOV 10,(T) * L'ADRESSE DE BRANCHEMENT DANS LES REGISTRES
 02B7 BE20 CPY 14,T * PF10,11, EST AFFECTEE AU COMPTEUR D'INSTRUC-
 02B8 CB01 MOV 11,(T) * TIONS POUR CAUSER UN BRANCHEMENT AU SOUS-
 02B9 BF20 CPY 15,T * PROGRAMME.
 02BA 146D JP FETCH

* * EXECUTION DE L'INSTRUCTION CLR.

02BB 2C00 CLR LF 12,0 * L'INSTRUCTION EST DE TYPE 0.
 02BC 009C JE ADRES * L'ADRESSE DE L'OPERANDE EST CALCULEE.
 02BD B800 ZOF 8 * LE CONTENU DU MOT A L'ADRESSE DE L'OPERANDE
 02BE B900 ZOF 9 * EST MIS A 0.
 02BF 0269 JE STORE *
 02C0 146D JP FETCH

* * EXECUTION DE L'INSTRUCTION IPT.

02C1 0197 IPT JE ENT * L'ENTIER EST AFFECTE AUX REGISTRES PF3,4.
 02C2 2C00 LF 12,0 * L'INSTRUCTION EST DE TYPE 0.
 02C3 002C JE ADRES * L'OPERANDE EST EXTRAIT.
 02C4 C301 MOV 3,(T) * L'ENTIER DANS LES REGISTRES PF3,4 EST
 02C5 BA20 CPY 10,T * TRANSFERE AUX REGISTRES PF10,11, ADDITIONNE
 02C6 CA01 MOV 4,(T) * A L'OPERANDE DANS LES REGISTRES PF8,9 ET LE
 02C7 B820 CPY 11,T * RESULTAT EST AFFECTE AU MOT A L'ADRESSE DE
 02C8 0097 JE ADDER * L'OPERANDE.
 02C9 0269 JE STORE *
 02CA 146D JP FETCH

* * EXECUTION DE L'INSTRUCTION CCA.

COMMENTS

LOCN CODE FLAGS LABELS OP * OPERANDS

* LE CCDE CONDITIONNEL EST INITIALISE A 0.

02CB 11FC CCA LT X'FC' *
 02CC E520 AND 1.(T)
 02CD C101 MOV 2.(T)
 02CE B220 CPY 2.(T)
 02CF 1107 LT X'07' * LE NOMBRE DE CARACTERES DANS LA CHAINE EST
 02D0 E220 AND 2.(T) * INDIQUE PAR LE REGISTRE PF2.
 02D1 42FF TZ 2.X'FF' * LE NOMBRE DE CARACTERES EST 0?
 02D2 1CE0 JP CA1

* TRAITEMENT DE LA COMPARAISON D'UN SEUL CARACTERE.

02D3 1080 SSF X'01' * CUI.
 02D4 1201 LM * LE PROCHAIN CARACTERE A L'ENTREE EST LU.
 02D5 AA03 RMF 10.(N) * ET AFFECTE AU REGISTRE PF9.
 02D6 1040 SPY *
 02D7 B920 CPY 3.T * LE CARACTERE A COMPARER, QUI SUIV L'INSTRUC-
 02D8 CE02 MOV 14.(M) * TION, EST LU ET AFFECTE AU REGISTRE T.
 02D9 AF03 RMF 15.(N) * LE COMPTEUR D'INSTRUCTIONS EST MIS A JOUR.
 02DA 8F40 INC 15 *
 02DB 8E80 ADD 14.L *
 02DC D938 XOR* 9.TC * LES DEUX CARACTERES SONT COMPARES.
 02DD 4004 TZ 0.X'04' * CARACTERES IDENTIQUES.
 02DE 8540 INC 5 * OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
 02DF 146D JP . FETCH DU CODE CONDITIONNEL.

* TRAITEMENT DE LA COMPARAISON D'UN OU PLUSIEURS CARACTERES.

02E0 1080 SSF * L'INDICATEUR DU PROCHAIN CARACTERE A L'ENTREE
 02E1 CA01 MOV 10.(T) * EST AFFECTE AU REGISTRE SF3.
 02E2 B320 CPY 3.T *
 02E3 CB01 MOV 11.(T) * LE COMPTEUR DES CARACTERES A L'ENTREE EST
 02E4 B420 CPY 4.T * AFFECTE AU REGISTRE SF4.
 02E5 1080 SSF *
 02E6 1201 LM X'01' * LE PROCHAIN CARACTERE A L'ENTREE EST LU
 02E7 AA03 RMF 10.(N) * ET AFFECTE AU REGISTRE PF9.
 02E8 1040 SPY *
 02E9 B920 CPY 9.T *
 02EA CE02 MOV 14.(M) * LE CARACTERE A COMPARER, QUI SUIV L'INSTRUC-
 02EB AF03 RMF 15.(N) * TION, EST LU ET AFFECTE AU REGISTRE T.
 02EC 8F40 INC 15 * LE COMPTEUR D'INSTRUCTIONS EST MIS A JOUR.
 02ED 8E80 ADD 14.L *
 02EE D938 XOR* 9.TC * LES CARACTERES SONT COMPARES.
 02EF 5004 TN 0.X'04' * CARACTERES IDENTIQUES?
 02F0 1CF8 JP CA3 * NON: PASSAGE AU PROCHAIN CARACTERE A L'ENTREE.
 02F1 01EF JE LIRE * OUI: LE NOMBRE DE CARACTERES EST DECREMENTE DE 1.
 02F2 9240 DEC 2 * LE NOMBRE DE CARACTERES A COMPARER?
 02F3 42FF TZ 2.X'FF' * AUTRES CARACTERES A COMPARER?
 02F4 1CE5 JP CA2 *
 02F5 020E JE NPRNT * NGN: LES CARACTERES LUS SONT AFFECTES AU
 * VECTEUR DE SORTIE.
 02F6 E540 INC 5 * LE CODE CONDITIONNEL INDIQUE QUE LES CHAINES
 02F7 146D JP FETCH DE CARACTERES SONT IDENTIQUES.
 02F8 9240 DEC 2 * LE RESTANT DES CARACTERES SUIVANT L'INSTRUC-
 02F9 52FF TN 2.X'FF' * TION SONT IGNORES ET LE COMPTEUR D'INSTRUC-
 02FA 1CFE JP *+4 * TION EST MODIFIE POUR INDICER L'ADRESSE DE
 02FB 8F40 INC 15 * LA PRECHAINE INSTRUCTION.
 02FC 8E80 ADD 14.L *

02FD 1CF8 JP CA3 *
 02FE 1080 SSF
 02FF C301 * L'INDICATEUR DU PROCHAIN CARACTERE A L'ENTREE
 0300 BA20 * EST RETABLI DANS LE REGISTRE SF10.
 0301 C401 * LE COMPTEUR DES CARACTERES A L'ENTREE EST
 0302 B820 * RETABLI DANS LE REGISTRE SF11.
 0303 1040 SPF
 0304 146D JP FETCH

* * * EXECUTION DE L'INSTRUCTION BRA.

0305 C501 * L'OPERATION OU EXCLUSIF EST EFFECTUE ENTRE
 0306 D129 XOR* 5.(T) * LE CODE CONDITIONNEL ET LE CODE DE LA
 0307 B220 CPY 2.(T) * CONDITION DANS L'INSTRUCTION ET LE RESULTAT
 * EST AFFECTE AU REGISTRE PF2.
 0308 4203 TZ 2.X*03* * CODES IDENTIFIQUES?
 0309 100D JP BR1 * * * NCN: LES CODES SONT DIFFERENTS.
 030A 4104 TZ 1.X*04* * CONDITION EGAL, PLUS PETIT OU PLUS GRAND?
 030B 100F JP BR1+2 *
 030C 10E5 JP JMP * OUI: LE BRANCHEMENT EST EFFECTUE.
 030D 4104 TZ 1.X*04* * CONDITION EGAL, PLUS PETIT OU PLUS GRAND?
 030E 10E5 JP JMP * *
 030F 1102 LT 2 * OUI: LE BRANCHEMENT N'EST PAS EFFECTUE ET
 0310 8F20 ADD 15.T * LE COMPTEUR D'INSTRUCTIONS EST INCREMENTE
 0311 8E80 ADD 14.L * DE 2 POUR INDICER L'INSTRUCTION SUIVANTE.
 0312 146D JP FETCH

* * * EXECUTION DE L'INSTRUCTION PSH.

0313 CE02 PSH MOV 14.(M) * L'OCTET SUIVANT CELUI DU CODE OPERATION EST
 0314 AF03 RMF 15.(N) * LU ET AFFECTE AU REGISTRE PF2.
 0315 B220 CPY 2.T *
 0316 42F0 TZ 2.X*F0* * LA PREMIERE ADRESSE SUIVANT L'INSTRUCTION EST
 0317 101C JP PSI DE MODE IMMEDIAT?
 0318 2C01 LF 12.1 * CUI: L'INSTRUCTION EST DE TYPE 1.
 0319 009C JE ADRES * L'OPERANDE EST EXTRAIT.
 031A 0242 JE PILE * LE MOT A L'ADRESSE SUIVANT L'INSTRUCTION EST
 031B 146D JP FETCH AFFECTE AU SOMMET DE LA PILE DU SYSTEME.
 031C 2C02 LF 12.2 * NCN: L'INSTRUCTION EST DE TYPE 2.
 031D 009C JE ADRES * LES OPERANDES SONT EXTRAITS.
 031E F900 SFL 9 * L'ADRESSE DU SOMMET DE LA PILE EST CALCULEE.
 031F F880 SFL 8.L *
 0320 1102 LT 2 * LE SOMMET DE LA PILE EST INCREMENTE DE 2.
 0321 8922 ADD 9.T(M) *
 0322 8882 ADD 8.L(M) *
 0323 A913 WMF 9.(N) * LE MOT A LA DEUXIEME ADRESSE SUIVANT
 0324 CA01 MOV 10.(T) * L'INSTRUCTION EST AFFECTE AU SOMMET DE LA
 0325 A0B WMF* 9.I(N) * PILE (SOIT A L'ADRESSE DANS LES REGISTRES
 0326 CB01 MOV 11.(T) * PF8,9).
 0327 F820 SFR 8 * LA NOUVELLE ADRESSE DU SOMMET DE LA PILE
 0328 F9A0 SFR 9.L * EST AFFECTE A LA PREMIERE ADRESSE SUIVANT
 0329 0269 JE STORE * L'INSTRUCTION.
 032A 146D JP FETCH

* EXECUTION DE L'INSTRUCTION TST.

0326 2C00 TST 12,0 * L'INSTRUCTION EST DE TYPE 0.
 032C 009C JE ADRES * L'OPERANDE EST EXTRAIT.
 032D 11FC LT X'FC' * LE CODE CONDITIONNEL EST INITIALISE A 0.
 032E E520 AND 5,T *
 032F 5880 TN 8,X'80' * LE BIT DE GAUCHE EST 0?
 0330 8540 INC 5 * OUI: L'INDICATEUR ZERO DU C.C. EST MIS A 1.
 0331 146D JP FETCH

* EXECUTION DE L'INSTRUCTION CRL.

0332 2C02 CRL 12,2 * L'INSTRUCTION EST DE TYPE 2.
 0333 009C JE ADRES * LES OPERANDES SONT EXTRAITS.
 0334 4A5F TZ 10,X'5F' * L'OPERANDE EST UN PARAMETRE?
 0335 1D3C JP CRI *
 0336 5A20 TN 10,X'20' *
 0337 1D3C JP CRI *
 0338 2801 LF 8,X'01' * OUI: L'INDICATEUR '0100' EST AFFECTE A
 0339 B900 ZOF 9 * L'ADRESSE DU PREMIER OPERANDE.
 033A 0269 JE STORE *
 033B 146D JP FETCH
 033C B800 ZOF 8 * NON: LE 1ER OCTET DE L'INDICATEUR EST MIS A 0.
 033D C801 MOV 11,(T) * LES 3 DERNIERS BITS DU CODE DU DEUXIEME
 033E B920 CPY 9,T * CARACTERE DE L'ETIQUETTE SONT AFFECTES AU
 033F 1107 LY X'07' * REGISTRE PF9, SOIT LE DEUXIEME OCTET DE
 0340 E920 AND 9,T * L'INDICATEUR.
 0341 FA00 SFL 10 * LES 5 DERNIERS BITS DU CODE DU PREMIER
 0342 FA00 SFL 10 * CARACTERE DE L'ETIQUETTE SONT AFFECTES
 0343 FA00 SFL 10 * AUX 5 PREMIERS BITS DU REGISTRE PF9 POUR
 0344 CA01 MOV 10,(T) * FORMER UN INDICATEUR DE 8 BITS DANS LE
 0345 C920 LOR 9,T * REGISTRE PF9.
 0346 0269 JE STORE * L'INDICATEUR EST AFFECTE A L'ADRESSE DU
 0347 146D JP FETCH PREMIER OPERANDE.

