

**LABORATOIRE D'INFORMATIQUE THEORIQUE  
& APPLICATIONS DE MARSEILLE**

**L.I.T.A.M.**

Faculté des sciences Economiques

**UNIVERSITE D'AIX-MARSEILLE II**

---

**ISSN 0291 - 5413**

|  |
|--|
| <b>INFORMATIQUE<br/>FONDAMENTALE<br/>&amp;<br/>APPLICATIONS</b>                |
| <b>Comité de<br/>réduction</b>   |
| <b>E. Bianco<br/>R. Cusin<br/>P. Isoardi<br/>J.P. Lehmann<br/>R. Stutzmann</b> |
| <b>Dépositaire<br/>B.U. Sc. Eco.<br/>Aix-Mars. II</b>                          |

**SOMMAIRE**

- P1... EDITORIAL  
PARADOXE.
- P6... SYSTEME UNIVERSEL.
- P16... Machine du modèle  
 $\alpha \rightarrow \beta$ ,  $x$  à décalage maximum.
- P34... YOUZZAYEDIBISAR

**MARS 1989**

Adresse postale : FACULTÉ DES SCIENCES ECONOMIQUES  
LITAM  
14 rue Puvis de CHAVANNES 13001 MARSEILLE  
91 90 13 20 P 420 et 421



## EDITORIAL

## PARADOXE

E. Bianco

Habitudes et inertie, frustrations et refoulements, les comportements sont-ils conditionnés par des compensations ?

D'abord et à la base de tout il y a l'idée de Génie, combien d'esprits qui auraient pu être simplement sympathiques ne se sont-ils pas torturés dans l'espoir à chaque instant démenti, d'être porteurs de l'étincelle.

Dans un monde faussement modeste où l'on ne supporte que les déclarations de médiocrité, l'expression de l'idée ne peut plus être que l'expression de la Parole de Dieu. Alors, seuls des êtres touchés par la grâce peuvent être reconnus comme porte-paroles. Tout n'est pas si simple, et pour être ainsi reconnu comme tel, il faut beaucoup de temps, suffisamment pour que l'impétrant soit mort depuis belle-lurette.

Aucun cerveau ne peut se développer hors du réseau social d'une multitude d'autres cerveaux, qui lui apportent formation et langage. Le génie ne serait plus alors que l'art de percevoir et d'exprimer des pensées en gestation dans la multitude. Les créateurs profitent des nouveaux matériaux pour imaginer des constructions dont l'intérêt n'apparaît qu'à partir du moment où on peut juger de son utilité. Ainsi

l'invention du virus informatique se révèle une saine réaction contre l'emprise centralisatrice d'un pouvoir boulimique et abêtissant.

Et puis vous avez les faux, ceux qui ont parfaitement compris la vanité de l'entreprise pour eux, mais aussi qu'il est beaucoup plus profitable de faire semblant. Et que ça passe très bien dans une organisation d'autant plus stable qu'elle fait essentiellement semblant. Par son impénétrabilité, la langue de bois est là pour montrer la profondeur et l'importance du Message.

On croit d'autant plus volontiers le Professeur Muzzarello qu'il arrive avec un matériel auréolé d'informatique et qu'il annonce que son avion va pouvoir renifler du pétrole. Même un président de la république se laisserait prendre à un tel piège. D'autant que les fonds engagés, quelle que soit l'issue de l'opération ne seront pas perdus pour tout le monde, même s'il n'en reste plus de traces comptables.

Contrairement au pétrole, l'argent n'a pas d'odeur, mais on se demande toutefois si l'avion n'était pas équipé davantage pour renifler l'argent.

C'est au sein de la société bourgeoise, liée au développement industriel qu'a dû également se

développer le perfectionnement des techniques pour alimenter la concurrence dans une économie de marché. De là est née la recherche qui a, comme la plupart des activités humaines largement échappé à ses objectifs d'origine. La systématique développée à son égard a servi aussi bien à la structurer elle-même qu'à l'étendre à pratiquement tous les domaines de la pensée. D'ailleurs, le mythe de la pensée pure convenablement plaqué sur le concept d'esprit issu du monde monothéiste en a bien facilité l'implantation de l'idée.

De nos jours la recherche se subdivise en deux parts inégales: la recherche de crédits, et l'autre. La recherche qui ramène des sous immédiatement et l'autre. Il faut des crédits pour rechercher des crédits, de plus en plus de crédits. L'autre, à côté de cela finit par apparaître un peu comme du temps perdu.

La Société révère l'être de génie dont le nom marque à jamais l'histoire de la découverte. Ce sont surtout les morts que l'on porte au pinacle, car plus aucune contradiction n'en est à craindre. Les Grands Prix sont exclusivement attribués à la Personnalité rayonnant le Symbole. Mais la personne seule non investie des Pouvoirs qui ne lui sont pas conférés, est objet négligeable face à l'Administration.

L'individu tout seul bénéficie d'une protection bien précaire, bardé de ses droits de l'Homme qui n'ont même pas de véritable statut juridique. Essayez de régler vous-même vos comptes, comme un grand, avec votre voisin qui vous empêche

régulièrement de dormir. Alors que colonel ou mieux encore général vous pourrez déporter, bousculer, hacher menu, n'apalmer à votre aise voire renverser le pouvoir en place, pour la plus grande gloire de la civilisation, la vraie, celle qui évangélise les sauvages.

La véritable question se pose à l'instant du choix de l'interlocuteur valable. Pourquoi voudriez-vous prendre au sérieux une personne qui ne parle qu'en son nom propre. Et qui en plus exprimerait peut-être des idées pour le moins bizarres. Tandis que si vous parlez au nom de la communauté, après avoir soigneusement mis au point un discours aseptisé qui ne froisse aucune personne de conséquence, alors là on est tout prêt à vous entendre.

La recherche c'est au fond un moyen d'expression individuel, dans la mesure où c'est la projection d'un regard neuf, personnel, sur l'univers. La recherche ne peut vraiment se développer que dans des conditions spécifiques de confiance et de calme. De telles conditions sont rares dans les grandes équipes où prédominent les contraintes de relation à l'administration, en échange des "moyens" fournis. Pénurie oblige.

C'est à propos de réflexions plus ou moins techniques que doivent se manifester d'importantes modifications dans les rapports qu'entretient l'individu avec le milieu ambiant. La recherche est essentiellement la recherche d'une personnalité, et cette démarche ne se fait pas, tant s'en faut, exclusivement dans des organisations

officielles. Mais quand elles peuvent survivre en tant que telles, ce sont les petites équipes qui conviennent le mieux à ce genre d'épanouissement, car l'établissement de rapports harmonieux y est encore possible.

Malheureusement les petites équipes sont superbement ignorées de l'administration dans leur incapacité de parler le bon langage responsable des grandes.

Peut-être est-il tout aussi difficile d'avoir à la fois la vocation d'enseigner à une jeunesse rétive, en bas dans les amphes et les labos, et de penser les grands problèmes avec une imagination de stratège, en haut, dans la Grande Administration.

L'âpre musique du langage vulgaire qui monte des profondeurs et trouble l'atmosphère par son odeur de sueur et sa couleur d'ongles sales, ne saurait parvenir jusqu'aux sphères éthérées de la pure organisation.

Car l'air des cimes a tôt fait de griser l'oiseau qui plane parmi les nuages, même s'il est de mauvais augure. Le ministre, même s'il fut jadis enseignant et contestataire ne saurait distraire la moindre seconde de son précieux temps pour venir voir sur place et interroger ceux qui travaillent et non plus les porte-paroles-porte-coton, trop occupé qu'il est à tartiner de sa vanité les écrans de télévision et autre médias à haut indice d'écoute.

En principe l'investigation de nature scientifique ne s'embarrasse pas de barrières, mais ce serait par trop négliger les modernes féodalités que de croire la pensée libre de s'ébattre. L'état devrait en principe jouer les régulateurs, en fait,

prendre en charge les problèmes fondamentaux dont l'objectif porte sur la décennie voire le demi-siècle, car le coût en est lourd et il paraît normal qu'il en soit supporté par la communauté la plus large.

Or, une telle influence de l'état est largement contrebalancée par l'existence des vrais grands centres de pouvoir. Les entreprises de distribution de l'énergie, pétrole, électricité, nucléaire font la loi dans le domaine de la chimie et de la physique, n'importe qui fait la loi en informatique. L'armée, comme une pieuvre à l'affût glisse ses tentacules partout.

Les formidables développements des industries qui traitent de l'énergie ont correspondu à un progrès quasi nul depuis Einstein dans les problèmes fondamentaux de la physique. Parallèlement les constructeurs d'ordinateurs annoncent périodiquement une nouvelle génération de machines révolutionnaire, alors que la nature du calculateur n'a pas évolué d'un iota depuis Turing.

On confond volontiers la cueillette d'une masse énorme d'informations avec la conception d'une théorie cohérente qui fait effectivement avancer la pensée.

