

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

COMPUTATION - INFORMATION

COMITE SCIENTIFIQUE

N° 56 - JUN 2000

*Patrick Abellard
Françoise Adreit
Jalal Almhana
France Chappaz
M'hamed Charifi
Roger Cusin
Bernard Goossens
Patrick Isoardi
Robert Jacquier
Jean - Philippe Lehmann
Nadia Mesli
Patrick Sanchez
Rolland Stutzmann
André Tricot*

DIRECTEUR

Jean - Michel Knippel

REDACTEUR EN CHEF

Edmond Bianco

REDACTEUR ADJOINT

Sami Hilala

SECRETARIAT

Kalassoumi Adjilani

Université de Provence
Equipe Hermès. Case 33
3, place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone: (0)4 91 10 62 30
Télécopie : (0)4 91 50 91 10

DEPOSITAIRE

Université de Provence
Bibliothèque Vniversitaire
3, place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone: (0)4 91 62 44 16
Télécopie : (0)4 91 95 75 57

1 EDITORIAL.

Informatique, économie et compagnie,

par Edmond Bianco

7 CHAPITRE 8.

**LA MACHINE UNIVERSELLE
DE LA PROCEDURE FORMELLE,**

par Edmond Bianco

27 VOZZAVEDIBISAR.

Der Professor (le professeur),

par Annagret Taeschner

<http://scamup.univ-mrs.fr/biaa>

<http://www.up.univ-mrs.fr>

Publication trimestrielle, gratuite, de l'Université de Provence

Edition 2001

ISSN 0291 - 5413

BULLETIN D'INFORMATIQUE APPROFONDIE ET APPLICATIONS

COMPUTATION - INFORMATION

COMITE SCIENTIFIQUE

N° 56 - JUIN 2000

Patrick Abellard
Françoise Adreit
Jalal Almhana
France Chappaz
M'hamed Charifi
Roger Cusin
Bernard Goossens
Patrick Isoardi
Robert Jacquier
Jean - Philippe Lehmann
Nadia Mesli
Patrick Sanchez
Rolland Stutzmann
André Tricot

DIRECTEUR

Jean - Michel Knippel

REDACTEUR EN CHEF

Edmond Bianco

REDACTEUR ADJOINT

Sami Hilala

SECRETARIAT

Kalassoumi Adjilani

Université de Provence
Equipe Hermès. Case 33
3, place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone: (0)4 91 10 62 30
Télécopie : (0)4 91 50 91 10

DEPOSITAIRE

Université de Provence
Bibliothèque Vniversitaire
3, place Victor Hugo
F - 13331 Marseille Cedex 3
Téléphone: (0)4 91 62 44 16
Télécopie : (0)4 91 95 75 57

1 **EDITORIAL.**
Informatique, économie et compagnie,

par Edmond Bianco

7 **CHAPITRE 8.**
LA MACHINE UNIVERSELLE
DE LA PROCEDURE FORMELLE,

par Edmond Bianco

27 **VOUZZAVEDIBISAR.**
Der Professor (le professeur),

par Annagret Taeschner

<http://scamup.univ-mrs.fr/biaa>

<http://www.up.univ-mrs.fr>

Publication trimestrielle, gratuite, de l'Université de Provence

Edition 2001

ISSN 0291 - 5413

EDITORIAL,

INFORMATIQUE, ECONOMIE et COMPAGNIE.

LE LANGAGE

Le langage est un bien curieux instrument. Il est certes, chargé de symboles, et en quelque sorte, c'est bien sa raison d'être. En effet, il est bien plus facile de désigner un objet au moyen d'un mot, que de présenter l'objet lui-même, surtout s'il est absent ou difficile à manipuler. Mais à force de désigner des objets réels à l'aide de symboles, on en est venu tout naturellement à désigner des êtres qui n'étaient plus des objets, mais des concepts. L'emploi même du langage y pousse irrésistiblement. Le simple fait d'utiliser des mots oblige à créer tout un vocabulaire utile pour préciser l'objet langage, et les mots: "mot", "phrase", "lettre", "caractère", "son", etc..., et à un niveau encore plus abstrait: "règle de grammaire", "accord des participes", "conjugaison", "déclinaison", etc..., sont indispensables pour cerner les notions employées. La langue parlée, la langue écrite, la langue enregistrée exigent des concepts précis bien que parfaitement abstraits, pour être correctement construites. Et ceci afin d'atteindre le but ultime de l'expression: la compréhension. Mais quand on commence à farfouiller dans le domaine sans limite de l'abstraction, la tentation est grande d'aller fort loin. Et il est très facile de dépasser les bornes...

D'autant que rapidement ce fameux langage devient ambigu, et jouer sur les ambiguïtés devient un nouveau jeu.