* EXECUTION DE L'INSTRUCTION POP.

0348 CE02 POP MOV 14,(M) * L'OCTET SUIVANT CELUI DU CODE OPERATION
 0349 AF03 RMF 15,(N) * EST LU ET AFFECTE AU REGISTRE PF2.
 034A B220 CPY 2,T *
 034B 42FF TZ 2,X'FF' * LES DEUX ADRESSES SUIVANT L'INSTRUCTION
 034C 1D50 JP PO1 * SONT DE MODE IMMEDIAT?
 034D 8F40 INC 15 * OUI: LE COMPTEUR D'INSTRUCTIONS EST
 034E 8880 ADD 14,L * INCREMENTE DE 1.
 034F 1469 JP RSP * LA PILE DU SYSTEME EST REINITIALISEE.
 0350 42F0 TZ 2,X'F0' * LA PREMIERE ADRESSE SUIVANT L'INSTRUCTION
 0351 1D57 JP PO2 * EST DE MODE IMMEDIAT?
 0352 2C01 LF 12,1 * OUI: L'INSTRUCTION EST DE TYPE 1.
 0353 009C JE ADRES * L'OPERANDE EST EXTRAIT.
 0354 0154 JE DPILE * LE MOT AU SOMMET DE LA PILE DU SYSTEME
 0355 0269 JE STORE * EST EXTRAIT ET AFFECTE A L'ADRESSE
 0356 146D JP FETCH SUIVANT L'INSTRUCTION.
 0357 8F40 INC 15 * NON: LE COMPTEUR D'INSTRUCTIONS EST
 0358 8E80 ADD 14,L * INCREMENTE DE 1.
 0359 C201 MOV 2,(T) * LES MODES D'ADRESSAGE DANS LE REGISTRE

035A B120 CPY 1,T * PF2 SCNT AFFECTES AU REGISTRE PFI.
 035B 7120 SRF 1 * LE CCONTENU DU REGISTRE PFI EST DEPLAC
 035C 2C00 LF 12,0 DE 4 BITS VERS LA DROITE.
 035D 009C JE ADRES * LE PREMIER OPERANDE EST EXTRAIT.
 035E F900 SFL 3 * L'ADRESSE DU SOMMET DE LA PILE EST CALCULEE.
 035F F880 SFL 3,L *
 0360 C802 MOV 3,(M) * LE MOY AU SOMMET DE LA PILE EST LU ET
 0361 A903 RMF 9,(N) * AFPECTE AU REGISTRES PF10,11.
 0362 BA20 CPY 13,T *
 0363 A9CB RMF* 9,I(N) *
 0364 BB20 CPY 11,T *
 0365 1102 LT 2 * L'ADRESSE DU SOMMET DE LA PILE EST
 0366 9920 SBT 9,T * DECREMENTEE DE 2 ET AFFECTEE A LA PREMIERE
 0367 9880 SBT 3,L * ADRESSE SUIVANT L'INSTRUCTION.
 0368 F820 SFR 8 *
 0369 F9A0 SFR 9,L *
 036A 0269 JE STORE *
 036B C201 MOV 2,(T) *
 036C B120 CPY 1,T * LES MODES D'ADRESSAGE DANS LE REGISTRE PF2
 036D 2C00 LF 12,0 * SONT TRANSFERES AU REGISTRE PFI.
 036E 009C JE ADRES * LE SECCND OPERANDE SUIVANT L'INSTRUCTION
 036F CA01 MOV 10,(T) * EST EXTRAIT.
 0370 BB20 CPY 8,T * LE CCONTENU DU SOMMET DE LA PILE DANS L'ES
 0371 CB01 MOV 11,(T) * REGISTRES PF10,11 EST TRANSFERE AUX
 0372 B920 CPY 9,T * REGISTRES PF8,9 ET AFFECTE A LA SECONDE
 0373 0269 JE STORE * ADRESSE SUIVANT L'INSTRUCTION.
 0374 146D JP FETCH *

*
 *
 * EXECUTION DE L'INSTRUCTION CCC.

0375 11FC CCC LT X'FC' * LE CODE CONDITIONNEL EST INITIALISE A 0.
 0376 E520 AND 5,T *
 0377 C101 MOV 1,(T) * LE CODE OPERATION EST AFFECTE AU REGISTRE
 0378 B220 CPY 2,T * PF2.
 0379 1080 SSF *
 037A 1201 LM X'01' * LE PROCHAIN CARACTERE A L'ENTREE EST LU ET
 037B AA03 RMF 10,(N) * AFFECTE AU REGISTRE PF9.
 037C 1040 SPF *
 037D B920 CPY 9,T *
 037E 4207 TZ 2,X'07' * CODE CARACTERE NUL?
 037F 1D83 JP CHIF * NGN.

*
 *
 * TRAITEMENT DU CODE CARACTERE NUL.

0380 59FF TN 9,X'FF' * LE CARACTERE A L'ENTREE EST LE CARACTERE NUL?
 0381 8540 INC 5 * OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
 0382 146D JP FETCH DU CODE CONDITIONNEL.
 0383 4206 TZ 2,X'06' * CODE CHIFFRE?
 0384 1D8A JP LETT * NON.

*
 *
 * TRAITEMENT DU CODE CHIFFRE.

0385 69D0 CP 9,X'00' * LE CARACTERE A L'ENTREE EST UN CHIFFRE
 0386 146D JP FETCH * ENTRE 0 ET 9?
 0387 69C6 CP 9,X'C6' *
 0388 8540 INC 5 * OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO

0389 146D	JP	FETCH	DU CODE CONDITIONNEL.
038A 4205	TZ	2.X*05	* CCDE LETTRE?
038B 1D91	JP	CHLT	* NCN.
* * TRAITEMENT DU CODE LETTRE.			
038C 698F	CP	9.X*BF	* LE CARACTERE A L'ENTREE EST UNE LETTRE
038D 146D	JP	FETCH	* ENTRE A ET Z?
038E 69A5	CP	9.X*AS	* OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
038F 8540	INC	5	DU CODE CONDITIONNEL.
0390 146D	JP	FETCH	* CCDE CHIFFRE OU LETTRE?
0391 4204	TZ	2.X*04	* NCN.
0392 1D9C	JP	SPEC	
* * TRAITEMENT DU CODE CHIFFRE OU LETTRE.			
0393 69D0	CP	9.X*00	* LE CARACTERE A L'ENTREE EST UN CHIFFRE
0394 146D	JP	FETCH	* ENTRE 0 ET 9?
0395 69C6	CP	9.X*C6	* OUI.
0396 1D9A	JP	*+4	* LE CARACTERE A L'ENTREE EST UNE LETTRE
0397 69BF	CP	9.X*BF	* ENTRE A ET Z?
0398 146D	JP	FETCH	
0399 69A5	CP	9.X*AS	* OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
039A 8540	INC	5	DU CODE CONDITIONNEL.
039B 146D	JP	FETCH	* CODE CARACTERE SPECIAL?
039C 4203	TZ	2.X*03	* NCN.
039D 1DAA	JP	PRMTR	
* * TRAITEMENT DU CODE CARACTERE SPECIAL.			
039E 69DF	CP	9.X*DF	* LE CARACTERE A L'ENTREE POSSEDE UN CODE
039F 146D	JP	FETCH	* ASCII ENTRE 21 ET 2F?
03A0 69D0	CP	9.X*00	* OUI.
03A1 1DA8	JP	*+7	* LE CARACTERE A L'ENTREE POSSEDE UN CODE
03A2 69C6	CP	9.X*C6	* ASCII ENTRE 3A ET 40?
03A3 146D	JP	FETCH	
03A4 69BF	CP	9.X*BF	* OUI.
03A5 1DA8	JP	*+3	* LE CARACTERE A L'ENTREE POSSEDE UN CODE
03A6 69A5	CP	9.X*AS	ASCII ENTRE 5B ET 5F?
03A7 146D	JP	FETCH	* OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
03A8 8540	INC	5	DU CODE CONDITIONNEL.
03A9 146D	JP	FETCH	
* * TRAITEMENT DE LA COMPARAISON DES PARAMETRES.			
03AA 1080	PRMTR	SSF	
03AB CA01	MOV	10.(T)	* L'INDICATEUR DU PROCHAIN CARACTERE A L'ENTREE
03AC B320	CPY	3.T	* EST AFFECTE AU REGISTRE SF3.
03AD CB01	MOV	11.(T)	* LE COMPTEUR DES CARACTERES A L'ENTREE EST
03AE B420	CPY	4.T	* AFFECTE AU REGISTRE SF4.
03AF 0TEF	JE	LIRE	* PASSAGE AU PROCHAIN CARACTERE A L'ENTREE.
03B0 1080	SSF		
03B1 292A	LF	8.X*2A	* LE CODE DE ** EST AFFECTE AU REGISTRE SF8.
03B2 1201	LM	X*01	* LE PROCHAIN CARACTERE A L'ENTREE EST LU ET
03B3 AA03	RMF	10.(N)	* AFFECTE AU REGISTRE T.
03B4 D838	XDR*	8.TC	* LE PROCHAIN CARACTERE A L'ENTREE EST LE
03B5 5004	TN	0.X*04	* CARACTERE **?

COMMENTS

UPPERANDS

LUCN CODE FLAGS LABELS CP *

03B6 1DCA	JP	CFIN	* NCN: IL NE SAGIT PAS D'UN PARAMETRE.
03B7 1040	SPF		
03B8 4202	TZ	2,X'02'	* CCDE PARAMETRE ETIQUETTE?
03B9 10BF	JP	LIS	* NCN.
* TRAITEMENT DU CODE PARAMETRE ETIQUETTE.			
03BA 69BF	CP	9,X'BF'	* LE CARACTERE A L'ENTREE EST UNE LETTRE
03BB 1DCA	JP	CFIN	* ENTRE A ET F?
03BC 69B9	CP	9,X'B9'	
03BD 2540	INC	5	* OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
03BE 1DCA	JP	CFIN	DU CODE CONDITIONNEL.
03BF 4201	TZ	2,X'01'	* CCDE PARAMETRE LISTE?
03C0 1DC6	JP	ARB	* NCN.
* TRAITEMENT DU CODE PARAMETRE LISTE.			
03C1 69AB	CP	9,X'AB'	* LE CARACTERE A L'ENTREE EST UNE LETTRE
03C2 1DCA	JP	CFIN	* ENTRE U ET Z?
03C3 69A5	CP	9,X'A5'	
03C4 2540	INC	5	* OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
03C5 1DCA	JP	CFIN	DU CODE CONDITIONNEL.
* TRAITEMENT DU CODE PARAMETRE ARBRE.			
03C6 69B7	CP	9,X'B7'	* LE CARACTERE A L'ENTREE EST UNE LETTRE
03C7 1DCA	JP	CFIN	* ENTRE I ET L?
03C8 69B3	CP	9,X'B3'	
03C9 2540	INC	5	* OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
* DU CODE CONDITIONNEL.			
03CA 1080	SSF		
03CB C301	MOV	3,(T)	* L'INDICATEUR DU PROCHAIN CARACTERE A L'ENTREE
03CC BA20	CPY	10,T	* EST RETABLI DANS LE REGISTRE SF10.
03CD C401	MOV	4,(T)	* LE COMPTEUR DES CARACTERES A L'ENTREE EST
03CE BB20	CPY	11,T	* RETABLI DANS LE REGISTRE SF11.
03CF 1040	SPF		
03D0 146D	JP	FETCH	
* EXECUTION DE L'INSTRUCTION BDY.			
03D1 2C00	BDX	LF 12,0	* L'INSTRUCTION EST DE TYPE 0.
03D2 009C	JE	ADRES	* L'INDEX EST EXTRAIT ET DECREMENTE DE 1.
03D3 9940	DEC	9	
03D4 9880	SBT	8,L	
03D5 4880	TZ	8,X'80'	* RESULTAT NEGATIF?
03D6 1DD9	JP	BD1	
03D7 0269	JE	STORE	* NON: LE RESULTAT EST AFFECTE A L'ADRESSE DE
03D8 1DES	JP	JMP	* L'OPERANDE ET LE BRANCHEMENT EST EFFECTUE.
03D9 1102	LT	2	* OUI: LE BRANCHEMENT N'EST PAS EFFECTUE ET LE
03DA 8F20	ADD	15,T	* COMPTEUR D'INSTRUCTIONS EST INCREMENTE DE 2
03DB 8E80	ADD	14,L	* POUR INDICHER L'ADRESSE DE L'INSTRUCTION
03DC 146D	JP	FETCH	SUIVANTE.
* EXECUTION DE L'INSTRUCTION MOV.			