On n'est même plus sûr que soit dénombrable l'ensemble des thèses de physique et d'informatique, financées par un employeur soucieux du lendemain proche, et qui se résument à la mise au point d'un programme à propos d'une application limitée.

Le soucis du profit à court terme, le manque certain d'une conception philosophique du monde

moderne, la faillite lamentable des idéologies qui rassemblaient utopie et espoirs, la prise de pouvoir effectif par la classe sociale des technocrates acculturés, voilà quelques causes qui ont des chances de n'être pas entièrement étrangères aux soubressauts économiques sociaux et écologiques de notre temps.

La courte vue de la survie immédiate talonnée par le fisc, autre citadelle de pouvoir quasi-indépendant, pousse à ramener la recherche à ses objectifs primitifs et tout-à-fait limités. Mais la complexification de la société exige de résister à cette tentation et de poursuivre l'effort fondamental.

Qu'on ne nous raconte pas que cinq cent mille tonnes de brut, en Alaska, ne peuvent que dépendre de la cravate d'un capitaine ivre, ou encore qu'un réservoir de lindane, non arrimé, doit son sort dans la Manche à une minute d'inattention.

Mais peut-être serait-il bon que le technocrate acquière rapidement une culture qui le replace dans le champ social, non comme une pièce de machine, son rôle actuel, mais comme un être humain conscient de la gravité de ses responsabilités. Cela me paraît le domaine de la recherche de loin le plus important. Aucune loi, aucun nouveau ministère, aucune commission ne peuvent remplacer cela.



**SYSTEME UNIVERSEL**

E. Bianco

C. R. Subject Classification Informatics : C53 C54 D31 D41

**Résumé.**

Nous avons déjà observé une vue partielle d'un système universel dans les articles précédents. On étudie ici le compilateur du langage HORS-TEXTE dont le rôle est de permettre de communiquer de manière universelle avec le système en lui demandant en gros deux sortes de services: le choix d'un sous-système pour pouvoir travailler, et l'enrichir d'un nouveau sous-système.



## SYSTEME UNIVERSEL

Dans l'article précédent je présentais la forme générale du système. Là, je vais achever de définir la partie initialisation et mise en place. Pour cela je vais apporter quelques légères et définitives modifications à la structure du VIIF. En effet quand le système démarre, il a tout d'abord à délimiter des images de tâches possibles. Ces tâches n'existent pas encore mais le système va tout de même commencer immédiatement à en explorer les images. Tout au début chaque tâche va se trouver sous le contrôle du compilateur Hors-Texte, c'est-à-dire qu'obligatoirement l'utilisateur devra demander un travail système. Il faut bien qu'il précise le sous-système qu'il veut utiliser, ou bien l'intégration d'un nouveau sous-système, etc.

A partir du moment où l'utilisateur a fourni le nom du sous-système dont il a besoin, le compilateur du HT crée le VIIF global qui lui correspond, et place la console de l'utilisateur sous contrôle du premier compilateur du sous-système. En conséquence immédiate il faut séparer le VIIF global en deux parties, la première que le système crée immédiatement au démarrage, et l'autre qui n'est créée qu'à la demande et qui disparaît quand l'utilisateur arrête son travail.

C'est un tableau de référence "pilote" qui sert à l'ouverture des

images de tâches. La file dynamique de références "TSS" et "DIS" sert à la construction du corps du VIIF.

### ECHANGES.

Les divers niveaux auxquels apparaissent les échanges se dégagent maintenant tout seuls. Le système a son propre échange pour le conversationnel à son niveau. Ensuite le conversationnel est mis à la disposition des compilateurs de sous-système, et ces compilateurs sont susceptibles d'émettre des demandes d'échanges avec les mémoires de masse.

Toujours dans le même esprit je vais essayer de compléter la construction du système. Dans le traitement du fini illimité, et en se plaçant dans un cadre ergonomique, on distingue plusieurs niveaux de calcul. Avec la gestion des sous-systèmes, et le contrôle des échanges qui leur est lié, le système assure la fonction de communication qui permet à l'utilisateur de choisir son service.

Il existe donc un véritable langage, le langage hors-texte, qui permet de choisir le sous-système dont on a besoin, et qui permet également d'enrichir le système de nouveaux sous-systèmes. Je fais abstraction de toutes les facilités qu'on pourrait aussi rajouter, mais qui n'apporteraient rien de plus à la validité du modèle que je présente.

L'objet de cet article porte donc sur le compilateur du langage hors-texte que je réduis aux deux sortes de phrases possibles:

sous-système <identificateur>;  
intégrer <identificateur>;

Je construis le compilateur de ce langage comme une section du système, bien entendu, il est lui

|             |   |                |           |                      |   |
|-------------|---|----------------|-----------|----------------------|---|
| [Phrase,pro | 0 | 1              | 2         | 2                    | 0 |
| lettre      |   | 'sous-système' | 'intégré' | toute lettre,chiffre | ; |
| [Phrase,id  | 0 | 1              | 1         |                      | 0 |
| lettre      |   | lettre latine  | chiffre   |                      | ; |

Je décris alors les contenus des files sur lesquelles travaille le compilateur du langage hors-texte. Le tableau dont les références sont:

libre,occupé  
a trois éléments par ligne et sert à gérer l'occupation de la file support

|   |
|---|
| ... [ $\omega_1$ , li1 , vi1 ] , [ $\omega_2$ , li2 , vi2 ] , [ $\lambda_3$ , , ] ... |
| caract                      caract                      caract                        |
| lieu vol                      lieu vol                      lieu vol                  |
| occupé                      libre   |

Rajouter une image de file revient à mettre dans l'élément "libre,lieu" la valeur "li2+vi2" qui indique la première place libre. Bien sûr il s'agit là d'un choix particulier de gestion de la place libre à remettre en cause dès qu'il faut réutiliser une place libérée. Mon propos se limite ici à montrer que la chose est possible. Le constructeur de système peut ainsi adapter à son problème la meilleure méthode de gestion.

aussi autojectif. L'analyse se fait à travers deux variables d'état matérialisées dans une ligne de la file des images des terminaux, de références

Phrase,pro  
Phrase,id

et Phrase,er sert à noter l'erreur. Le tableau des valeurs en est le suivant:

par le système. Le premier élément contient  $\omega$  ou  $\lambda$  selon que l'image réelle de file qui y est notée est soit occupée, soit libre. La référence 'occupé' repère le dernier élément noté et 'libre' le premier élément disponible.

Je vais supposer que le descriptif définitif d'un sous-système est enregistré en deux files, l'une qui contient par ligne des triplets constitués d'un identificateur à longueur quelconque et de deux entiers, l'autre qui contient la liste des descriptifs de toutes les files de tous les compilateurs qui constituent le sous-système.

Je fais en outre une hypothèse à propos de la place occupée par les valeurs des variables de référence,

des index, des variables de répétition, les prédicats, la commutation de coupure. Je considère qu'à la liste des files locales d'un compilateur s'ajoutent systématiquement quatre files dont le volume est calculé pour

pouvoir contenir toutes les valeurs de ces variables rencontrées par le compilateur du PFS dans contexte et texte du compilateur traité.

La file dynamique en "LISTE, FLISTE et CLIST se représente ainsi:

---

|   |                                       |
|---|---------------------------------------|
| {nom SSYST, accès grille, Nbcomp(k)}, {nom 1er comp, N° SSYST, N° 1er LISTE | {nom 1er comp, N° SSYST, N° 1er CLIST |
|---|---------------------------------------|

---

comp}, {nom C1, p1, p'1}, {nom C2, p2, p'2}, ..... , {nom Ck, pk, p'k}

---

Le tableau en GRILLE,GRILLEC contient les descriptifs de toutes les files du sous-système, chacun des descriptifs contient en fait un statut global de la file, le mode, le type, localisation-volume pour le directory

dans le cas d'une file dynamique ou support, et rien pour un tableau, localisation-volume pour le corps de la file, enfin le numéro de la file si elle est externe.

---

|   |  |
|---|--|
| L11, L12, ..... , L1q1, (Lf), L21, L22, ... L2q2, (Lf), L31, ... , L3q3, (Lf), ..... , GRILLE | L11, L12, ..... , L1q1, (Lf), L21, L22, ... L2q2, (Lf), L31, ... , L3q3, (Lf), ..... , GRILLEC |
|---|--|

---

Lk1, Lk2, .... , Lkqk, (Lf), Le1, Le2, .... , Lez, (Lf), LRF.

---

Les q1, q2, ... ,qk sont au moins égaux à 4 à cause des files destinées à contenir les valeurs des variables-nom. Les Lij sont les images des files locales à chaque compilateur, les Lem sont les images des files externes, et LRF contient l'indication du maximum de descriptifs à réserver pour le transfert de l'information de l'instruction d'arrêt à l'insertion du compilateur suivant.

|     |              |
|-----|--------------|
| mij | 'définitif'  |
|     | 'provisoire' |
|     | 'calculé'    |
| tij | 'tableau'    |
|     | 'dynamique'  |
|     | 'support'    |

Les descriptifs Lf et LRF ont les contenus suivants:

Lf = 'fin', - , - , - , - , - , - .

LRF = - , - , - , Nb RF, - , - , - .

$Lij = mij, tij, \lambda ij, vij, \lambda'ij, v'ij, nij$

C'est à partir des deux précédentes files que je vais constituer le VIIF global qui porte sur deux files à la fois. L'une,

tableau, a pour références PILOTE, PIL, l'autre, dynamique, a pour références TSS, DIS, DISC, DISF.

---

{Enum tâche, Cnssyst, Cn , Cnb comp, Cnum comp, ---, --- , Caccès}  
 NTERM NSS ETACH k NCOMP ECH ETECH AX  
 PIL

---

{Cnss, Cn tâche, Ck , Cn° comp, Cn term}, {VIIF1}, {VIIF2}, ..... , {VIIFk}, < pool >.  
 1 2 3 4 5  
 DIS DIS DISC POOL

---

Le pool se détaille ainsi:

---

{CL11}, {CL12}, ... {CLez}, {CLRF1}, {CLRF2}, ... , {CLRFp}  
 POOL POOLC

---

Je rappelle la liste des files déclarées en-tête du M-CONTEXTE. Il manque encore celle qui sert au compilateur du PFS pour noter les caractéristiques des files du

compilateur traité. C'est à partir de cette file qu'on peut constituer les deux files qui sont utilisées ici et référées en GRILLE et LISTE.

---

100.000 , 10.000 dynamique (par, Fc, Nr ), 1 ( 3(2), (1), 4(2), 5(1), 3(2), (3), (2) )  
 , 2 ( files(2) ) , 3 ( 5(2) ) , 5 ( 2(1), 3(2) ) : TSS : DIS : DISC : DISF :  
 DISK : POOL : POOLC : POOLF ;

1.000 tableau ( NTERM(2), NSS(2), ETACH(1), k(2), NCOMP(2), ECH(1), ETECH(1), AX(2) ) : PILOTE : PIL ;

10.000 tableau ( caract(1), lieu(2), vol(2) ) : Libre : occupé : Libres ;

1 tableau ( dispo(2), entrée(2) ) : réserve;

1.000.000 support ( ) : P : Supplib : Conf ;

5 : 50 cartouche ( etat(1), sens(1), type(1), Nfil(2), Nelem(2), Nlog1(2), Nlog2(3), Vol(2) ) : carte : carts : cartc ;

50 tableau ( type1(1), numéro(2) ) : cartlib : cartocc : cartocc2 ;

1 : 1 tableau ( etat(1), Nombre(2), Nomb1(2) ) : calcul ;

2 : valeur tableau ( introduisez le nombre de terminaux à lancer ) : message ;

3 : 100 tableau ( lu(1), report(1), pro(1), id(1), er(1) ) : Phrase ;

4 : 100 cartouche ( etsys(1), sensys(1), typsys(1), Nfsys(2), Nelsys(2), logsys1(2), logsys2(3), volsys(2) ) : lecture : ecriture ;

1.000 , 30 dynamique ( ssnom, accès ) , 1 ( Nblet(1), 2(2) ) : LISTE : FLIST : CLIST

10.000 tableau ( mod(1), typ(2), rien1(2), dir(2), rien2(2), volume(2), Num(2) ) :  
GRILLE : GRILLEC ;

2.000 , 100 dynamique (let) , 1 ( nlet(1) ) : Nom ;

---

Je construis ici le compilateur du langage HORS\_TEXTE , et seulement la partie qui traite de la mise en place du YIIF global pour une tâche. Je compléterai ultérieurement l'intégration d'un nouveau sous-système, travail accompli par la section CREATION .

section HORSTEXTE ;  
 Nom := initial ;  
 let := 1 ;

aiguillage LETTRE ( [Phrase,report = ( "soussyst" : APPEL , "intégré" : INTEG , "A" : ident , "B" : ident , "C" : ident , ... "Z" : ident , "O" : ident , "1" : ident , ... , "9" : ident , ";" : misenplace ) ) ;

APPEL : aiguillage ER ( [Phrase,pro ≠ ( 0 : Erreur , sinon Analyse ) ) ;  
 Erreur : [Phrase,er := 1 ;  
sortie Erreur ;

Analyse : [Phrase,pro := 1 ;  
sortie Analyse ;

nœud ER ;

sortie APPEL ;

INTEG : aiguillage ER1 ( [Phrase,pro ≠ ( 0 : Err1 , sinon Analyse1 ) ) ;  
 Err1 : [Phrase,er := 2 ;

sortie Err1 ;

Analyse1 : [Phrase,pro := 2 ;  
sortie Analyse ;

nœud ER1 ;

sortie INTEG ;

IDENT : [nom,let := [Phrase,report ;  
let := let + 1 ;

sortie IDENT ;

misenplace : aiguillage CHOIXHT ( [Phrase,pro = ( 1 : placessyst , 2 : créessyst) )

placessyst : nlet := let - 1 ;  
ssnom := 1 ;

itére RECHSSYST ( LISTE de initial à FLIST , cond [nom = [LISTE,ssnom ) ;  
répétition RECHSSYST ( vrai : Trouvé , faux : Suite ) ;

Suite : [Phrase,er := 3 ;  
sortie Suite ;

Trouvé : Accès := ssnom + 1 ;  
Grille := [LISTE,Accès → initial ;  
CLIST := → LISTE ;  
GRILLEC := GRILLE ;

insérer IMPLANT ;

TSS := → POOLF ;  
sortie Trouvé ;

nœud RECHSSYST ;

sortie placessyst ;

créessyst : insérer CREATION ;  
sortie créessyst ;

nœud CHOIXHT ;

sortie misenplace ;

nœud LETTRE ;

fin section HORSTEXTE ;

section IMPLANT ;

[PIL,NSS := [CLIST,2 ;

[PIL,ETACH := "ssyst" ;

[PIL,k := [LISTE,3 ;

[PIL,NCOMP := [CLIST,1 ;

[PIL,AX := rang TSS ;

La section IMPLANT crée le VIIF global et en même temps l'image de la tâche en construction. Or, cette image est faite pour être explorée lors de la rotation de distribution du système.

TSS := [LISTE,2 → initial ;

ligne TSS , 3 ;

[TSS,1 := [CLIST,2 ;

[TSS,2 := "ssyst" ;

[TSS,3 := [LISTE,3 ;

[TSS,4 := [CLIST,3 ;

[TSS,5 := [PIL,1 ;

DIS := → TSS ;

POOL := [LISTE,3 → DIS ;

DISF := [TSS,3 → TSS ;

itère CreatVIIF ( DISC de DIS à DISF ) ;

ligne DISC,1 ;

répéter CreatVIIF ;

Création des lignes de chaque VIIF local.

[calcul,Nomb1 := [LISTE,3 + 1 ;

CLIST := → CLIST ;

insérer TRANSFERT ;

itère Grand ( DISC de DIS à DISF ) ;

[DISC,5 := [calcul,Nomb1 ;

répétition Grand ;

Mise en place en une boucle de la valeur unique D d'accès aux descripteurs des files externes.

insérer TRANSFILES ;

[POOL,1 := "fin" ;

GRILLEC := → GRILLEC ;

```

POOL := → POOL ;
[calcul,Nomb1 := [calcul,Nomb1 + 1 ;

itère SurFR ( DISK de DIS à DISF ) ;
    [DISK,7 := [calcul,Nomb1 ;
répéter SurFR ;

POOLF := [GRILLEC,4 → POOL ;

itère Resplace ( POOLC de POOL à POOLF ) ;
    ligne POOLC , 5 ;
répéter Resplace ;

    fin section IMPLANT ;

```

Mise en place en chaque VIIF local de la valeur FR unique d'accès aux descriptifs de transfert d'arrêt.

```

    section TRANSFERT ;
itère VIIF ( DISC de DIS à DISF ) ;
    [DISC,1 := [CLIST,2 ;
    [DISC,2 := [CLIST,3 ;
    [DISC,3 := [calcul,Nomb1 ;
    [calcul,Nom1 := [calcul,Nomb1 + 1 ;

```

La section TRANSFERT charge la liste des VIIF locaux avec les valeurs de localisation des accès aux files, qui se trouvent dans le pool.