03DD 2C02 LF 12,2 * L'INSTRUCTION EST DE TYPE 2.
 03DE 009C JE ADRES * LES OPERANDES SONT EXTRAITS
 03DF CA01 MOV 10,(T) * LE DEUXIEME OPERANDE DANS LE REGISTRE PF10,11
 03E0 B820 CPY 8,T * EST TRANSFERE AUX REGISTRES PF8,9 ET
 03E1 CB01 MOV 11,(T) * AFFECTE A L'ADRESSE DU PREMIER OPERANDE.
 03E2 B920 CPY 9,T *
 03E3 0269 JE STORE *
 03E4 1460 JP FETCH *

*
 *
 * EXECUTION DE L'INSTRUCTION JMP.

03E5 0147 JMP DEPL * L'ADRESSE DE BRANCHEMENT EST CALCULEE ET
 03E6 C301 MOV 3,(T) * AFFECTEE AU COMPTEUR D'INSTRUCTIONS DANS
 03E7 BE20 CPY 14,T * LES REGISTRES PF14,15.
 03E8 C401 MOV 4,(T) *
 03E9 BF20 CPY 15,T *
 03EA 1460 JP FETCH *
 0400 ORG X'400'

*
 *
 * EXECUTION DE L'INSTRUCTION BIN.

0400 2C00 BIN LF 12,0 * L'INSTRUCTION EST DE TYPE 0.
 0401 009C JE ADRES * L'ADRESSE DE L'OPERANDE EST EXTRAITE.
 0402 11FC LT X'FC' * LE CODE CONDITIONNEL EST INITIALISE A 0.
 0403 E520 AND 5,T *
 0404 1080 SSF *
 0405 1201 LM X'01' * LE CHIFFRE A L'ENTREE EST LU ET AFFECTE
 0406 AA03 RMF 10,(N) * AU REGISTRE SF6.
 0407 B620 CPY 6,T *
 0408 B300 ZOF 3 * LE CONTENU DES REGISTRES SF3,4 EST REMIS A 0.
 0409 B400 ZOF 4 *
 040A 2504 LF 5,4 *
 040B F400 BNT SFL 4 * LE COMPTEUR EST INITIALISE A 4.
 040C F380 SFL 3,L * LA SUITE DE CHIFFRES A L'ENTREE SERA AFFECTEE
 040D F400 SFL 4 * AUX REGISTRES SF3,4.
 040E F380 SFL 3,L *
 040F F400 SFL 4 *
 0410 F380 SFL 3,L *
 0411 F400 SFL 4 *
 0412 F380 SFL 3,L *
 0413 110F LT X'0F' *
 0414 E621 AND 6,T(T) *
 0415 C420 LOR 4,T *
 0416 5540 DEC 5 *
 0417 01EF JE LIRE * LE COMPTEUR EST DECREMENTE DE 1.
 0418 0257 JE PRINT * PASSAGE AU PROCHAIN CARACTERE A L'ENTREE.
 0419 1080 SFL * LE CARACTERE EST AFFECTE A L'IMPRIMANTE.
 041A 1201 LM X'01' * LE CARACTERE A L'ENTREE EST LU ET AFFECTE
 041B AA03 RMF 10,(N) * AU REGISTRE SF6.
 041C B620 CPY 6,T *
 041D 6D00 CP 6,X'D0' * LE CARACTERE EST UN CHIFFRE ENTRE 0 ET 9?
 041E 1427 JP UNIT *
 041F 66C6 CP 6,X'C6' *
 0420 1422 JP *+2 * CUI.
 0421 1427 JP UNIT * NCN.

COMMENTS

LOCN CODE FLAGS LABELS OP * OPERANDS

LOCN	CODE	FLAGS	LABELS	OP	* OPERANDS	COMMENTS
0422	45FF			TZ	5,X'FF'	* LE CCMPTEUR EST 0?
0423	140B			JP	BN1	
0424	1040			SPF		* OUI: LE NOMBRE A L'ENTREE DEPASSE 9999 ET
0425	8540			INC	5	* DONC LE BIT ZERO DU CODE CONDITIONNEL EST
0426	0060			JE	FETCH	MIS A 1.
0427	8700		UNIT	ZOF	7	* LE CHIFFRE REPRESENTANT LES UNITES EST
0428	C401			MOV	4,(T)	* AFFECTE AUX REGISTRES SF7,8.
0429	B820			CPY	8,T	
042A	110F			LT	X'0F'	
042B	E820			AND	8,T	
042C	7420			SRF	4	* LE CHIFFRE REPRESENTANT LES DIZAINES EST
042D	E420			AND	4,T	* AFFECTE AU REGISTRE SF4.
042E	54FF		DIZ	TN	4,X'FF'	* LE CHIFFRE REPRESENTANT LES DIZAINES EST 0?
042F	1439			JP	CENT+4	* CUI.
0430	9440			DEC	4	* NCN: CE CHIFFRE EST DECREMENTE DE 1.
0431	110A			LT	X'0A'	* LE RESULTAT DANS LES REGISTRES SF7,8 EST
0432	8820			ADD	8,T	* INCREMENTE DE 10.
0433	8780			ADD	7,L	
0434	142E			JP	DIZ	
0435	C301		GENT	MOV	3,(T)	* LE CHIFFRE REPRESENTANT LES CENTAINES EST
0436	B420			CPY	4,T	* AFFECTE AU REGISTRE SF4.
0437	110F			LT	X'0F'	
0438	E420			AND	4,T	* LE CHIFFRE REPRESENTANT LES CENTAINES EST 0?
0439	54FF			TN	4,X'FF'	* OUI.
043A	1440			JP	MILL	* NCN: CE CHIFFRE EST DECREMENTE DE 1.
043B	9440			DEC	4	* LE RESULTAT DANS LES REGISTRES SF7,8 EST
043C	1164			LT	X'64'	* INCREMENTE DE 100.
043D	8820			ADD	8,T	
043E	8780			ADD	7,L	
043F	1438		MILL	JP	CENT+3	
0440	7320			SRF	3	* LE CHIFFRE REPRESENTANT LES MILLIERS EST
0441	110F			LT	X'0F'	* AFFECTE AU REGISTRE SF3.
0442	E320			AND	3,T	
0443	53FF			JP	3,X'FF'	* LE CHIFFRE REPRESENTANT LES MILLIERS EST 0?
0444	144B			TN	BN2	* OUI.
0445	9340			DEC	3	* NCN: CE CHIFFRE EST DECREMENTE DE 1.
0446	11E8			LT	X'E8'	* LE RESULTAT DANS LES REGISTRES SF7,8 EST
0447	8820			ADD	8,T	* INCREMENTE DE 1000.
0448	1103			LT	X'03'	
0449	87A0			ADD	7,LT	
044A	144Q			JP	MILL	
044B	C701		BN2	MOV	7,(T)	* LE NOMBRE RESULTANT DANS LES REGISTRES SF7,8
044C	1040			SPF		* EST TRANSFERE AUX REGISTRES PF8,9 ET
044D	B820			CPY	8,T	* AFFECTE A L'ADRESSE DE L'OPERANDE.
044E	1080			SSF		
044F	C801			MOV	8,(T)	
0450	1040			SPF		
0451	0920			CPY	9,T	
0452	0269			JE	STORE	
0453	006D			JE	FETCH	

* * EXECUTION DES INSTRUCTIONS ADD ET SUB.

* * * L'INSTRUCTION EST DE TYPE 2.

* * * LES OPERANDES SONT EXTRAITS.

* * * INSTRUCTION SUB?

ADS

LF

12,2

ADRES

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

* * *

COMMENTS

LOCN CODE FLAGS LABELS OP * OPERANDS

0483 5002 TN 0.2 * RESULTAT POSITIF?
0484 3502 AF 5.2 * CUI: LE PREMIER OPERANDE EST PLUS PETIT.

0485 006D JE FETCH
0486 4002 TZ 0.2 * RESULTAT NEGATIF?
0487 3502 AF 5.2 * CUI: LE PREMIER OPERANDE EST PLUS PETIT.
0488 4004 TZ 0.4 * RESULTAT ZERO?
0489 6540 INC 5 * CUI: LES 2 OPERANDES SONT IDENTIQUES.
048A 006D JE FETCH

* *
* * EXECUTION DE L'INSTRUCTION DNS.

048B 2C02 LF 12.2 * L'INSTRUCTION EST DE TYPE 2.
048C 009C JE ADRES * LES OPERANDES SONT EXTRAITS.
048D 025F JE SMPL * LA NOUVELLE PILE EST INITIALISEE.
048E CA01 MOV 10.(T) * L'INDICATEUR DANS LES REGISTRES PF10.11
048F B620 CPY 6.T * (LE SECOND OPERANDE) EST TRANSFERE
0490 BC20 CPY 12.T * AUX REGISTRES PF6.7 AINSI QU'AUX REGISTRES
0491 CB01 MOV 11.(T) * PF12.SF9.
0492 B720 CPY 7.T *
0493 1080 SSF *
0494 B920 CPY 9.T *
0495 1040 SPF *
0496 487F TZ 8.X.7F. * L'INDICATEUR DANS LA 1ERE LISTE EST NUL?
0497 149B JP *+4 *
0498 49FF TZ 9.X.FF. *
0499 149B JP *+2 *

049A 1486 JP DN7 *
049B CC01 MOV 12.(T) * NON: L'INDICATEUR DU DEBUT DE LA SECONDE
049C BA20 CPY 10.T * LISTE DANS LES REGISTRES PF12.SF9 EST
049D 1080 SSF * AFFECTE AUX REGISTRES PF10.11.
049E C901 MOV 9.(T) *
049F 1040 SPF *
04A0 BB20 CPY 11.T * L'INDICATEUR DANS LA 2E LISTE EST NUL?
04A1 4A7F TZ 10.X.7F. *
04A2 14A6 JP *+4 *
04A3 4BFF TZ 11.X.FF. *
04A4 14A6 JP *+2 *

04A5 14B1 JP DN5 * NON: LES ETIQUETTES SONT COMPAREES.
04A6 04F1 JE E.1EG * ETIQUETTES IDENTIQUES?
04A7 5501 TN 5.X.01. *
04A8 14AF JP DN4 *
04A9 0185 JE EPL * CUI: LES INDICATEURS SONT AFFECTES A LA PILE.
04AA 01A0 JE FILS * PASSAGE AUX FILS AINES.
04AB 01BE JE F3LS *

04AC 1496 JP DN1 *
04AD 4504 TZ 5.X.04. * LA RELATION -DANS- EST SATISFAITE?
04AE 14B4 JP DN6 *
04AF 01E0 JE F2ERE * NON: PASSAGE AU FRERE SUIVANT DANS LA
04B0 14A1 JP DN2 * DEUXIEME LISTE.
04B1 11FB LT X.FB. * LA RELATION -DANS- N'EST PAS ENCORE
04B2 E520 AND 5.T * SATISFAITE.
04B3 148B JP DN8 *
04B4 01D1 JE FIERE * PASSAGE AU FRERE SUIVANT DANS LA 1ERE LISTE.
04B5 1496 JP DN1 *
04B6 1104 LT X.04. * LA RELATION -DANS- EST SATISFAITE JUSQU'A
04B7 C520 LOR 5.T * DATE.

COMMENTS

LOCN CODE FLAGS LABELS OP * OPERANDS

04B8 5DFF TN 13.X'FF' * LA PILE EST VIDE?
 04B9 0551 JE COND * CUI: FIN DU TRAITEMENT.
 04BA 016E JE DPL * NCN: LES INDICATEURS SONT RETIRES DU SOMMET
 04BB 14AD JP DN3 * DE LA PILE.

*

* EXECUTION DE L'INSTRUCTION HRS.