Dans le pool on trouve la suite des listes de descriptifs de files locales, suivie des descriptifs des files externes, donc communes à tous les compilateurs, enfin suivie des descriptifs de transfert de l'instruction d'arrêt.

```

    CLIST := → CLIST ;

insérer TRANSFILES ;

    [DISC,6 := [calcul,Nomb1 - 4 ;
    GRILLEC := → GRILLEC ;

```

```

répétition VIIF ;

    fin section TRANSFERT ;

```

```

    section TRANSFILES ;
itère FILES ( cond [GRILLEC,1 = "fin" ) ;

```

```

    aiguillage MODEF ( [GRILLEC,1 = ("calculé" : COPIE , "définitif" : Attribut ,
        "provisoire" : Attribut ) ) ;
COPIE : [POOL := [GRILLEC ;
    sortie COPIE ;

```



Attribut : ligne POOL , 5 ;  
aiguillage TypeF ( [GRILLEC,2 = ( "dyn" : Dynam , "tab" : TAB ) ) ;

Dynam : [libre,lieu := [occupé,lieu + [occupé,vol ;

[libre,vol := [GRILLEC,DIR ;

[libre,caract := "ø" ;

[POOL,3 := [libre,lieu ;

[POOL,4 := [GRILLEC,DIR ;

occupé := libre ;

libre := → libre ;

sortie Dynam ;

La section TRANSFILES copie l'image des files notées en "GRILLE", dans le VIIF, en même temps, elle réserve une place en file support pour les files à volume déclaré. Pour les autres, le type est dupliqué, et le système devra effectuer la réservation pendant le calcul.

TAB : sortie TAB ;

nœud TypeF ;

[libre,lieu := [occupé,lieu + [occupé,vol ;

[libre,vol := [GRILLEC,Volum ;

[POOL,5 := [libre,lieu ;

[POOL,6 := [GRILLEC,Volum ;

[libre,caract := "ø" ;

occupé := libre ;

libre := → libre ;

[POOL,1 := [GRILLEC,1 ;

[POOL,2 := [GRILLEC,2 ;

[POOL,7 := [GRILLEC,Num ;

sortie Attribut ;

nœud MODEF ;

POOL := → POOL ;

GRILLEC := → GRILLEC ;

[calcul,Nomb1 := [calcul,Nomb1 + 1 ;

répétition FILES ;

fin section TRANSFILES ;

Reste à compléter la section CREATION dont le rôle consiste à créer un chapitre supplémentaire dans la liste des sous-systèmes qui appartient au système. Je pars de l'hypothèse que le système dispose d'une file dans laquelle le compilateur du PFS a noté les informations suffisantes glanées au bon moment.

En fait, en tête de chaque compilateur on déclare deux noms, celui du compilateur lui-même, et celui du sous-système auquel il

appartient. Le compilateur y note également la description des files rencontrées dans le contexte, puis il rajoute quatre pseudo-files destinées à contenir les images matérielles de toutes les autres variables: les références, les index, les variables de répétition, et tous les prédicats des diverses formes de commutation. Les accès au code image du texte y seront également notés sous leurs deux aspects: accès à la section INITIALE et accès au corps du programme.



**MACHINE DU MODELE  $\alpha \rightarrow \beta, x$   
A DECALAGE MAXIMUM**

Omar Kettani

C. R. Subject Classification Informatics D31

**Résumé.**

Dans le cas où la tête de lecture d'une machine du genre des machines de Turing, est capable de chevaucher deux cases à la fois, on est amenés à faire évoluer la proportion de chevauchement des cases. En passant à la limite du possible la tête va lire le maximum du contenu d'une case et le minimum de l'autre. Ce minimum ne peut jamais aller au delà de deux valeurs différentes. D'où l'utilisation du binaire privilégiée pour représenter les contenus de cases. Le minimum commun se limite alors à un seul bit binaire. Cet article montre comment ceci suffit pour faire voyager toute l'information voulue lors des déplacements de la tête de lecture.

## MACHINE DU MODELE $\alpha \rightarrow \beta, X$ À DECALAGE MAXIMUM

### 1 INTRODUCTION

Il a déjà été prouvé que le modèle  $\alpha \rightarrow \beta, X$  à décalage quasi-maximum, c'est à dire pour lequel l'intersection des espaces sous l'emprise du bloc de traitement (BT) entre deux instants consécutifs est de deux bits, possède la puissance du calculable [1]. La question qui se pose alors est celle de savoir si un décalage maximum, c'est à dire pour lequel l'intersection des espaces sous l'emprise du bloc de traitement entre deux instants consécutifs est d'un bit, peut être envisageable sans perte de puissance de calcul. Dans ce qui suit, on va pouvoir répondre par l'affirmative à cette question. Pour cela, on va donner deux démonstrations permettant d'établir, indépendamment les unes des autres, ce résultat.

La première, dite autonome car elle ne fait intervenir aucune autre définition du calculable, montre que les machines du modèle  $\alpha \rightarrow \beta, X$  définies dans le cadre du "demi-mouvement" peuvent être décrites et représentées par un modèle  $\alpha \rightarrow \beta, X$  à décalage maximum (un bit de chevauchement). Par ailleurs, il a déjà été prouvé [2] que les machines du modèle  $\alpha \rightarrow \beta, X$  définies dans le cadre du "demi-mouvement" sont équivalentes aux machines de Turing et ont par conséquent la puissance du calculable.

La deuxième démonstration, utilisant le fait qu'une machine de Turing quelconque a la même puissance qu'une machine de Turing à 2 états [3], permet d'établir que toute machine de Turing à 2 états est équivalente à une machine du modèle  $\alpha \rightarrow \beta, X$  à décalage maximum.

### 2 DEMONSTRATION D'EQUIVALENCE ENTRE LE MODELE $\alpha \rightarrow \beta, X$ À DEMI-DECALAGE ET LE MODELE $\alpha \rightarrow \beta, X$ À DECALAGE MAXIMUM

L'intérêt de cette démonstration, interne au modèle, est qu'elle ne fait appel à aucune autre définition du calculable.

Soit une machine du modèle  $\alpha \rightarrow \beta, x$  définie dans le cadre du "demi-mouvement". Elle est caractérisée par un ensemble fini de règles du type:

$$a_1 a_2 \dots a_n b_1 b_2 \dots b_n \rightarrow a'_1 a'_2 \dots a'_n b'_1 b'_2 \dots b'_n x$$

où  $x$  qui vaut  $g$  ou  $d$  selon que le déplacement commandé est de  $n$  bits à gauche ou à droite. Chaque lettre est un bit.

Supposons alors que la position courante à l'instant  $t$  de la machine soit celle-ci:

$$\overline{d_1 d_2 \dots d_n \boxed{a_1 a_2 \dots a_n b_1 b_2 \dots b_n} g_2 \dots g_n}$$

Où  $\boxed{\phantom{a_1 a_2 \dots a_n b_1 b_2 \dots b_n}}$  désigne l'emprise du bloc de traitement.

et que la transformation à opérer soit:

$$ab \rightarrow a'b', d$$

où on a posé:

$$a = a_1 a_2 \dots a_n$$

$$b = b_1 b_2 \dots b_n$$

$$a' = a'_1 a'_2 \dots a'_n$$

$$b' = b'_1 b'_2 \dots b'_n$$

à l'instant  $t+1$  elle sera :

$$\overline{d_1 d_2 \dots d_n a'_1 a'_2 \dots a'_n \boxed{b'_1 b'_2 \dots b'_n} g_1 \dots g_n}$$

Supposons alors que la configuration courante à l'instant  $t$  de la machine soit celle-ci, qui correspond à la situation initiale I:

$$\overline{\dots \Delta a_{i_1} a_{i_2} \dots \boxed{a_{i_p} a_{i_{p+1}}} a_{i_{p+2}} \dots a_{i_k} \Delta \dots}$$

et que la transformation à opérer soit:

$$a_{i_p} a_{i_{p+1}} \rightarrow a'_{i_p} a'_{i_{p+1}} d$$

Alors à l'instant  $t+1$  on doit aboutir à la situation II:

$$\dots \Delta a_{i_1} a_{i_2} \dots a'_{i_p} \boxed{a'_{i_{p+1}} a_{i_{p+2}}} \dots a_{i_k} \Delta \dots$$

On représente la situation courante ainsi dans la machine à décalage maximum :

$$\dots \Delta c e a_{i_1} c e a_{i_2} \dots c e a_{i_3} \dots c e a_{i_{p-2}} c e a_{i_{p-1}} \boxed{1 c e a_{i_p} 1 c e a_{i_{p+1}} 1} c e a_{i_{p+2}} 1 c e a_{i_{p+3}} 1 c e \dots 1 c e a_{i_{k-1}} 1 c e a_{i_k} 1 c e \Delta \dots$$

où  $c=0 \dots 0$  n fois,  
 $e=00000$   
 $e_0=00001$

On a donc séparé, pour des raisons qui apparaîtront au fur et à mesure de la démonstration, les symboles de la configuration courante de la machine à simuler par la séquence  $cce$  pour ceux situés à gauche de l'emprise du bloc de traitement, et par la séquence  $1ce_0$  pour ceux se trouvant sous l'emprise du bloc de traitement enfin par la séquence  $1ce$  pour ceux situés à droite de l'emprise du bloc de traitement.

Alors que l'emprise du bloc de traitement sur la machine à "demi-mouvement" se faisait sur une suite de  $2n$  bits, ici, la même suite est représentée par  $1+n+5+n+1+n+5+n+1=4n+13$  bits.

Le but est de pouvoir aboutir en un temps fini, par un ensemble fini de règles à la situation finale correspondant à II:

$$\dots \Delta c e a_{i_1} c e a_{i_2} c e a_{i_3} \dots c e a_{i_{p-1}} c e a_{i_p} \boxed{1 c e a'_{i_{p+1}} 1 c e a_{i_{p+2}} 1} c e a_{i_{p+3}} 1 c e \dots 1 c e a_{i_{k-1}} 1 c e a_{i_k} 1 c e \Delta \dots$$

### 2.1 LA METHODE UTILISEE

Elle va consister à créer une succession de situations qui vont permettre en entrée de phase de calcul, d'opérer la transformation requise, puis de décaler vers la gauche (respectivement vers la droite pour le cas où la règle à opérer est  $a_{i_p} a_{i_{p+1}} \rightarrow a'_{i_p} a'_{i_{p+1}}$  g) toute la configuration d'un symbole, tout en indiquant la nouvelle position du bloc de traitement. En entrée de phase on a:

$$\dots \Delta c e a_{i_1} c e a_{i_2} \dots c e a_{i_3} \dots c e a_{i_{p-2}} c e a_{i_{p-1}} \boxed{1 c e a_{i_p} 1 c e a_{i_{p+1}} 1} c e a_{i_{p+2}} 1 c e a_{i_{p+3}} 1 c e \dots 1 c e a_{i_{k-1}} 1 c e a_{i_k} 1 c e \Delta \dots$$

En fin de phase on a (si la règle à opérer est  $e_i e_{i_p+1} \rightarrow a'_i a'_{i_p+1} d$ ):

---

$\Delta 0 e a_{i_1} 0 e a_{i_2} 0 e a_{i_3} \dots \dots \dots 0 e a_{i_{p-1}} 0 e a_{i_p} \boxed{1 e a_{i_p+1} 1 e a_{i_p+2} 1} e a_{i_p+3} 1 e e \dots \dots \dots 1 e a_{i_k-1} 1 e a_{i_k} 1 e \Delta \dots$

---

Toute la difficulté réside dans le fait qu'on ne dispose, à chaque pas de calcul, que d'une quantité minimale (1 bit) pour transporter l'information d'une zone à l'autre de la configuration marquée par un symbole  $e_i$  ou  $E_i$  variant de 1 à 8. Si bien qu'on a recours à des compteurs pour effectuer progressivement de telles opérations. Les règles à appliquer doivent toutes avoir des parties gauches différentes, pour éviter qu'il y ait ambiguïté au moment de leur application.

Afin de pouvoir donc construire un ensemble de règles univoque, c'est à dire tel qu'à chaque pas de calcul, une et une seule règle soit applicable, le vecteur  $e$  prendra plusieurs valeurs distinctes qu'on indexera  $e_0, e_1, e_2, \dots, e_8, E_1, \dots, E_8$  caractérisant les différents états des zones mémoires où se trouvent les symboles de la configuration. On notera l'analogie avec la méthode utilisée par Shannon [19] pour ramener toute machine de Turing à une machine de Turing à deux états.

## 2.2 LES NOTATIONS

$e$  caractérise un état courant.

$e_0$  représente la zone sous l'emprise du bloc de traitement en entrée de phase de calcul.

$e_1$  représente la zone sous l'emprise du bloc de traitement après transformation par la règle applicable.

$e_2$  caractérise l'état courant de la partie droite de la configuration par rapport au bloc de traitement après le passage de celui-ci vers la droite.

$e_3$  représente un état de "réception" (celui du symbole droit de la zone immédiatement à gauche).

$e_4$  représente un état de "transmission" (celui du symbole droit de la zone immédiatement à droite de celle marquée par  $e_3$ ).

$e_5$  représente la zone sous l'emprise du bloc de traitement lorsque celui-ci retourne vers la gauche.

$e_6$  représente un état de "réception" analogue à  $e_3$ , mais valable pour les symboles de la configuration se trouvant à gauche de l'emprise du bloc de traitement.

$e_7$  représente un état de "transmission" analogue à  $e_4$ , mais valable pour les symboles de la configuration se trouvant à gauche de l'emprise du bloc de traitement.

$e_8$  caractérise l'état courant de la partie gauche de la configuration par rapport au bloc de traitement après le passage de celui-ci vers la gauche.



De même,  $E_1, \dots, E_6$ , vont jouer des rôles analogues pour le cas où la règle applicable est  $a_i a_{i+1} \rightarrow a'_i a'_{i+1} \xi$

$E_1$  représente la zone sous l'emprise du bloc de traitement après transformation par la règle applicable.

$E_2$  caractérise l'état courant de la partie gauche de la configuration par rapport au bloc de traitement après le passage de celui-ci vers la gauche.

$E_3$  représente un état de "réception" (celui du symbole droit de la zone immédiatement à droite).

$E_4$  représente un état de "transmission" (celui du symbole droit de la zone immédiatement à droite de celle marquée par  $E_3$ ).

$E_5$  représente la zone sous l'emprise du bloc de traitement lorsque celui-ci retourne vers la droite.

$E_6$  représente un état de "réception" analogue à  $e_3$ , mais valable pour les symboles de la configuration se trouvant à droite de l'emprise du bloc de traitement.

$E_7$  représente un état de "transmission" analogue à  $E_4$ , mais valable pour les symboles de la configuration se trouvant à droite de l'emprise du bloc de traitement.

$E_8$  caractérise l'état courant de la partie droite de la configuration par rapport au bloc de traitement après le passage de celui-ci vers la droite.

## 2.3 LA DEMONSTRATION

### 2.3.1 LE MOUVEMENT A DROITE

On va maintenant décrire l'ensemble des règles qui vont permettre d'appliquer la méthode:

La première  $r_1$  va consister à opérer la transformation  $xy \rightarrow x'y'$  puis à marquer le lieu sous l'emprise du bloc de traitement par  $e_1$  puis à amorcer la progression du bloc de traitement vers la droite en effectuant un déplacement maximum par rapport à sa position précédente : c'est à dire un mouvement  $d=4n+12$  bits:

$$r_1 : 1 c e_0 x 1 c e_0 y 1 \rightarrow 1 c e_1 x' 1 c e_1 y' 1, d$$

La deuxième règle  $r_2$  aura pour effet de faire se continuer la progression du bloc de traitement vers la droite tout en marquant cette partie de la configuration par  $e_2$ :

$$r_2 : 1 c e x 1 c e y 1 \rightarrow 1 c e_2 x 1 c e_2 y 1, d$$

(Dans tout ce qui suit,  $x$  et  $y$  désignent des éléments de l'alphabet de la machine à simuler et on appellera le schéma de règle correspondant plus brièvement la "règle").

La réitération de la règle  $r_2$  fait aboutir le BT à l'extrémité droite de la configuration caractérisée par la situation particulière:  $1 c e x 1 c e \Delta 1$  ( $x=\Delta$  possible suivant la parité du cardinal de la configuration courante)

A ce stade la configuration courante se présente ainsi:

$$\dots \Delta 0 c e a_1 0 c e a_2 \dots 0 c e a_3 \dots \dots 0 c e a_{p-2} 0 c e a_{p-1} 1 c e a_p 1 c e a_{p+1} 1 c e 2 a_{p+2} 1 c e 2 a_{p+3} 1 c e 2 \dots 1 c e 2 a_{k-1} \boxed{1 c e a_k 1 c e \Delta 1} \dots$$

où on applique alors la règle  $r_3$  qui aura pour effet de décaler d'un "cran" vers la droite le symbole  $x$ ; de lui substituer le vecteur  $c$  qui aura pour but de "recevoir" le symbole situé à gauche de la séquence  $1 c e_3$  puis à marquer cette zone par  $e_3$  qui est caractéristique d'un état de réception avant d'effectuer un mouvement  $g=-(4n+12)$  vers la gauche:

$$r_3 : 1 c e x 1 c e \Delta 1 \rightarrow 1 c e_3 c 1 c e_3 x 1, g$$

On découvre alors pour la première fois la situation:  $1 c e_2 x 1 c e_2 y 1$ .

La règle  $r_4$  aura alors pour but de changer "l'état"  $e_2$  en  $e_4$  caractéristique d'un "état de transmission" (celle du vecteur  $y=y^2=y_1 \dots y_n$  vers la droite). Cette "transmission" sera contrôlée par un compteur  $c^i=1 \dots 1, 0 \dots 0$  contenant  $i$  fois 1 et  $n-i$  fois 0, avant d'effectuer un mouvement vers la droite:

$$r_4 : 1 c e_2 x 1 c e_2 y 1 \rightarrow 1 c e_4 x 1 c^i e_4 1 y^2, d$$

La règle  $r_5$  aura pour effet de recevoir la  $n$ ème composante  $y_n$  du vecteur  $y$ , l'incrémement du compteur  $c$  puis permettra d'effectuer de nouveau le mouvement  $g=-(4n+12)$  vers la gauche:

$$r_5 : y_n c e_3 c 1 c e_3 z 1 \rightarrow 1 c e_3 y_n c_{n-1} 1 c^i e_3 z 1, g$$

où  $c_{n-1}$  est le vecteur à  $n-1$  composantes nulles.