04BC 2C0C HRS 12.2 * L'INSTRUCTION EST DE TYPE 2.
 04BD 009C JE ADRES * LES OPERANDES SONT EXTRAITS.
 04BE 025F JE SMPL * LA NOUVELLE PILE EST INITIALISEE.
 04BF 11FB LT X'FB' * LA RELATION -HORS- N'EST PAS ENCORE
 04C0 E520 AND 5.T * Satisfait.
 04C1 4A7F TZ 10.X'7F' * L'INDICATEUR DU DEBUT DE LA SECONDE LISTE
 04C2 14C6 JP *+4 * EST NUL?
 04C3 4BFF TZ 11.X'FF' *
 04C4 14C6 JP *+2 *
 04C5 0551 JE COND * CUI: FIN DU TRAITEMENT.
 04C6 CA01 MOV 10.(T) * NCN: L'INDICATEUR DANS LES REGISTRES PF10.11
 04C7 B620 CPY 6.T * EST TRANSFERE AUX REGISTRES PF6.7.
 04C8 CB01 MOV 11.(T) *
 04C9 B720 CPY 7.T *
 04CA 11FB LT X'FB' * LA RELATION -HORS- N'EST PAS ENCORE
 04CB E520 AND 5.T * Satisfait.
 04CC 487F TZ 8.X'7F' * L'INDICATEUR DANS LA 1ERE LISTE EST NUL?
 04CD 14D1 JP *+4 *
 04CE 49FF TZ 9.X'FF' *
 04CF 14D1 JP *+2 *
 04D0 0551 JE COND * CUI: FIN DU TRAITEMENT.
 04D1 C601 MOV 6.(T) * L'INDICATEUR DANS LES REGISTRES PF6.7
 04D2 BA20 CPY 10.T * EST TRANSFERE AUX REGISTRES PF10.11.
 04D3 C701 MOV 7.(T) *
 04D4 B820 CPY 11.T *
 04D5 04F1 JE ETIEG * LES ETIQUETTES SONT COMPAREES.
 04D6 4501 TZ 5.X'01' * ETIQUETTES IDENTIQUES?
 04D7 14E3 JP HRS * NCN: PASSAGE AU FRERE SUIVANT DANS LA
 04D8 01E0 JE F2ERE * DEUXIEME LISTE.
 04D9 4A7F TZ 10.X'7F' * L'INDICATEUR DANS LA 2E LISTE EST NUL?
 04DA 14D5 JP HR3 *
 04DB 4BFF TZ 11.X'FF' *
 04DC 14D5 JP HR3 *
 04DD 01D1 JE FIERE * CUI: PASSAGE AU FRERE SUIVANT DANS LA
 04DE 487F TZ 8.X'7F' * PREMIERE LISTE.
 04DF 14D1 JP HR2 * L'INDICATEUR DANS LA 1ERE LISTE EST NUL?
 04E0 49FF TZ 9.X'FF' *
 04E1 14D1 JP HR2 *
 04E2 14E7 JP HR6 * CUI.
 04E3 0185 JE EPL * LES INDICATEURS SONT AFFECTES A LA PILE.
 04E4 01A0 JE FILS * PASSAGE AUX FILS AINES.
 04E5 01BE JE F3LS *
 04E6 14CA JP HR1 *
 04E7 1104 LT X'04' * LA RELATION -HORS- EST VERIFIEE JUSQU'A
 04E8 C520 LOR 5.T * DATE.
 04E9 5DFF TN 13.X'FF' * LA PILE EST VIDE?
 04EA 0551 JE COND * CUI: FIN DU TRAITEMENT.

04EB 016E JE DPL * NCN: LES INDICATEURS SONT RETIRES DU
04EC 14DB JP HR4 * SOMMET DE LA PILE.

* * * EXECUTION DE L'INSTRUCTION CLB.

04ED 2C02 LF 12.2 * L'INSTRUCTION EST DE TYPE 2.
04EE 009C JE ADRES * LES OPERANDES SONT EXTRAITS.
04EF 04F1 JE ETIEG * LES ETIQUETTES SONT CMPAREES.
04FO 006D JE FETCH

* * * SOUS-PROGRAMME QUI VERIFIE L'EQUALITE ENTRE DEUX ETIQUETTES ET
* * * QUI MODIFIE LE CODE CONDITIONNEL SELON LE RESULTAT. SI LES
* * * ETIQUETTES SONT IDENTIQUES, L'INDICATEUR ZERO DU CODE
* * * CONDITIONNEL DANS LE REGISTRE PF5 EST MIS A 1. AUTREMENT IL
* * * EST REMIS A 0.

04F1 11FC LT X'FC' * LE CODE CONDITIONNEL EST INITIALISE A 0.
04F2 E520 AND 5.T * L'ADRESSE ABSOLUE DE LA PREMIERE ETIQUETTE
04F3 1180 LT X'80' * EST CALCULEE A PARTIR DE L'INDICATEUR
04F4 8929 ADD* 9.T(T) * DANS LES REGISTRES PF8,9 ET AFFECTEE AUX
04F5 1080 SSF * REGISTRES SF3,4.
04F6 B420 CPY 4.T *
04F7 1040 SPF *
04F8 1101 LT X'01' *
04F9 88A9 ADD* 8.LT(T) *
04FA 1080 SSF *
04FB B320 CPY 3.T *
04FC F400 SFL 4 *
04FD F380 SFL 3.L *
04FE 1040 LT X'80' * L'ADRESSE ABSOLUE DE LA DEUXIEME ETIQUETTE
04FF 1180 LT 11.T(T) * EST CALCULEE A PARTIR DE L'INDICATEUR
0500 8B29 ADD* * DANS LES REGISTRES PF10,11 ET AFFECTEE
0501 1080 SSF *
0502 B620 CPY 6.T * AUX REGISTRES SF5,6.
0503 1040 SPF *
0504 1101 LT X'01' *
0505 8AA9 ADD* 10.LT(T) *
0506 1080 SSF *
0507 B520 CPY 5.T *
0508 F600 SFL 6 *
0509 F580 SFL 5.L *
050A C302 MOV 3.(M) * LE PREMIER CARACTERE DE LA PREMIERE
050B A403 RME 4.(N) * ETIQUETTE EST LU ET AFFECTE AU REGISTRE
050C 8440 INC 4 * SF7.
050D B720 CPY 7.T *
050E C502 MOV 5.(M) * LE PREMIER CARACTERE DE LA SECONDE
050F A603 RME 6.(N) * ETIQUETTE EST LU ET AFFECTE AU REGISTRE
0510 8640 INC 6 * SF8.
0511 B820 CPY 8.T *
0512 9738 SBT* 7.TC * LES CARACTERES SONT CMPARES.
0513 5004 TN 0.X'04' * CARACTERES IDENTIQUES?
0514 1060 RSP * NCN: FIN DU TRAITEMENT.
0515 C302 MOV 3.(M) * CUI: LE SECOND CARACTERE DE LA PREMIERE
0516 A403 RME 4.(N) * ETIQUETTE EST LU ET AFFECTE AU REGISTRE
0517 8440 INC 4 * SF7.

LOCN	CODE	FLAGS	LABELS	OP	* OPERANDS	COMMENTS	PAGE	32
0518	8720			CPY	7,T	*		
0519	8380			ADD	3,L	*		
051A	C502			MOV	5,(M)	* LE SECOND CARACTERE DE LA SECONDE		
051B	A603			RMF	6,(N)	* ETIQUETTE EST LU ET AFFECTE AU REGISTRE		
051C	8640			INC	06	* SF8.		
051D	8820			CPY	8,T	*		
051E	8580			ADD	5,L	*		
051F	9738			SBT*	7,TC	* LES CARACTERES SONT CCMARES.		
0520	5004			TN	0,X'04'	* CARACTERES IDENTIQUES?		
0521	1060			RSP		* NCN: FIN DU TRAITEMENT.		
0522	4780			TZ	7,X'90'	* FIN DE LA PREMIERE ETIQUETTE?		
0523	1526			JP	*+3			
0524	5880			TN	8,X'90'	* FIN DE LA SECONDE ETIQUETTE?		
0525	150A			JP	ET1	* NON.		
0526	1040			SPF				
0527	8540			INC				
0528	1020			RTN	5	* OUI: L'INDICATEUR ZERO DU C.C. EST MIS A 1.		

* EXECUTION DE L'INSTRUCTION CLT.									
0529	2C02		CLT	LF	12,2	* L'INSTRUCTION EST DE TYPE 2.			
052A	009C			JE	ADRES	* LES OPERANDES SONT EXTRAITS.			
052B	025F			JE	SMPL	* LA NOUVELLE PILE EST INITIALISEE.			
052C	487F		LT1	TZ	8,X'7F'	* L'INDICATEUR DANS LA PREMIERE LISTE EST NUL?			
052D	1539			JP	LT2	*			
052E	49FF			TZ	9,X'FF'	*			
052F	1539			JP	LT2				
0530	11FB			LT	X'FB'	* OUI: L'EGALITE N'EST PAS ENCORE VERIFIEE.			
0531	E520			AND	5,T	*			
0532	4A7F			TZ	10,X'7F'	* L'INDICATEUR DANS LA SECONDE LISTE EST NUL?			
0533	1551			JP	COND	*			
0534	48FF			TZ	11,X'FF'	*			
0535	1551			JP	COND	* NCN: FIN DU TRAITEMENT.			
0536	1104			LT	X'04'	* CUI: L'EGALITE EST VERIFIEE JUSQU'A DATE.			
0537	C520			LOR	5,T	*			
0538	154D			JP	LT6				
0539	4A7F		LT2	TZ	10,X'7F'	* L'INDICATEUR DANS LA SECONDE LISTE EST NUL?			
053A	1540			JP	LT3	*			
053B	48FF			TZ	11,X'FF'	*			
053C	1540			JP	LT3				
053D	11FB			LT	X'FB'	* OUI: LES DEUX LISTES SONT DIFFERENTES.			
053E	E520			AND	5,T	*			
053F	1551			JP	COND				
0540	04F1		LT3	JE	ETIEG	* LES ETIQUETTES SONT COMPAREES.			
0541	4501			TZ	5,X'01'	* ETIQUETTES IDENTIQUES?			
0542	1546			JP	LT4	* OUI.			
0543	11FB			LT	X'FB'	* NCN: LES DEUX LISTES SONT DIFFERENTES.			
0544	E520			AND	5,T	*			
0545	1551			JP	COND				
0546	01B5		LT4	JE	EPL	* LES INDICATEURS SONT AFFECTES A LA PILE.			
0547	01A0			JE	FILS	* PASSAGE AUX FILS AINES.			
0548	01AF			JE	F2L5	*			
0549	152C			JP	LT1				
054A	01D1		LT5	JE	FIERE	* PASSAGE AUX FRERES SUIVANTS.			
054B	01E0			JE	F2ERE	*			
054C	152C			JP	LT1				

COMMENTS

LOCN CODE FLAGS LABELS OP * OPERANDS

0540 50FF TN 13,X'FF' * LA PILE EST VIDE?
 054E 1551 JP COND * CUI: FIN DU TRAITEMENT.
 054F 016E JE DPL * NON: LES INDICATEURS SONT RETIRES DU
 0550 154A JP LT5 SCMMET DE LA PILE.
 0551 11FC LT X'FC' * LE CODE CONDITIONNEL EST INITIALISE A 0.
 0552 E520 AND 5.T *
 0553 4504 TZ 5,X'04' * LA CCNDITION EST SATISFAITE?
 0554 8540 INC 5 * CUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
 0555 006D JE FETCH DU CODE CONDITIONNEL.

* * EXECUTION DE L'INSTRUCTION CAR.

0556 2C02 CAR LF 12,2 * L'INSTRUCTION EST DE TYPE 2.
 0557 009C JE ADRES * LES OPERANDES SONT EXTRAITS.
 0558 025F JE SMPL * LA NOUVELLE PILE EST INITIALISEE.
 0559 155B JP *+2
 055A 0185 JE EPL * LES INDICATEURS SONT AFFECTES A LA PILE.
 055B 04F1 JE ETIEG * LES ETIQUETTES SONT COMPAREES.
 055C 5501 TN 5,X'01' * ETIQUETTES IDENTIQUES?
 055D 1573 JP CT5
 055E 01A0 JE FILS * CUI: PASSAGE AUX FILS AINES?
 055F 01AF JE F2LS *
 0560 487F TZ 8,X'7F' * L'INDICATEUR DANS LE PREMIER ARBRE EST NUL?
 0561 156B JP CT3 *
 0562 49FF TZ 9,X'FF' *
 0563 156B JP CT3 * NCN.
 0564 4A7F TZ 10,X'7F' * L'INDICATEUR DANS LE DEUXIEME ARBRE EST NUL?
 0565 1573 JP CT5 *
 0566 48FF TZ 11,X'FF' *
 0567 1573 JP CT5 * NCN.
 0568 1104 LT X'04' * CUI: L'EQUALITE EST VERIFIE JUSQU'A DATE.
 0569 C520 LOR 5.T *
 056A 1576 JP CT6 *
 056B 4A7F TZ 10,X'7F' * L'INDICATEUR DANS LE DEUXIEME ARBRE EST NUL?
 056C 155A JP CT1 *
 056D 48FF TZ 11,X'FF' *
 056E 155A JP CT1 * NCN.
 056F 1573 JP CT5 * CUI.
 0570 01D1 JE F1ERE * PASSAGE AUX FRERES SUIVANTS.
 0571 01E0 JE F2ERE *
 0572 1560 JP CT2 *
 0573 11F8 LT X'FB' * LES DEUX ARBRES SONT DIFFERENTS.
 0574 E520 AND 5.T *
 0575 1551 JP COND *
 0576 50FF TN 13,X'FF' * LA PILE EST VIDE?
 0577 1551 JP COND * CUI: FIN DU TRAITEMENT.
 0578 016E JE DPL * LES INDICATEURS SONT RETIRES DU SOMMET
 0579 1570 JP CT4 DE LA PILE.

* * EXECUTION DE L'INSTRUCTION COL.