Ceci amorce un processus de transport du vecteur  $y$  vers la droite, processus contrôlé par le compteur  $c^i$ . On notera l'analogie avec la méthode utilisée par Shannon pour démontrer que toute machine de Turing peut être réduite à une machine de Turing à 2 états équivalente.

Le processus se poursuit par la mise en œuvre des règles:

$$r_{5,i} : 1 c e_4 x 1 c^i e_4 1 c_i y^{2i+1} 1 \rightarrow 1 c e_4 x 1 c^{i+1} e_4 1 c_{i+1} y^{2i}, d$$

où  $y^i = y_1 \dots y_i$  et  $y^{2i}$  est le vecteur  $y^{2i}$  "privé" de sa dernière composante.

$$\text{et } r_{6,i} : y_{2i} c e_3 y_{2i+1} \dots y_{2i} c_i 1 c^i e_3 z 1 \rightarrow 1 c e_3 y_{2i} \dots y_{2i} c_{i-1} 1 c^{i+1} e_3 z 1, g$$

Le processus s'achève par les règles suivantes:

$$r_7 : 1 c e_4 x 1 c^2 e_4 1 c_2 1 \rightarrow 1 c e_4 x 1 c^2 e_4 1 c_2 0, d$$

où le dernier bit 0 transformé sert à indiquer qu'il ne s'agit pas de la position caractéristique de l'emprise du bloc de traitement.

et

$$r_8 : 0 c e_3 y_1 \dots y_{2i} c_2 1 c^2 e_3 z 1 \rightarrow 0 c e_3 y_1 c e_3 z 1, g$$

La règle  $r_8$  remet en état courant ( substitution de  $e_3$  par  $e$ ; compteur  $c_2$  remis à zéro i.e.  $c$  ) la zone de "réception" afin de préparer la prochaine phase de calcul. Après un mouvement à gauche, on aboutit pour la première fois à la situation suivante  $1 c e_4 x 1 c^2 e_4 1 c_2 0$ , où la zone marquée par  $e_4$  doit jouer le rôle zone de réception (caractérisée par  $e_3$ ) : c'est ce que fait la règle  $r_9$  suivante qui agit de manière analogue à  $r_5$ :

$$r_9 : 1 c e_4 x 1 c^2 e_4 1 c_2 0 \rightarrow 1 c e_3 c 1 c e_3 x 1, g$$

Puis on reboucle sur  $r_4$ , jusqu'à aboutir au lieu marqué  $1 c e_1 x 1 c e_1 y 1$ , caractérisant la position du bloc de traitement de la machine à simuler; alors on applique la règle  $r_{10}$  qui agit de manière analogue à  $r_4$  où  $e_5$  va jouer le rôle de  $e_4$  sauf en  $r_7$ :

$$r_{10} : 1 c e_1 x 1 c e_1 y 1 \rightarrow 1 c e_5 x 1 c^1 e_5 1 y^2, d$$

Il faudra donc prévoir en plus les règles suivantes:

$$r'_{5,i} : 1 c e_5 x 1 c^i e_5 1 c_i y^{2i+1} 1 \rightarrow 1 c e_5 x 1 c^{i+1} e_5 1 c_{i+1} y^{2i}, d$$

analogue à  $r_{5,i}$  ( où  $e_5$  remplace  $e_4$  ),

puis

$$r'_7 : 1 c e_5 x 1 c^2 e_5 1 c_2 1 \rightarrow c e_5 x 1 c^2 e_5 1 c_2 1, d$$

analogue à  $r_7$  (où  $e_5$  remplace  $e_4$  à ceci près qu'on transmet le bit 1 à droite au lieu de 0 pour indiquer qu'il faudra décaler vers la droite la marque  $e_0$  caractérisant la position du bloc de traitement de la machine à simuler). C'est ce que fait la règle suivante, qui agit de manière analogue à  $r_8$ :

$$r'_8 : 1 c e_3 y_1 \dots y_n c_n 1 c^2 e_3 z 1 \rightarrow 1 c e_0 y c 1 c e_0 z 1, g$$

Après un mouvement à gauche, on aboutit pour la première fois à la situation:

$$1 c e_5 x 1 c^2 e_5 1 c_n 1$$

où la zone marquée par  $e_5$  doit jouer le rôle zone de réception (caractérisée par  $e_6$ ) : c'est ce que fait la règle  $r'_9$  suivante qui agit de manière analogue à  $r_9$ , à ceci près que  $e_3$  est remplacé par  $e_6$  qui joue le même rôle (celui de réception) et que les bits 1 des séquences  $1 c e$  sont remplacés par 0 pour mettre en évidence qu'on traite la partie de la configuration se trouvant à gauche du bloc de traitement de la machine à simuler :

$$r'_9 : 1 c e_5 x 1 c^2 e_5 1 c_n 1 \rightarrow 1 c e_6 c 0 c e_6 x 0, g$$

Après un mouvement à gauche, on aboutit pour la première fois à la situation  $0 c e x 0 c e c 1$ , où on applique la règle  $r'_{10}$  qui agit de manière analogue à  $r_{10}$  où  $e_7$  va jouer le rôle de  $e_4$  :

$$r'_{10} : 0 c e x 0 c e y 1 \rightarrow 0 c e_7 x 0 c^1 e_7 0 y^2, d$$

Il faudra donc prévoir les règles suivantes où  $e_3$  (respectivement  $e_7$ ) est remplacé par  $e_6$  qui joue le même rôle, celui de réception (respectivement de transmission) et où les bits 1 des séquences  $1 c e$  sont remplacés par 0 :

$$r'_{11} : y_n c e_6 c 0 c e_6 z 0 \rightarrow 0 c e_6 y_n c_{n-1} 0 c^1 e_6 z 0, g$$

puis

$$r'_{12,j} : 0 c e_7 x 0 c^i e_7 0 c_j y^{2-i+1} 0 \rightarrow 0 c e_7 x 0 c^{i+1} e_7 1 c_{i+1} y^{2-i}, d$$

où  $y^i = y_1 \dots y_i$  et  $y^{2-i}$  est le vecteur  $y^{2-i}$  "privé" de sa dernière composante.

$$r'_{13,j} : y_{2-i} c e_6 y_{2-i+1} \dots y_n c_i 0 c^1 e_6 z 0 \rightarrow 0 c e_6 y_{2-i} \dots y_n c_{i-1} 0 c^{i+1} e_6 z 0, g$$

Le processus s'achève par les règles suivantes:

$$r'_{14} : 0 c e_7 x 0 c^2 e_7 0 c_n 0 \rightarrow 0 c e_7 x 0 c^2 e_7 0 c_n 0, d$$

et

$$r'_8 : 0 c e_6 y_1 \dots y_n c_n 0 c^2 e_6 z 1 \rightarrow 1 c e_8 y c 0 c e_8 z 0, g$$

où  $e_8$  va jouer un rôle analogue à  $e_2$  : il signifiera le passage vers la gauche du bloc de traitement. On applique alors la règle  $r''_4$  qui agit de manière analogue à  $r_4$  où  $e_7$  va jouer le rôle de  $e_4$  :

$$r''_4 : 0 c e_7 x 0 c^2 e_7 0 c_n 1 \rightarrow 1 c e_6 c 0 c e_6 x 0, g$$

Puis on reboucle sur  $r'_4$ . Le processus de décalage se poursuit alors progressivement jusqu'à atteindre l'extrémité gauche de la configuration courante lorsqu'on aboutit à la situation :  $0 c e_7 \Delta 0 c^2 e_7 0 c^2 1$ . On applique alors la règle  $r_{11}$  qui aura pour effet d'amorcer le "retour" vers la droite (plus précisément vers la position marquée  $e_0$  et qui caractérise l'emplacement de la nouvelle emprise du bloc de traitement de la machine à simuler) en remettant en "état courant" la configuration ( $e_7$  remplacé par  $e$ ) et en transportant vers la droite le bit 1 pour indiquer qu'il s'agit de la phase de retour:

$$r_{11} : 0 c e_7 \Delta 0 c^2 e_7 0 c_n 1 \rightarrow 0 c e \Delta 0 c e \Delta 1, d$$