057A 1080 COL SSF
 057B CA01 MOV 10,(T) * L'INDICATEUR DU PROCHAIN CARACTERE A L'ENTREE
 057C B320 CPY 3,T * EST AFFECTE AU REGISTRE SF3.
 057D 1040 SPF

LOCN	CODE	FLAGS	LABELS	OP	* OPERANDS	COMMENTS	PAGE 34
057E	B602		ZOF	6,(M)	* LE CCNTENU DU REGISTRE R0 EST LU ET AFFECTE		
057F	2710		LF	7,X*10	* AUX REGISTRES PF6,7.		
0580	A703		RMF	7,(N)	*		
0581	B620		CPY	6,T	*		
0582	A7C3		RMF	7,I(N)	*		
0583	B720		CPY	7,T	*		
0584	F700		SFL	7	* L'ADRESSE DANS LE VECTEUR GRAMMAIRE A LAQUELLE		
0585	F680		SFL	5,L	* LE CODE DE L'ETIQUETTE DOIT ETRE AFFECTE EST		
0586	2803		LF	8,X*03	* CALCULEE ET AFFECTEE AUX REGISTRES PF6,7.		
0587	2900		LF	9,X*00	*		
0588	C901		MOV	9,(T)	*		
0589	B720		ADD	7,T	*		
058A	C801		MOV	8,(T)	*		
058B	B6A0		ADD	6,LT	*		
058C	1080		SSF		* LE PREMIER CARACTERE DE L'ETIQUETTE EST LU.		
058D	1201		LM	X*01	*		
058E	AA03		RMF	10,(N)	*		
058F	1040		SPF		*		
0590	B820		CPY	8,T	* LE CARACTERE EST AFFECTE AU REGISTRE PF8.		
0591	01EF		JE	LIRE	* PASSAGE AU PROCHAIN CARACTERE A L'ENTREE.		
0592	1080		SSF		*		
0593	1201		LM	X*01	* LE PROCHAIN CARACTERE A L'ENTREE EST LU		
0594	AA03		RMF	10,(N)	* ET AFFECTE AU REGISTRE PF9.		
0595	1040		SPF		*		
0596	B920		CPY	9,T	*		
0597	6920		CP	9,X*00	* LE CARACTERE DANS LE REGISTRE PF9 EST UN		
0598	1585		JP	LB2	* CHIFFRE ENTRE 0 ET 9?		
0599	69C6		CP	9,X*C6	*		
059A	15A0		JP	*+0	* OUI: LE CARACTERE FAIT PARTIE DE L'ETIQUETTE.		
059B	69BF		CP	9,X*BF	* LE CARACTERE DANS LE REGISTRE PF9 EST UNE		
059C	1585		JP	LB2	* LETTRE ENTRE A ET Z?		
059D	69A5		CP	9,X*A5	*		
059E	15A0		JP	*+2	* OUI: LE CARACTERE FAIT PARTIE DE L'ETIQUETTE.		
059F	1585		JP	LB2	* NCN.		
05A0	01EF		JE	LIRE	* PASSAGE AU PROCHAIN CARACTERE A L'ENTREE.		
05A1	1080		SSF		*		
05A2	1201		LM	X*01	* LE PROCHAIN CARACTERE A L'ENTREE EST LU		
05A3	AA03		RMF	10,(N)	* ET AFFECTE AU REGISTRE PF10.		
05A4	1040		SPF		*		
05A5	BA20		CPY	10,T	*		
05A6	6AD0		CP	10,X*00	* LE CARACTERE DANS LE REGISTRE PF10 EST		
05A7	1586		JP	LB3	* UN CHIFFRE ENTRE 0 ET 9?		
05A8	6AC6		CP	10,X*06	*		
05A9	15AF		JP	*+6	* OUI: LE CARACTERE FAIT PARTIE DE L'ETIQUETTE.		
05AA	6ABF		CP	10,X*BF	* LE CARACTERE DANS LE REGISTRE PF10 EST		
05AB	1586		JP	LB3	* UNE LETTRE ENTRE A ET Z?		
05AC	6AA5		CP	10,X*A5	*		
05AD	15AF		JP	*+2	* OUI: LE CARACTERE FAIT PARTIE DE L'ETIQUETTE.		
05AE	1586		JP	LB3	* NCN.		
05AF	0269		JE	STORE	* LES CARACTERES LUS SONT AFFECTES AU VECTEUR		
					* GRAMMAIRE.		
0580	1102		LT	X*02	* L'ADRESSE DANS LE VECTEUR GRAMMAIRE EST		
0581	8720		ADD	7,T	* INCREMENTEE DE 2.		
0582	8680		ADD	6,L	*		
0583	CA01		MOV	10,(T)	*		
0584	1590		JP	LB1			
0585	2920		LF	9,X*20	* L'ETIQUETTE CONTIENT UN NOMBRE IMPAIR DE		

* * CARACTERES. LE CODE D'UN ESPACE EST DONC
* * AFFECTE COMME DERNIER CARACTERE AU REGISTRE
* * PF9.

* L'INDICATEUR DE LA FIN DE L'ETIQUETTE EST
* AFFECTE AU BIT DE GAUCHE DU REGISTRE PF9.
* LES DERNIERS CARACTERES DE L'ETIQUETTE SONT
* AFFECTES AU VECTEUR GRAMMAIRE.
* LE NOUVEL INDICATEUR DE LA POSITION PRESENTE
* DANS LE VECTEUR GRAMMAIRE EST CALCULE ET
* AFFECTE AU REGISTRE R0.

* LES CARACTERES LUS SONT IMPRIMES.

EXECUTION DE L'INSTRUCTION APP.

05C8 2C00	APP	LF	12.0	ADRES	* L'INSTRUCTION EST DE TYPE 0.
05C9 009C	*	JE			* L'OPERANDE QUI INDIQUE LE VECTEUR CONDITION * PRESENTE EST EXTRAIT.
05CA B600		ZOF	5		* L'ADRESSE DU VECTEUR CONDITION PRESENTE EST
05CB BA00		ZOF	10		* AFFECTEE AUX REGISTRES PF6.7 ALORS QUE
05CC 4903		TZ	9.X*03		* L'ADRESSE DU VECTEUR CONDITION PRECEDENTE
05CD 15D1		JP	AP1		* EST AFFECTEE AUX REGISTRES PF10.11.
05CE 2720		LF	7.X*20		
05CF 2B5C		LF	11.X*5C		
05D0 15D3		JP	*+3		
05D1 275C	AP1	LF	7.X*5C		
05D2 2B20		LF	11.X*20		
05D3 2D14		LF	13.20		* LE COMPTEUR EST INITIALISE A 20.
05D4 CA02	AP2	MOV	10.(M)		* LE MOT DANS LE VECTEUR CONDITION PRECEDENTE
05D5 AB03		RMF	11.(N)		* EST LU ET AFFECTE AU REGISTRE PF8.
05D6 B820		CPY	8.T		
05D7 C602		MOV	9.(M)		* LE MOT DANS LE VECTEUR CONDITION PRESENTE
05D8 A703		RMF	7.(N)		* EST LU.
05D9 C821		LOR	8.T(T)		* LE LOGIQUE ENTRE LES DEUX MOTS EST EFFECTUE
05DA A713	*	WMF	7.(N)		* ET LE RESULTAT EST AFFECTE AU MOT DANS * LA CONDITION PRESENTE.
05DB 8740		INC	7		* PASSAGE AU MOT SUIVANT DANS LE VECTEUR
05DC 8680		ADD	6.L		* CONDITION PRESENTE.
05DD 8B40		INC	11		* PASSAGE AU MOT SUIVANT DANS LE VECTEUR
05DE 8A90		ADD	10.L		* CONDITION PRECEDENTE.
05DF 9D40		DEC	13		* LE COMPTEUR EST DECREMENTE DE 1.
05E0 40FF		TZ	13.X*FP		* LE COMPTEUR EST 0?
05E1 15D4		JP	AP2		* NCN.
05E2 005D		JE			

FETCH

* EXECUTION DE L'INSTRUCTION TPA.

05E3 2C01	TPA	LF	12.T	* L'INSTRUCTION EST DE TYPE 1.
05E4 009C		JE	ADRES	* L'OPERANDE EST EXTRAIT.
05E5 11FC		LT	X.FC	* LE CCDE CONDITIONNEL EST INITIALISE A 0.
05E6 E520		AND	5.T	*
05E7 8000		ZOF	13	* LE CCDE A COMPARER EST INITIALISE DANS PF13.
05E8 485F		TZ	8.X.5F	* L'OPERANDE EST UN PARAMETRE?
05E9 15F5		JP	TAI	*
05EA 5820		TN	8.X.20	*
05EB 15F5		JP	TAI	* NCN: L'OPERANDE EST UNE ETIQUETTE.
05EC 110F		LT	X.OF	* OUI: LE CODE DU PARAMETRE EST AFFECTE AU
05ED E920		AND	9.T	* REGISTRE PF9.
05EE 8D40		INC	13	* LE CCDE A COMPARER EST INCREMENTE DE 1.
05EF 69FA		CP	9.X.FA	* PARAMETRE ETIQUETTE?
05F0 15F5		JP	TAI	* CUI.
05F1 8D40		INC	13	* NCN: LE CCDE A COMPARER EST INCREMENTE DE 1.
05F2 69F6		CP	9.X.F6	* PARAMETRE ARBRE?
05F3 15F5		JP	TAI	* CUI.
05F4 8D40		INC	13	* NCN: LE CODE A COMPARER EST INCREMENTE DE 1.
05F5 1103	TAI	LT	X.O3	* LE CCDE DANS LE REGISTRE PF13 ET CELUI DANS
05F6 E129		AND*	1.T(T)	* L'INSTRUCTION SONT COMPARES.
05F7 9D20		SBT	13.T	*
05F8 5DFF		TN	13.X.FF	* CODES IDENTIQUES?
05F9 8540		INC	5	* OUI: LE RESULTAT EST INDIQUE PAR LE BIT ZERO
05FA 006D		JE	FETCH	* DU CODE CONDITIONNEL.

* EXECUTION DE L'INSTRUCTION RTS.

05FB 0154	RTS	JE	DPILE	* L'ADRESSE DE RETOUR EST RETIREE DU SOMMET
05FC 8B01		MOV	8.(T)	* DE LA PILE ET AFFECTEE AUX REGISTRES
05FD BA20		CPY	10.T	* PF10.11.
05FE C901		MOV	9.(T)	*
05FF BB20		CPY	11.T	*
0600 0154		JE	DPILE	* LE NCMBRE DE VARIABLES EST RETIRE DU SOMMET
0601 C901		MOV	9.(T)	* DE LA PILE ET AFFECTE AU REGISTRE PF13.
0602 BD20		CPY	13.T	*
0603 5DFF	RTI	TN	13.X.FF	* AUTRES VARIABLES DANS LA PILE?
0604 1C0E		JP	RT2	*
0605 0147		JE	DEPL	* OUI: L'ADRESSE DE LA VARIABLE EST CALCULEE.
0606 0154		JE	DPILE	* LA VARIABLE EST RETIREE DE LA PILE.
0607 C301		MOV	3.(T)	* L'ADRESSE DE LA VARIABLE DANS LES REGISTRES
0608 B620		CPY	5.T	* PF3.4 EST TRANSFEREE AUX REGISTRES PF6.7.
0609 C401		MOV	4.(T)	*
060A B720		CPY	7.T	*
060B 0269		JE	STORE	* LA VARIABLE EST AFFECTE A CETTE ADRESSE.
060C 9D40		DEC	13	* LE NOMBRE DE VARIABLES EST DECREMENTE DE 1.
060D 1C03		JP	RT1	*
060E CA01	RT2	MOV	10.(T)	* L'ADRESSE DE RETOUR EST AFFECTEE AU COMPTEUR
060F BE20		CPY	14.T	* D'INSTRUCTION.
0610 CB01		MOV	11.(T)	*
0611 BF20		CPY	15.T	*
0612 006D		JE	FETCH	*

* EXECUTION DE L'INSTRUCTION ROC.

0613	1080	* RDC	SSF		
0614	CA01		MOV	10.(T)	* L'INDICATEUR DU PROCHAIN CARACTERE A L'ENTREE
0615	B320		CPY	3.T	* EST AFFECTE AU REGISTRE SF3.
0616	1040		SPF		
0617	01EF		JE		
0618	5103		TN	LIRE	* PASSAGE AU PROCHAIN CARACTERE A L'ENTREE.
0619	006D		JE	1.X*03	* CARACTERES LUS A ETRE IMPRIMES?
061A	4101		TZ	FETCH	* NCN.
061B	1C1E		JP	1.X*01	* ESPACES A ETRE IMPRIMES?
061C	0257		JE	*+3	
061D	006D		JE	PRINT	* NCN: SEUL LE DERNIER CARACTERE LU EST
061E	020E		JE	FETCH	IMPRIME.
061F	006D		JE	NPRINT	* CUI: TCUS LES CARACTERES LUS. INCLUANT LES
			JE	FETCH	ESPACES, SONT IMPRIMES.

** EXECUTION DE L'INSTRUCTION WRC.

0620	CE02	WRC	MOV	14.(M)	* LE CARACTERE SUIVANT L'INSTRUCTION EST LU
0621	AF03		RMF	15.(N)	* ET AFFECTE AU REGISTRE PF9.
0622	8F40		INC	15	* DE PLUS, LE COMPTEUR D'INSTRUCTIONS EST MIS
0623	8920		CPY	9.T	* A JOUR.
0624	8E80		ADD	14.L	
0625	49FB		TZ	9.X*FB	* LE CARACTERE QUI SUIV L'INSTRUCTION POSSEDE
0626	1C2B		JP	*+5	* UN CODE DE 04 (EOT) ?
0627	5904		TN	9.X*04	
0628	1C2B		JP	*+3	* NCN.
0629	1080		SSF		* OUI: LE COMPTEUR DES CARACTERES A LA SORTIE
062A	2F4F		LF	15.X*4F	* DANS LE REGISTRE SF15 EST MIS A 79.
062B	0220		JE	OUTPUT	* LE CARACTERE EST AFFECTE AU FICHIER DE SORTIE.
062C	006D		JE	FETCH	

** EXECUTION DES INSTRUCTIONS SET ET CLI.

062D	2C00	SET	LF	12.0	* L'INSTRUCTION EST DE TYPE 0.
062E	009C		JE	ADRES	* L'OPERANDE EST EXTRAIT.
062F	117F		LT	X*7F	* LE BIT DE GAUCHE DE L'OPERANDE EST
0630	E820		AND	8.T	* REMIS A 0.
0631	1180		LT	X*80	
0632	5110		TN	1.X*10	* INSTRUCTION SET?
0633	C820		LOR	8.T	* CUI: LE BIT GAUCHE DE L'OPERANDE EST MIS A 1.
0634	0269		JE	STORE	* L'OPERANDE EST RETOURNE A SON ADRESSE.
0635	006D		JE	FETCH	

** EXECUTION DE L'INSTRUCTION TPP.

0636	2C00	TPP	LF	12.0	* L'INSTRUCTION EST DE TYPE 0.
0637	009C		JE	ADRES	* L'OPERANDE QUI INDIQUE LE VECTEUR EST EXTRAIT.
0638	026F		JE	VECPP	* L'ADRESSE DU VECTEUR EST CALCULEE.
0639	2A00		LF	10.X*00	* L'ADRESSE DU VECTEUR MEMBRE GAUCHE EST CAL-
063A	2834		LF	11.X*34	* CULEE ET AFFECTEE AUX REGISTRES PF10,11.
063B	11FC		LT	X*FC	* LE CODE CONDITIONNEL EST INITIALISE A 0.
063C	E520		AND	5.T	
063D	2D14		LF	13.20	* LE COMPTEUR EST INITIALISE A 20.
063E	CA02	TP1	MOV	10.(M)	* LE MCT DANS LE VECTEUR MEMBRE GAUCHE EST LU

LOCN CODE	FLAGS	OP	* OPERANDS	COMMENTS	PAGE 38
063F	AB03	RMF	11,(N)	* ET AFFECTE AU REGISTRE PF8.	
0640	B820	CPY	8,T		
0641	C602	MOV	6,(M)	* LE MCT DANS LE VECTEUR A ETRE VERIFIE EST LU.	
0642	A703	RMF	7,(N)		
0643	D829	XOR*	8,T(T)	* LES PARAMETRES PRESENTS DANS LE VECTEUR A	
0644	E820	AND	8,T	* ETRE VERIFIE SONT AUSSI PRESENT DANS LE	
0645	D820	XUR	8,T	* VECTEUR MEMBRE GAUCHE?	
0646	48FF	TZ	8,X*FF		
0647	006D	JE	FETCH		
0648	8740	INC	7		
0649	8580	ADD	6,L	* OUI: PASSAGE AU MOT SUIVANT DANS LE VECTEUR	
064A	8840	INC	11	* A ETRE VERIFIE.	
064B	8A80	ADD	10,L	* PASSAGE AU MOT SUIVANT DANS LE VECTEUR	
064C	9D40	DEC	13	* MEMBRE GAUCHE.	
064D	4DFF	TZ	13,X*FF	* LE COMPTEUR EST DECREMENTE DE 1.	
064E	1C3E	JP	TP1	* LE COMPTEUR EST 0?	
064F	8540	INC	5		
0650	006D	JE	FETCH	* OUI: LA PRESENCE DES PARAMETRES EST VERIFIEE.	
* * * EXECUTION DE L'INSTRUCTION SPM. * * *					
0651	11F8	LT	X*FB		
0652	E520	AND	5,T		
0653	5101	TN	1,X*01	* LE CODE DANS L'INSTRUCTION EST 01?	
0654	1C59	JP	SP1		
0655	4102	TZ	1,X*02		
0656	1C5E	JP	SP2		
0657	023B	JE	PUT	* OUI: LA LIGNE EST IMPRIMEE.	
0658	1C78	JP	SP3		
0659	5102	TN	1,X*02	* LE CODE DANS L'INSTRUCTION EST 10?	
065A	1C5E	JP	SP2	* NON.	
065B	1000	NOP		* SORTIE SUR LA MEME LIGNE.	
065C	023B	JE	PUT	* LA LIGNE EST IMPRIMEE.	
065D	006D	JE	FETCH		
065E	CE02	MOV	14,(M)	* LA LIMITE INFERIEURE DU VECTEUR DE SORTIE	
065F	AF03	RMF	15,(N)	* SUR L'IMPRIMANTE EST LUE ET AFFECTEE AU	
0660	8F40	INC	15	* REGISTRE SF13.	
0661	8E80	ADD	14,L	* DE PLUS, LE COMPTEUR D'INSTRUCTIONS EST	
0662	1080	SSF		* MIS A JOUR.	
0663	BD20	CPY	13,T		
0664	1040	SPF			
0665	CE02	MOV	14,(M)	* LA LIMITE SUPERIEURE DU VECTEUR DE SORTIE	
0666	AF03	RMF	15,(N)	* SUR L'IMPRIMANTE EST LUE ET AFFECTEE AU	
0667	8F40	INC	15	* REGISTRE SF14.	
0668	8E80	ADD	14,L	* DE PLUS, LE COMPTEUR D'INSTRUCTIONS EST	
0669	1080	SSF		* MIS A JOUR.	
066A	BE20	CPY	14,T		
066B	4D7F	TZ	13,X*7F	* LA LIMITE INFERIEURE EST 0?	
066C	9D40	DEC	13	* NON.	
066D	9E40	DEC	14		
066E	4E80	TZ	14,X*80	* LA LIMITE SUPERIEURE EST > 128 ?	
066F	2EFF	LF	14,X*FF	* OUI: LA LIMITE SUPERIEURE EST MISE A 127.	
0670	117F	LT	X*7F	* LE BIT DE GAUCHE DES 2 OCTETS INDICANT LES	
0671	EE20	AND	14,T	* LIMITES, DANS LES REGISTRES SF13 ET SF14.	
0672	ED21	AND	13,T(T)	* EST MIS A 0.	
0673	9E3B	SBT*	14,TC	* LA LIMITE INFERIEURE EST PLUS GRANDE QUE	

0674	4002	TZ	0,X'02'	* LA LIMITE SUPERIEURE?
0675	BD00	ZOF	13,X'03'	* OUI: LA LIMITE INFERIEURE EST MISE A 0.
0676	4103	TZ	1,X'03'	* LE CODE DANS L'INSTRUCTION EST 11?
0677	023B	JE	PUT	* CUI: LA LIGNE EST IMPRIMEE.
0678	1080	SSF		
0679	2C80	LF	12,X'80'	* LE VECTEUR DE SORTIE SUR L'IMPRIMANTE EST
067A	1120	LT	X'20'	* INITIALISE A DES ESPACES.
067B	1201	LM	X'01'	*
067C	AC13	WMF	12,(N)	*
067D	8C40	INC	12	*
067E	4CFF	TZ	12,X'FF'	*
067F	1C7C	JP	*-3	*
0680	CD01	MOV	13,(T)	* L'INDICATEUR DU PROCHAIN CARACTERE A LA SORTIE
0681	5C20	CPY	12,T	* EST INITIALISE A LA LIMITE INFERIEURE DU
0682	1040	SPF		VECTEUR DE SORTIE.
0683	006D	JE	FETCH	
* * EXECUTION DE L'INSTRUCTION DEC.				
0684	2C00	DEC	12,0	* L'INSTRUCTION EST DE TYPE 0.
0685	009C	JE	ADRES	* L'OPERANDE QUI CONSISTE EN UN NOMBRE BINAIRE
0686	C801	MOV	8,(T)	* EST EXTRAIT ET AFFECTE AUX REGISTRES PF10,11.
0687	BA20	CPY	10,T	*
0688	C901	MOV	9,(T)	*
0689	B920	CPY	11,T	*
068A	2600	LF	6,X'00'	* L'ADRESSE DES CONSTANTES 10,000, 1,000, 100
068B	2718	LF	7,X'18'	* ET 10 EST AFFECTE AUX REGISTRES PF6,7.
068C	2404	LF	4,4	* LE COMPTEUR EST INITIALISE A 4.
068D	2930	LF	9,X'30'	* LES 4 BITS DE GAUCHE DU CODE DES CHIFFRES
* SONT AFFECTES AU REGISTRE PF9.				
068E	C602	MOV	6,(M)	* LA CONSTANCE A L'ADRESSE DANS LES REGISTRES
068F	A703	RMF	7,(N)	* PF6,7 EST SOUSTRAITE DU NOMBRE DANS LES
0690	9B20	SBT	11,T	* REGISTRES PF10,11.
0691	A7CB	RMF*	7,I(N)	*
0692	9A80	SBT	10,LTC	* RESULTAT NEGATIF?
0693	4002	TZ	0,X'02'	* CUI.
0694	1C97	JP	DC2	* NON: LE CODE DU CHIFFRE EST INCREMENTE DE 1.
0695	8940	INC	9	
0696	1C8E	JP	DC1+1	
0697	C602	MOV	6,(M)	* LA CONSTANCE A L'ADRESSE DANS LES REGISTRES
0698	A703	RMF	7,(N)	* PF6,7 EST ADDITIONNEE AU NOMBRE DANS LES
0699	8B20	ADD	11,T	* REGISTRES PF10,11 AFIN QUE LE NOMBRE
069A	A7CB	RMF*	7,I(N)	* REDEVienne POSITIF.
069B	8AA0	ADD	10,LT	*
069C	4120	TZ	1,X'20'	* LE CHIFFRE DANS LE REGISTRE PF9 EST AFFECTE
069D	1CA0	JP	*+3	* SOIT AU VECTEUR DE SORTIE SUR L'IMPRIMANTE,
069E	022E	JE	PCHAR	* SOIT AU FICHIER DE SORTIE, DEPENDANT DU BIT
069F	1CA1	JP	*+2	* DE ORCITE DU CODE OPERATION.
06A0	0220	JE	OUTPUT	*
06A1	3702	AF	7,2	* PASSAGE A LA PROCHAINE CONSTANCE.
06A2	9440	DEC	4	* LE COMPTEUR EST DECREMENTE DE 1.
06A3	4AFF	TZ	4,X'FF'	* LE COMPTEUR EST 0?
06A4	1C8D	JP	DC1	* NCN.
06A5	1130	LT	X'30'	* CUI: LE CHIFFRE REPRESENTANT LES UNITES
06A6	CB21	LOR	11,T(T)	* EST AFFECTE AU REGISTRE PF9.
06A7	B920	CPY	9,T	*