On applique ensuite les règles  $r_{12}$  et  $r_{13}$  qui auront pour effet de remettre en état courant la partie gauche de la configuration par rapport au lieu marqué par  $e_0$  ( $e_6$  et  $e_8$  remplacés par  $e$ )

$$r_{12} : 1 c e_6 x 0 c e_6 y 0 \rightarrow 0 c e x 0 c e y 0, d$$

$$r_{13} : 0 c e_8 x 0 c e_8 y 0 \rightarrow 0 c e x 0 c e y 0, d$$

jusqu'à atteindre  $0 c e_8 x 0 c e_8 y 1$  où on applique alors:

$$r_{14} : 0 c e_8 x 0 c e_8 y 1 \rightarrow 0 c e x 0 c e y 1, d$$

Après un mouvement à droite, on aboutit à la situation  $1 c e_0 x 1 c e_0 y 1$  qui caractérise l'entrée d'une nouvelle phase de calcul. La configuration se présente alors ainsi:

---


$$\dots \Delta 0 c e a_{i_1} 0 c e a_{i_2} 0 c e a_{i_3} \dots 0 c e a_{i_{p-1}} 0 c e a_{i_p} \boxed{1 c e a_{i_{p+1}} 1 c e a_{i_{p+2}} 1} c e a_{i_{p+3}} 1 c e \dots 1 c e a_{i_{k-1}} 1 c e a_{i_k} 1 c e \Delta \dots$$


---

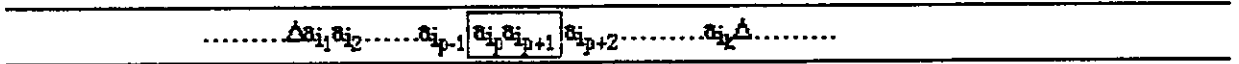
Ce qui correspond bien au résultat souhaité.

2.3.2 LE MOUVEMENT A GAUCHE

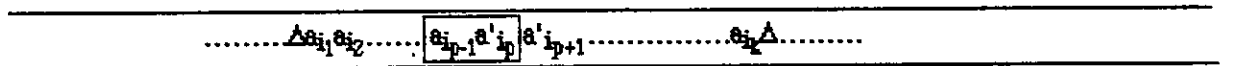
On va maintenant examiner le cas où la règle applicable est :

$$a_{i_p} a_{i_{p+1}} \rightarrow a'_{i_p} a'_{i_{p+1}} \cdot \xi$$

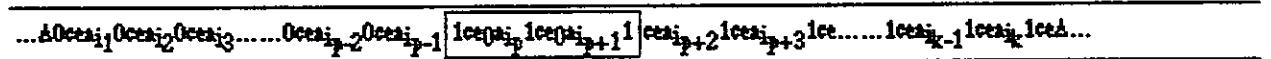
Supposons que la position courante à l'instant t de la machine à simuler soit celle-ci :



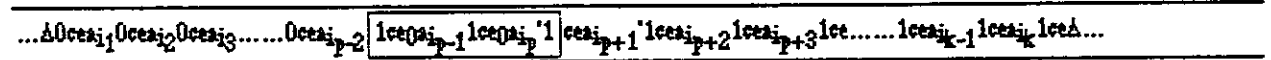
Alors à l'instant t+1 on doit aboutir à :



Comme pour le cas où la règle applicable est :  $a_{i_p} a_{i_{p+1}} \rightarrow a'_{i_p} a'_{i_{p+1}} \cdot d$ , on représente ainsi, la situation courante dans la machine à décalage maximum :



Le but est de pouvoir aboutir en un temps fini, par un ensemble fini de règles à la configuration :



Pour cela on utilisera des règles qu'on notera  $R_1$  analogues à celles ( notées  $r_j$ ) utilisées pour le cas où la règle applicable est :  $a_{i_p} a_{i_{p+1}} \rightarrow a'_{i_p} a'_{i_{p+1}} \cdot d$ .

La première règle  $R_1$  va consister à opérer la transformation  $xy \rightarrow x'y'$  puis à marquer le lieu sous l'emprise du bloc de traitement par  $E_1$  puis à amorcer la progression du bloc de traitement vers la gauche en effectuant un déplacement maximum par rapport à sa position précédente : c'est à dire un mouvement  $g = -(4n+12)$  bits :

$$R_1 : 1c_{e_0} x 1c_{e_0} y 1 \rightarrow 0c_{E_1} x' 1c_{E_1} y' 1, \quad g$$

La deuxième règle  $R_2$  aura pour effet de continuer la progression du bloc de traitement vers la gauche tout en marquant cette partie de la configuration par  $E_2$ :

$$R_2: 0 c e x 0 c e y 0 \rightarrow 0 c E_2 x 0 c E_2 y 0, g$$

La répétition de la règle  $R_2$  fait aboutir à l'extrémité gauche de la configuration caractérisée par la situation particulière

$0 c e \Delta 0 c e x 0$  ( $x = \Delta$  possible suivant la parité du cardinal de la configuration courante) où on applique alors la règle  $R_3$  qui aura pour effet de décaler d'un "cran" vers la gauche le symbole  $x$ ; de le substituer par le vecteur  $c$  qui aura pour but de "recevoir" le symbole situé à droite de la séquence  $0 c E_3$  puis à marquer cette zone par  $E_3$  qui est caractéristique d'un état de réception avant d'effectuer un mouvement  $d = +(4n+12)$  vers la droite:

$$R_3: 0 c e \Delta 0 c e x 0 \rightarrow 0 c E_3 x 0 c E_3 c 0, d$$

On découvre alors pour la première fois la situation  $0 c E_2 x 0 c E_2 y 0$ ; la règle  $R_4$  aura pour but de changer "l'état"  $E_2$  en  $E_4$  caractéristique d'un "état de transmission" (celle du vecteur  $x = x^1 = x_1 \dots x_n$  vers la gauche). Cette "transmission" sera contrôlée par un compteur  $c^i = 1 \dots 1, 0 \dots 0$  contenant  $i$  fois 1 et  $n-i$  fois 0, avant d'effectuer un mouvement vers la gauche:

$$R_4: 0 c E_2 x 0 c E_2 y 0 \rightarrow x_1 \dots x_n 0 E_4 c 0 c^i E_4 y 0, g$$

La règle  $R_5$  aura pour effet de recevoir la première composante  $x_1$  du vecteur  $x$ ; à incrémenter le compteur  $c$  puis à effectuer de nouveau le mouvement  $d = +(4n+12)$  vers la droite:

$$R_5: 0 c E_3 y 0 c E_3 c_n x_1 \rightarrow 0 c^i E_3 y 0 c E_3 c_{n-1} x_1 0, d$$

où  $c_{n-1}$  est le vecteur à  $n-1$  composantes nulles.

Ceci amorce un processus de transport du vecteur  $x$  vers la gauche, processus contrôlé par le compteur  $c^i$ . Le processus se poursuit par la mise en oeuvre des règles:

$$R_{5,i}: 0 x_1 \dots x_n c_{n-1} E_4 c 0 c^i E_4 y 0 \rightarrow x_1 \dots x_n c_{n-1} 0 E_4 c 0 c^{i+1} E_4 y 0, g$$

et

$$R_{6,i}: 0 c^i E_3 y 0 c E_3 c_{n-1} x_1 \dots x_1 \rightarrow 0 c^{i+1} E_3 y 0 c E_3 c_{n-1} x_1 \dots x_1 0, d$$

Le processus s'achève par l'application des règles suivantes:

$$R_7: 0 c_n E_4 c 0 c^2 E_4 y 0 \quad \rightarrow \quad 0 c E_4 c 0 c^2 E_4 y 0, g$$

où le bit 0 "transmis" à gauche sert à indiquer qu'il ne s'agit pas de la position caractérisant l'emprise du bloc de traitement.

Puis on applique:

$$R_8: 0 c^2 E_3 y 0 c E_3 0 x_1 \dots x_n \quad \rightarrow \quad 0 c e y c 0 c e x 1, d$$

La règle  $R_8$  remet en état courant ( substitution de  $E_3$  par  $e$ ; compteur  $c_n$  remis à zéro i.e.  $c$ ) la zone de "réception" afin de préparer la prochaine phase de calcul. Après un mouvement à droite, on aboutit pour la première fois à la situation suivante:  $1 c_n E_4 c 0 c^2 E_4 y 0$ , où la zone marquée par  $E_4$  doit jouer le rôle zone de réception (caractérisée par  $E_3$ ) : c'est ce que fait la règle  $R_9$  suivante qui agit de manière analogue à  $R_3$ :

$$R_9: 1 c_n E_4 c 0 c^2 E_4 y 0 \quad \rightarrow \quad 0 c E_3 x 0 c E_3 c 0, d$$

Puis on reboucle sur  $R_4$ , jusqu'à aboutir au lieu marqué  $0 c E_1 x 1 c E_1 y 1$ , caractérisant la position du bloc de traitement de la machine à simuler, alors on applique la règle  $R_{10}$  qui agit de manière analogue à  $R_4$  où  $E_5$  va jouer le rôle de  $E_4$  excepté en  $R_7$

$$R_{10}: 0 c E_1 x 1 c E_1 y 1 \quad \rightarrow \quad x_1 \dots x_n 1 E_5 c 1 c^1 E_5 y 1, g$$

Il faudra donc prévoir en plus les règles suivantes:

$$R'_{5,j}: 1 x_1 \dots x_n c_{n-i} E_5 c 1 c^i E_5 y 1 \rightarrow x_1 \dots x_n c_{n-i} 1 E_5 c 1 c^{i+1} E_5 y 1, g$$

analogue à  $R_{5,i}$  ( où  $E_5$  remplace  $E_4$ )

puis

$$R'_7: 1 c_n E_5 c 1 c^2 E_5 y 1 \rightarrow 1 c E_5 c 1 c^2 E_5 y 1, g$$

analogue à  $R_7$  (où où  $E_5$  remplace  $E_4$  à ceci près qu'on transmet le bit 1 à gauche au lieu de 0 pour indiquer qu'il faudra décaler vers la gauche la marque  $e_0$  caractérisant la position du bloc de traitement de la machine à simuler pour le pas de calcul suivant). C'est ce que fait la règle suivante, qui agit de manière analogue à  $R_8$ :



$$R'_8: 0c^2E_3y0cE_3x_1\dots x_n1 \rightarrow 1ce_0yc1ce_0x0,d$$

Après un mouvement à droite, on aboutit pour la première fois à la situation  $0c_nE_5c1c^2E_5y1$ , où la zone marquée par  $E_5$  doit jouer le rôle de zone de réception (caractérisée par  $E_5$ ): c'est ce que fait la règle  $R'_9$  suivante qui agit de manière analogue à  $R_9$ , à ceci près que  $E_3$  est remplacé par  $E_6$  qui joue le même rôle (celui de réception) et que les bits 1 des séquences  $1ce$  sont remplacés par 0 pour mettre en évidence qu'on traite la partie de la configuration se trouvant à droite bloc de traitement de la machine à simuler :

$$R'_9: 0c_nE_5c1c^2E_5y1 \rightarrow 1cE_6x1cE_6c0,d$$

Après un mouvement à droite, on aboutit pour la première fois à la situation  $0cex1ce c1$ , où on applique la règle  $R''_4$  qui agit de manière analogue à  $R_4$  où  $E_7$  va jouer le rôle de  $E_4$ :

$$R''_4: 0cex1cey1 \rightarrow x_1\dots x_n1E_7c1c^1E_7y1,g$$

Il faudra donc prévoir les règles suivantes où  $E_3$  (respectivement  $E_7$ ) est remplacé par  $E_6$  qui joue le même rôle, celui de réception (respectivement de transmission) et que les bits 0 de la séquence  $0ce$  sont remplacés par 1 :

$$R''_5: 1cE_6y1cE_6c_nx_1 \rightarrow 1c^1E_6y1cE_6c_{n-1}x_11,d$$

puis

$$R''_{5,i}: 1x_1\dots x_n c_{n-i}E_7c1c^iE_7y1 \rightarrow x_1\dots x_n c_{n-i}1E_7c1c^{i+1}E_7y1,g$$

$$R_{6,i}: 1c^iE_6y1cE_6c_{n-i}x_1\dots x_i \rightarrow 1c^{i+1}E_6y1cE_6c_{n-i-1}x_1\dots x_i1,d$$

Le processus s'achève par l'application des règles suivantes:

$$R''_7: 1c_nE_7c1c^2E_7y1 \rightarrow 0cE_7c1c^2E_7y1,g$$

analogue à  $R_7$  (où où  $E_7$  remplace  $E_4$  à ceci près qu'on transmet le bit 0 à gauche au lieu de 1 pour indiquer qu'il ne s'agit pas de la position caractéristique du bloc de traitement de la machine à simuler pour le pas de calcul suivant). C'est ce que fait la règle suivante, qui agit de manière analogue à  $R_8$ :

$$R''_8 : 1 c^2 E_6 y 1 c E_6 x_1 \dots x_n 0 \quad \rightarrow \quad 1 c E_8 y c 1 c E_8 x 0, d$$

où  $E_8$  va jouer un rôle analogue à  $E_2$ : il signifiera le passage vers la droite du bloc de traitement. On applique alors la règle  $R'''_4$  qui agit de manière analogue à  $R_4$  où  $E_7$  va jouer le rôle de  $E_4$ :

$$R'''_4 : 0 c_n E_7 c 1 c^2 E_7 y 1 \quad \rightarrow \quad 1 c E_6 x 1 c E_6 c 1, d$$

Puis on reboucle sur  $R''_4$ .

Le processus de décalage se poursuit alors progressivement jusqu'à atteindre l'extrémité droite de la configuration courante lorsqu'on aboutit à la situation suivante:  $0 c_n E_7 c 1 c^2 E_7 \Delta 1$ . On applique alors la règle  $R_{11}$  qui aura pour effet d'amorcer le "retour" vers la gauche (plus précisément vers la position marquée  $e_0$  et qui caractérise l'emplacement de la nouvelle emprise du bloc de traitement de la machine à simuler) en remettant en état courant la configuration ( $E_7$  remplacé par  $e$ ) tout en transportant vers la gauche le bit 1 pour indiquer qu'il s'agit de la phase de retour.

$$R_{11} : 0 c_n E_7 c 1 c^2 E_7 \Delta 1 \quad \rightarrow \quad 1 c e \Delta 1 c e \Delta 1, g$$

On applique ensuite les règles  $R_{12}$  et  $R_{13}$  qui auront pour effet de remettre en état courant la partie droite de la configuration par rapport au lieu marqué par  $e_0$  ( $E_6$  et  $E_8$  remplacés par  $e$ )

$$R_{12} : 1 c^2 E_6 y 1 c E_6 x_1 \dots x_n 1 \quad \rightarrow \quad 1 c e y 1 c e x 1, g$$

$$R_{13} : 1 c E_8 y 1 c E_8 x 1 \quad \rightarrow \quad 1 c e y 1 c e x 1, g$$

jusqu'à atteindre la situation  $1 c e_0 x 1 c e_0 y 1$  qui caractérise l'entrée d'une nouvelle phase de calcul. La configuration se présente alors ainsi:

---


$$\dots \Delta 0 c e a_{i_1} 0 c e a_{i_2} 0 c e a_{i_3} \dots 0 c e a_{i_{p-2}} \boxed{1 c e a_{i_{p-1}} 1 c e a_{i_p} 1 c e a_{i_{p+1}} 1 c e \dots 1 c e a_{i_{k-1}} 1 c e a_{i_k} 1 c e a_{i_{k+1}} \dots}$$


---

Ce qui correspond bien au résultat souhaité.



**DOUZZAEDIBISAB****CISTE**

Dieu-le Père était un artiste. Aussi aimait-il, pendant la Génèse, se reposer de temps en temps en faisant quelque chose de beau. Particulièrement épuisé par la conception des dinosaures, décida-t-il de se reposer en créant les fleurs.

Il en avait déjà imaginé quelques dizaines de milliers, toutes étalées sur son immense table de travail, quand il se dit soudain qu'elles seraient bien plus belles un petit peu repassées. Sitôt dit sitôt fait, et pendant que Maïa l'Abeille, éblouie, bourdonnait amoureusement tout autour de la tête du Maître, Dieu repassait, défripait des pétales éclatants de couleurs, tuyautait d'élégants cépales, donnait un petit coup de ci, de là, à quelque feuillage un peu fatigué.

Maïa à qui toute l'odeur de pollen, renforcée par la chaleur du fer, montait à la tête, saisie d'une ivresse folle, se précipita sur l'Œil

de Dieu-le-Père qui l'attirait comme un céleste miroir.

Dieu, agacé par cette sarabande fit un pas de recul et mit le Pied sur Haïka la vipère tendrement lovée près de Lui, et qui, somnolente, se chauffait à son soleil.

Haïka surprise mordit Dieu au talon. **"Nom de Moi !"** gronda-t-il dans un coup de tonnerre qui roula longtemps sur le nouveau Monde en pleine gestation, et qui rendit définitivement presque sourd tout ce qui vivait de reptiles à cette époque.

L'éclair, lui, fut brutal et fit sauter tous les plombs. Le fer à repasser était en panne. Il ne restait plus qu'une fleur, c'était le Ciste.

Et Dieu-le-Père, tout penaud de s'être ainsi laissé aller, se dit: **"J'arrête là !"**

Voilà pourquoi la fleur de ciste avec ses quatre pétales d'un délicieux vieux rose ou blanc pétillant sur le fond de délicat velours gris-vert du feuillage, est désormais restée toute froissée.

**LA PENSEE DE LA SEMAINE.**

Il est parfaitement impossible de gagner du temps, par contre il est tout à fait raisonnable de concevoir qu'on puisse éviter d'en perdre... malheureusement c'est compter sans l'administration.

Les lunettes sont une prothèse pour la vue déficiente, l'ordinateur est un projet de prothèse pour la pensée, la jambe de bois une prothèse pour un membre amputé, le pouvoir, lui, est une prothèse du sexe.