LOCN	CODE	FLAGS	LABELS	OP	* OPERANDS	CCMMENTS	PAGE
06A8	4120			TZ	1.X'20'	* LE CHIFFRE DANS LE REGISTRE PF9 EST AFFECTE	40
06A9	1CAC			JP	**3	* SOIT AU VECTEUR DE SORTIE SUR L'IMPRIMANTE,	
06AA	022E			JE	PCHAR	* SOIT AU FICHIER DE SORTIE, DEPENDANT DU BIT	
06AB	1CAD			JP	**2	* DE DROITE DU CODE OPERATION.	
06AC	0220			JE	OUTPUT		
06AD	005D			JE	FETCH		
** EXECUTION DE L'INSTRUCTION CDP.							
06AE	1080			SSP			
06AF	CA01			MOV	10.(T)	* L'INDICATEUR DU PROCHAIN CARACTERE A L'ENTREE	
06B0	B320			CPY	3.T	* EST AFFECTE AU REGISTRE SF3.	
06B1	1040			SPF			
06B2	B602			ZOF	6.(M)	* LE CCNTENU DU REGISTRE R0 EST LU ET AFFECTE	
06B3	2710			LF	7.X'10'	* AUX REGISTRES PF6.7.	
06B4	A703			RMF	7.(N)		
06B5	B620			CPY	6.T		
06B6	A7C3			RMF	7.I(N)		
06B7	B720			CPY	7.T		
06B8	F700			SFL	7	* L'ADRESSE DANS LE VECTEUR GRAMMAIRE A	
06B9	F680			SFL	6.L	* LAQUELLE LE CODE DU PARAMETRE DOIT ETRE	
06BA	2803			LF	8.X'03'	* AFFECTE EST CALCULEE ET AFFECTE AUX	
06BB	2900			LF	9.X'00'	* REGISTRES PF6.7.	
06BC	C901			MOV	9.(T)		
06BD	8720			ADD	7.T		
06BE	C801			MOV	8.(T)		
06BF	86A0			ADD	6.LT		
06C0	1080			SSP			
06C1	1201			LM	X'01'	* LE PREMIER CARACTERE DU PARAMETRE EST LU	
06C2	AA03			RMF	10.(N)	* ET AFFECTE AU REGISTRE PF9.	
06C3	1040			SPF			
06C4	B920			CPY	9.T		
** TRAITEMENT DU PARAMETRE LISTE.							
06C5	69AB			CP	9.X'AB'	* CARACTERE ENTRE U ET Z?	
06C6	1CCC			JP	PAR	* NON.	
06C7	110F			LT	X'0F'	* LE CCDE TYPE DU PARAMETRE LISTE EST	
06C8	E920			AND	9.T	* AFFECTE AU REGISTRE PF9.	
06C9	1105			LT	X'05'		
06CA	8920			ADD	9.T		
06CB	1CD6			JP	PA3		
** TRAITEMENT DU PARAMETRE ARBRE.							
06CC	69B7			PAP			
06CD	1CD3			CP	9.X'B7'	* CARACTERE ENTRE I ET L?	
06CE	110F			JP	PET	* NON.	
06CF	E920			LT	X'0F'	* LE CCDE TYPE DU PARAMETRE ARBRE EST	
06D0	1103			AND	9.T	* AFFECTE AU REGISTRE PF9.	
06D1	9920			LT	X'03'		
06D2	1CD6			SBT	9.T		
				JP	PA3		
** TRAITEMENT DU PARAMETRE ETIQUETTE.							
06D3	110F			PET	LT	* LE CCDE TYPE DU PARAMETRE ETIQUETTE EST	

LUCN	CODE	FLAGS	LABELS	OP	* OPERANDS	COMMENTS	PAGE	41
06D4	E920	AND		9.T		* AFFECTE AU REGISTRE PF9.		
06D5	9940	DEC		9				
06D6	01EF	JE	PA3	LIRE		* PASSAGE AU CARACTERE SUIVANT, SOIT L'INDICE.		
06D7	01EF	JE		LIRE		* PASSAGE AU CARACTERE SUIVANT, SOIT L'INDICE.		
06D8	1080	SSF						
06D9	1201	LM		X'01'		* LE CARACTERE SUIVANT (L'INDICE) EST LU		
06DA	AA03	RMF		10.(N)		* ET AFFECTE AU REGISTRE PF8.		
06DB	1040	SPF						
06DC	B820	CPY		8.T		* LE CARACTERE EST UN CHIFFRE ENTRE 0 ET 9?		
06DD	68D0	CP		8.X'D0'				
06DE	1CE9	JP		PA4				
06DF	68C6	CP		8.X'C6'				
06E0	1CE2	JP		PA4				
06E1	1CE9	JP		PA4		* NON.		
06E2	01EF	JE		LIRE		* OUI: PASSAGE AU PROCHAIN CARACTERE A L'ENTREE.		
06E3	F800	SFL		8		* LE CODE DE L'INDICE EST AFFECTE AUX 4 BITS		
06E4	F800	SFL		8		* DE GAUCHE DU REGISTRE PF9.		
06E5	F800	SFL		8				
06E6	F800	SFL		8				
06E7	C801	MOV		8.(T)				
06E8	C920	LOR		9.T				
06E9	2820	LF	PA4	8.X'20'		* LE CODE DES PARAMETRES EST AFFECTE A PF8.		
06EA	0269	JE		STORE		* LE CODE COMPLET DU PARAMETRE EST AFFECTE		
06EB	C901	MOV		9.(T)		* AU VECTEUR GRAMMAIRE.		
06EC	B320	CPY		3.T		* LE DEUXIEME OCTET DU CODE DU PARAMETRE EST		
06ED	1102	LT		X'02'		* AFFECTE AU REGISTRE PF3.		
06EE	8720	AOD		7.T		* LE NOUVEL INDICATEUR DE LA POSITION PRESENTE		
06EF	8680	ADD		6.L		* DANS LE VECTEUR GRAMMAIRE EST CALCULE ET		
06F0	1103	LT		X'03'		* AFFECTE AU REGISTRE R0.		
06F1	9620	SBT		6.T				
06F2	F620	SFR		6				
06F3	F7A0	SFR		7.L				
06F4	B802	ZOF		8.(M)				
06F5	2910	LF		9.X'10'				
06F6	A913	WMF		9.(N)				
06F7	C601	MOV		6.(T)				
06F8	A9D3	WMF		9.I(N)				
06F9	C701	MOV		7.(T)				
06FA	020E	JE		NPRNT		* LES CARACTERES LUS SONT IMPRIMES.		
						* AFFECTATION DU PARAMETRE DANS LE VECTEUR PARAMETRE PRESENT.		
06FB	2C00	LF		12.0		* L'INSTRUCTION EST DU TYPE 0.		
06FC	009C	JE		ADRES		* L'OPERANDE EST EXTRAIT.		
06FD	026F	JE		VECPD		* L'ADRESSE DU VECTEUR EST CALCULEE.		
06FE	53F0	TN	PA5	3.X'F0'		* LA POSITION DU MOT DANS LE VECTEUR PARAMETRES		
06FF	1D05	JP		PA6		* PRESENTS DEVANT RETENIR LA PRESENCE DU		
0700	1102	LT		X'02'		* PARAMETRE EST INDIQUEE PAR L'INDICE, SOIT		
0701	8720	ADD		7.T		* LES 4 BITS DE GAUCHE DU CODE DU PARAMETRE		
0702	1110	LT		X'10'		* DANS LE REGISTRE PF3. L'ADRESSE DE CE MOT		
0703	9320	SBT		3.T		* EST CALCULEE ET AFFECTEE AUX REGISTRES		
0704	1CFE	JP		PA5		* PF6,7.		
0705	4308	IZ	PA6	3.X'08'		* L'ADRESSE DE L'OCTET DEVANT RETENIR LE PRE-		
0706	8740	INC		7		* SENCE DU PARAMETRE EST CALCULEE.		
0707	2980	LF		9.X'80'		* LA POSITION EXACTE DU BIT SERVANT A RETENIR		
0708	5307	TN		3.X'07'		* LA PRESENCE DU PARAMETRE EST CALCULEE ET		

OP	PA7	PA7	COMMENTS
0709 100D	JP	PA7	* UN 1 EST AFFECTE A CETTE POSITION DANS LE
070A F920	SFR	9	* REGISTRE PF9.
070B 9340	DEC	3	*
070C 1D08	JP	PA6+3	* LE BIT EST AFFECTE AU VECTEUR PARAMETRE
070D 1200	LM	X'00'	* INDIQUANT LA PRESENCE DU PARAMETRE CODE.
070E A703	RMF	7.(N)	**
070F C921	LUR	9.(T)	**
0710 A713	WMF	7.(N)	*
0711 006D	JE	FETCH	*

* * EXECUTION DE L'INSTRUCTION CPP.

OP	PA7	PA7	COMMENTS
0712 2C00	CPP	LF 12.0	* L'INSTRUCTION EST DE TYPE 0.
0713 009C	JE	ASRES	* L'OPERANDE INDIQUANT LE VECTEUR A ETRE INITIALIA-
0714 026F	JE	VECP	* LISE EST EXTRAIT ET L'ADRESSE DU VECTEUR
			* EST CALCULEE.
0715 B800	ZOF	8	* LE CCNTENU DES REGISTRES PF8,9 EST REMIS A 0.
0716 B900	ZOF	9	*
0717 2D0A	LF	13.10	* LE COMPTEUR EST INITIALISE A 10.
0718 0269	JE	STORE	* LE MOT DU VECTEUR EST INITIALISE A 0.
0719 1102	LT	2	* PASSAGE AU MOT SUIVANT DANS LE VECTEUR.
071A 8720	ADD	7.1	*
071B 8680	ADD	6.L	*
071C 9D40	DEC	13	* LE COMPTEUR EST DECREMENTE DE 1.
071D 4DFF	TZ	13.X'FF'	* LE COMPTEUR EST 0?
071E 1D18	JP	CPI	* NCN.
071F 006D	JE	FETCH	*

* * EXECUTION DE L'INSTRUCTION DOT.

OP	PA7	PA7	COMMENTS
0720 2C00	DOT	LF 12.0	* L'INSTRUCTION EST DE TYPE 0.
0721 009C	JE	ADRES	* L'ADRESSE DE L'ARBRE EST EXTRAITE.
0722 025F	JE	SMPL	* LA NOUVELLE PILE EST INITIALISEE.
0723 C801	MOV	8.(T)	* L'INDICATEUR DU DEBUT DE L'ARBRE DANS LES
0724 BA20	CPY	10.(T)	* REGISTRES PF8,9 EST TRANSFERE AUX REGISTRES
0725 C901	MOV	9.(T)	* PF10,11.
0726 8B20	CPY	11.1	*
0727 8D40	INC	13	* L'INDICATEUR DU NOMBRE D'ELEMENTS DANS LA
			* PILE EST INCREMENTE DE 1.
0728 8440	INC	4	* L'INDICATEUR DU SOMMET DE LA PILE EST
0729 8443	INC	4.(N)	* INCREMENTE DE 2.
072A A392	WMF	3.L(M)	* LE CCNTENU DES REGISTRES PF10,11 EST AFFECTE
072B CA01	MOV	10.(T)	* AU SOMMET DE LA PILE.
072C A4DB	WMF	4.(N)	*
072D CB01	MOV	11.(T)	*
072E 0754	JE	SORTI	* L'ETIQUETTE EST AFFECTEE A LA SORTIE.
072F 01AF	JE	F2LS	* PASSAGE AU FILS AINE.
0730 4AFF	TZ	10.X'FF'	* L'INDICATEUR DE LA POSITION PRESENTE DANS
0731 1D34	JP	13	* L'ARBRE EST NUL?
0732 5BFF	TN	11.X'FF'	*
0733 1D48	JP	D04	* OUI.
0734 2928	LF	9.X'28'	* NON: LE CODE DU CARACTERE '()' EST AFFECTE
			* AU REGISTRE PF9.
0735 4110	SQCAR	1.X'10'	* LE REGISTRE PF9 EST AFFECTE
0736 1D39	JP	13	* LE CARACTERE DANS LE REGISTRE PF9 EST IMPRIMENTE.

LOCN	CODE	FLAGS	LABELS	OP	* OPERANDS	COMMENTS	PAGE 43
0737	022E			JE	PCHAR	* SOIT AU FICHIER DE SORTIE, DEPENDANT DU 4IEME	
0738	1D27			JP	DO1	* BIT DE GAUCHE DU CODE OPERATION.	
0739	0220			JE	OUTPUT		
073A	1D27			JP	DO1		
073B	01E0		DO2	JE	F2ERE	* PASSAGE AU FRERE SUIVANT.	
073C	4AFF			TZ	10,X'FF'	* L'INDICATEUR DE LA POSITION PRESENTE DANS	
073D	1D40			JP	*+3	* L'ARBRE EST NUL?	
073E	5BFF			TN	11,X'FF'		
073F	1D42			JP	DO3		
0740	292C			LF	9,X'2C'	* OUI.	
0741	1D35			JP	SOCAR	* NON: LE CODE DU CARACTERE '0' EST AFFECTE AU	
0742	2929			LF	9,X'29'	REGISTRE PF9.	
0743	4110		DO3	TZ	1,X'10'	* LE CODE DU CARACTERE '1' EST AFFECTE A PF9.	
0744	1D47			JP	*+3	* LE CARACTERE DANS LE REGISTRE PF9 EST AFFECTE	
0745	022E			JE	PCHAR	* SOIT AU VECTEUR DE SORTIE SUR L'IMPRIMANTE.	
0746	1D48			JP	*+2	* SOIT AU FICHIER DE SORTIE, DEPENDANT DU 4IEME	
0747	0220			JE	OUTPUT	* BIT DE GAUCHE DU CODE OPERATION.	
0748	9D40		DO4	DEC	13	* L'INDICATEUR DU NOMBRE D'ELEMENTS SUR LA PILE	
0749	C403			MOV	4,(N)	* EST DECREMENTE DE 1.	
074A	A302			RMF	3,(M)	* L'ELEMENT AU SOMMET DE LA PILE EST AFFECTE	
074B	BA20			CPY	10,T	* AUX REGISTRES PF10,11.	
074C	A4CB			RMF	4,I(N)		
074D	BB20			CPY	11,T		
074E	1102			LT	2	* L'INDICATEUR DU SOMMET DE LA PILE EST	
074F	9420			SBT	4,T	* DECREMENTE DE 2.	
0750	9380			SBT	3,L		
0751	4DFF			TZ	13,X'FF'	* LA PILE EST VIDE?	
0752	103B			JP	DO2	* NCN.	
0753	006D			JE	FETCH		
0754	CA01			MOV	10,(T)	* L'INDICATEUR DANS LES REGISTRES PF10,11 EST	
0755	B620			CPY	6,T	* TRANSFERE AUX REGISTRES PF6,7.	
0756	CB01			MOV	11,(T)		
0757	B720			CPY	7,T		
0758	F700			SFL	7'	* L'ADRESSE ABSOLUE DU PREMIER MOT DE	
0759	F680			SFL	6,L	* L'ETIQUETTE EST CALCULEE A PARTIR DE	
075A	1100			LT	X'00'	* L'INDICATEUR DANS LES REGISTRES PF6,7.	
075B	8720			ADD	7,T		
075C	1103			LT	X'03'		
075D	86A0			ADD	6,L		
075E	C602			MOV	6,(M)	* LE CARACTERE DANS LE PREMIER OCTET DU MOT	
075F	A703		SOI	RMF	7,(N)	* EST LU ET AFFECTE AU REGISTRE PF9.	
0760	B920			CPY	9,T		
0761	117F			LT	X'7F'	* LE BIT DE GAUCHE DE L'OCTET EST MIS A 0.	
0762	E920			AND	9,T		
0763	4110			TZ	1,X'10'	* LE CARACTERE DANS LE REGISTRE PF9 EST AFFECTE	
0764	1D67			JP	*+3	* SOIT AU VECTEUR DE SORTIE SUR L'IMPRIMANTE.	
0765	022E			JE	PCHAR	* SOIT AU FICHIER DE SORTIE, DEPENDANT DU 4IEME	
0766	1D68			JP	*+2	* BIT DE GAUCHE DU CODE OPERATION.	
0767	0220			JE	OUTPUT		
0768	A7C3			RMF	7,I(N)	* LE CARACTERE DANS LE DEUXIEME OCTET DU MOT	

* * SOUS-PROGRAMME QUI AFFECTE L'ETIQUETTE DONT L'ADRESSE EST DANS
 * * LES REGISTRES PF10,11 SOIT AU VECTEUR DE SORTIE SUR L'IMPRIMANTE.
 * * SOIT AU FICHIER DE SORTIE.

4

LOCN	CODE	FLAGS	LABELS	UP	OPERANDS	COMMENTS	PAGE
0769	B920	CPY			9.T	* EST LU ET AFFECTE AU REGISTRE PF9.	44
076A	4980	TZ			9.X.80	* DERNIER CARACTERE DE L'ETIQUETTE?	
076B	1D74	JP			S02	* CUI.	
076C	4110	TZ			1.X.10	* NCN: LE CARACTERE DANS LE REGISTRE PF9 EST	
076D	1D70	JP			*+3	* AFFECTE SOIT AU VECTEUR DE SORTIE SUR	
076E	022E	JE			PCHAR	* L'IMPRIMANTE, SOIT AU FICHIER DE SORTIE,	
076F	1D71	JP			*+2	* DEPENDANT DU 4IEME BIT DE GAUCHE DU CODE	
0770	0220	JE			OUTPUT	* OPERATION.	
0771	8740	INC			7	* PASSAGE AU MOT SUIVANT.	
0772	8680	ADD			6.L	*	
0773	1D5E	JP			S01		
0774	495F	TZ	S02		9.X.5F	* LE DERNIER CARACTERE EST UN ESPACE?	
0775	1D78	JP			*+3	*	
0776	4920	TZ			9.X.20	*	
0777	1020	RTN				* CUI.	
0778	117F	LT			X.7F	* NCN: LE BIT DE GAUCHE DU DERNIER OCTET EST	
0779	E920	AND			9.T	* MIS A 0.	
077A	4110	TZ			1.X.10	* LE CARACTERE DANS LE REGISTRE PF9 EST AFFECTE	
077B	107E	JP			*+3	* SUIVANT AU VECTEUR DE SORTIE SUR L'IMPRIMANTE.	
077C	022E	JE			PCHAR	* SOIT AU FICHIER DE SORTIE, DEPENDANT DU 4IEME	
077D	1020	RTN				* BIT DE GAUCHE DU CODE OPERATION.	
077E	0220	JE			OUTPUT	*	
077F	1020	RTN					
		END					

Cet appendice présente un langage assembleur proposé pour faciliter la programmation du système. Entre autres, ce langage élimine le calcul des adresses et simplifie la représentation des nombres et des caractères.

B.1 Language source.

Avec l'assembleur, le langage source devient une succession de commandes symboliques. Ces commandes prennent la forme suivante:

NOM: OPERATION OPERANDES

- zone NOM.

La zone NOM peut contenir un symbole d'au plus 6 caractères qui représente l'adresse du premier octet à cet emplacement.

- zone OPERATION.

Cette zone doit contenir le code mnémorique d'une instruction machine ou d'une commande de l'assembleur.

- zone OPERANDES.

Les symboles dans cette zone identifient les adresses des opérandes ainsi que les modes d'adressage à être utilisés. Cette zone est vide dans le cas des instructions qui ne nécessitent aucun opérande.

B.2 Constantes.

Une constante peut représenter soit un nombre décimal, un nombre hexadécimal, une suite de caractères ou une adresse.

- constantes décimales.

Toute constante décimale est représentée par un nombre décimal entre -32768 et 32767. (Ex: -32 , 538)

- constantes hexadécimales.

Toute constante hexadécimale est représentée par un nombre hexadécimal d'au plus 4 caractères, les lettres A à F étant utilisés pour identifier les entiers 10 à 15. Ce nombre apparaît entre guillemets et doit être précédé de la lettre X. (Ex: X'FE' , X'802A')

- caractères.

Toute suite de caractères utilisée comme constante doit apparaître entre guillemets et être précédé de la lettre A. (Ex: A'ABC' , A'UNE CHAINE')

- adresses.

Une adresse symbolique peut être utilisée comme constante si elle est précédée du caractère #. Par exemple, #MEMBRE peut représenter la constante hexadécimale X'02FF' si le symbole MEMBRE représente l'adresse 02FF.

B.3 Adressage.

Dépendant du mode d'adressage, il existe plusieurs façons de représenter l'adresse d'une opérande. Une représentation symbolique est donc associée à chacun des modes d'adressage.

<u>MODE D'ADRESSAGE</u>	<u>REPRESENTATION SYMBOLIQUE</u>
Immédiat	nnn X'h..h' 1 à 4 chiffres
Relatif	#adr
Direct	adr
Indirect	@#adr
Registre	éadr
Vecteur indexé reg. 0	Rn
Vecteur indexé	V(R0)
Vecteur indexé moins 1	V(adr)
Vecteur indexé moins 2	V(adr-1)
Registre indexé	V(adr-2)
	adr(Rn)

Les abréviations utilisées dans ce tableau sont définies à la fin de l'appendice.

B.4 Instructions machine.

Le tableau suivant définit la représentation symbolique de chacune des instructions du système.

INSTRUCTION MACHINEREPRESENTATION SYMBOLIQUE

Jump to subroutine

JSR n,n,adr,adr,...
n+n adresses

Jump to subfunction

JSF n,n,adr,adr,adr,...
n+n adresses

Return from subroutine

RTS adr,...
0 à 3 adresses

Branch on condition

Bcn adr

Branch on index

BEX opr,adr

Unconditionnal jump

JMP adr

No operation

NCP

Halt

HLT

Compare characters

CCA A'c..c'
1 à 7 caractères

Compare character code

CCA* A'c'

Compare

CCC cde

Test indicator

CMF opr,opr

Test inclusion

TST opr

Test exclusion

TTN opr,opr

Compare labels

TTX opr,opr

Compare lists

CLB opr,opr

Compare trees

CLT opr,opr

Test parameter

CTR opr,opr

Clear

TPA cd,opr

Move

CIR opr

Addition

MOV opr,opr

Subtraction

APD opr,opr

Push stack

SUB opr,opr

PSH opr,...
1 ou 2 opérandes

Pop stack	POP <u>opr,...</u> 0 à 2 opérandes
Increment pointer	IPT nn,opr
Set pointer	SPT nn,opr,opr
Set indicator	SET opr
Clear indicator	CLI opr
Classify rule	CRL opr,opr
Add present parameters	APP opr
Clear present parameters	CPP opr
Test present parameters	TPP opr
Read character	RDC RDC+ RDC*
Write character	WRC A'c' WRC X'hh'
Print characters	PRT opr
Set printer mode	SPM lim,lim SPM* lim,lim SPM/ SPM+
Swap input output	SIO
Decode and output tree	DOT opr DOT* opr
Code binary	BIN opr
Code decimal	DEC opr DEC* opr
Code label	CDL
Code parameter	CDP opr

Les abréviations utilisées dans ce tableau sont définies à la fin de l'appendice.

B.5 Commandes assembleur.

Les commandes suivantes servent à contrôler le processus d'assemblage.

ORG - Set origin.

Cette commande est suivie d'une adresse hexadécimale et indique que l'instruction suivante doit débiter à cette adresse. Par exemple,

```
ORG  X'2FE'  
CLR  N
```

indique que le code de l'instruction CLR N doit débiter à l'adresse 02FE.

DC - Define constant.

Cette commande sert à définir des constantes et doit donc être suivie soit d'une constante décimale, d'une constante hexadécimale ou d'une suite de caractères. Par exemple,

```
      N : DC  X'FC'  
CHAINE : DC  A'LABEL'
```

affecte le nombre 00FC à l'adresse N et les caractères L, A, B, E et L, codés en ASCII, débutant à l'adresse CHAINE.

END - End of assembly.

Cette commande termine le processus d'assemblage.

B.6 Abréviations.

Les abréviations suivantes ont été utilisées dans la représentation symbolique des adresses ainsi que des instructions.

adr	:	adresse symbolique
c	:	caractère quelconque
cd	:	code entre 0 et 3 (2 bits)
cde	:	code entre 0 et 7 (3 bits)
cn	:	condition (EQ, LT, GT, NE, LE ou GE)
h	:	chiffre hexadécimal entre 0 et F
lim	:	limite (nombre décimal entre 0 et 127)
n	:	chiffre entre 0 et 3
nn	:	nombre décimal entre -128 et 127
nnn	:	nombre décimal entre -32768 et 32767
opr	:	adresse d'un opérande, de mode quelconque

BIBLIOGRAPHIE

- (1.) Berge, C. " Graphes et hypergraphes ", Monographies universitaires de mathématiques, DUNOD, 1970.
- (2.) Chu, Y. " Computer organization and microprogramming ", Prentice-Hall, 1972.
- (3.) Colmerauer, A. " Les SYSTEMES-Q ou un formalisme pour analyser et synthétiser des phrases sur ordinateur ", Université de Montréal, Publication interne no. 43, Sept. 1970.
- (4.) Eckhouse, R.H. Jr " Minicomputer systems: Organization and programming ", Prentice-Hall series in automatic computation, 1975.
- (5.) Horowitz, E. et Sahni, S. " Fundamentals of data structures ", Computer software engineering series, Computer Science Press, 1976.
- (6.) Husson, S.S. " Microprogramming: Principles and practices ", Prentice-Hall, 1970.
- (7.) Knuth, D.E. " The art of computer programming ", Vol. 1, " Fundamental algorithms ", Addison-Wesley, 1969.
- (8.) " Microprogramming handbook ", Second Edition, Microdata Corporation, 1972.

INDEX DES INSTRUCTIONS

Addition (ADD)	85
Add Present Parameter (APP)	97
BRAnch on condition (BRA)	63
Branch on inDeX (BDX)	64
Classify RuLe (CRL)	94
CLear (CLR)	83
CLear Indicator (CLI)	93
Clear Present Parameters (CPP)	98
code BINary (BIN)	110
code DECimal (DEC)	111
CoDe Label (CDL)	112
CoDe Parameter (CDP)	114
CoMPare (CMP)	73
Compare Character Code (CCC)	71
Compare ChAracters (CCA)	69
Compare LaBels (CLB)	77
Compare LiSTs (CLT)	78
Compare TRees (CTR)	79
Decode and Output Tree (DOT)	109
Increment PoiNter (IPT)	90

MOVe (MOV)	84
POP stack (POP)	88
PRinT characters (PRT)	104
PuSH stack (PSH)	87
ReaD Character (RDC)	101
SET indicator (SET)	92
Set PoinTer (SPT)	91
Set Printer Mode (SPM)	106
SUBtraction (SUB)	86
Swap Input Output (SIO)	108
Test eXclusion (TTX)	76
Test iNclusion (TTN)	75
TeST indicator (TST)	74
Test PARameter (TPA)	80
Test Present Parameters (TPP)	99
WRite Character (WRC)	103