C'est ainsi qu'il est un domaine où l'imagination langagière éclate merveilleusement en des feux d'artifices sémantiques à côté desquels les feux d'artifices pyrotechniques ne sont que bougies de l'ancien temps. Le comportement de l'homme par rapport à lui-même et par rapport à ce qu'on appelle encore "la Nature", est à lui seul un univers extraordinaire. Qu'on réfléchisse un instant aux notions véhiculées par les termes suivants pris presque au hasard parmi un foule d'autres:

superstition, croyances, religion, secte, au-delà, famille, patrie, armée, honneur, honnêteté, responsabilité, respectabilité, environnement, écologie, agriculture, élevage, guerre, économie, maintien de l'ordre, etc...

Si la science semble s'en tenir aux seuls enchaînements de faits reliés entre eux par des causes et des conséquences, et exige en plus que les processus décrits soient reproductibles à volonté, il y a bien des chapitres dans la connaissance humaine auxquels on accole en vain, me semble-t-il, des termes qui s'en réclament. C'est largement le cas pour l'ensemble des élucubrations qu'on rassemble sous le vocable "d'économie", qui vont du calcul pénible des ménages pour surnager au milieu de toutes les charges, en passant par le budget de l'état qui s'équilibre en toute aisance à grands coups d'impôts, sans oublier les scandales du Crédit Lyonnais, celui d'Elf, celui des frégates de Taiwan, blanchiment d'argent sale en tous genres, vente d'armes aux pays sous développés en échanges d'étranges contrats, tout ce qu'on ignore, les fluctuations de la bourse et autres acrobaties.

L'économie une science?

Après tout pourquoi pas? Bien que la question soit amusante, rien n'empêche de se la poser. Mais alors quels sont les actes indéfiniment reproductibles qui mettent en évidence une propriété fondamentale? En cherchant bien on peut effectivement en extraire plusieurs, mais pas forcément ceux qui sont officiellement reconnus. Tentons courageusement d'en essayer une liste.

POLITIQUE ET ECONOMIE, POUVOIR=DROITE

Si, faire de l'économie c'est se préoccuper des flux de ressources, alors, un peu à l'extérieur de l'économie mais avec, dessus, une incidence directe, il existe une curieuse métamorphose qui se révèle très systématique: un homme de droite qui arrive au pouvoir reste un homme de droite, quelles que soient les promesses électorales qu'il a pu faire. Par opposition, un homme qui se dit de gauche, quand il arrive au pouvoir, se met instantanément à pratiquer une politique de droite. Inutile, je crois de donner des exemples récents. Mais, si l'on remonte au début du siècle dernier, Clémenceau en fut un exemple frappant, si j'ose dire, par sa schizophrénie politique. Il agissait à gauche, dès qu'il était dans l'opposition, mais devenait un féroce "répressionnaire" dès qu'il était au pouvoir. Sans doute, ne peut-il exister d'autre pouvoir qu'un pouvoir de droite, car le pouvoir politique repose essentiellement sur le pouvoir économique, qui n'est rien d'autre que l'accumulation de ressources entre seulement quelques mains. Se faire connaître des masses exige un porte-voix extrêmement coûteux. Et sous peine d'une rapide extinction de voix l'utilisateur de cet amplificateur peut difficilement devenir autre chose qu'un porte parole du précieux donateur, qui le lui a procuré.

RELAXATION

Plus directement liée au domaine économique, une autre loi semble se dégager, quand on observe l'évolution des sociétés au cours des siècles. Sans pour autant s'attacher aux détails qui dépendent des moyens techniques de l'époque car ceux-là peuvent être très différents, alors que les grandes lignes, elles, persistent. Elles traversent les âges. Et ces lignes ressemblent fort à une suite de dents de scies, toutes semblables, mais de plus en plus grandes au fur et à mesure que les moyens s'accroissent. Le principe est simple: une communauté réussit au cours du temps à créer une société riche et organisée, c'est la partie ascendante et lente de la dent. Une autre communauté, qualifions là de barbare, surgit et dans l'espoir de s'emparer des richesses, détruit tout, trop stupide le plus souvent, pour profiter des richesses accumulées. C'est la partie descendante de la dent. La déchéance est, en général, rapide. Nous avons affaire à une sorte de phénomène de relaxation, un peu comme une chasse d'eau, j'arrête là ma comparaison.

Nombre de cités florissantes ont ainsi disparu, Troie, Carthage et bien d'autres. Mais, à ces époques reculées, tout était plus simple, les Grecs, gens civilisés ne pouvaient détruire une civilisation simplement parce qu'elle leur portait ombrage, alors on inventait la légende de la belle Hellène. De même, Rome, ne pouvant supporter l'outrecuidance d'une Carthage rebelle à la colonisation, inventait les guerres puniques, et en même temps un nouveau moyen d'enrichir une "sous-communauté".

COMMUNAUTE BARBARE

La communauté barbare, fondant sur une communauté riche pour la piller, était un concept plein d'avenir qui devait se développer et fleurir jusqu'à nos jours, toutefois en se transformant profondément, pour s'adapter. Actuellement, cette engeance de rapaces ne se concentre plus seulement en une cité, qui lorgne vers une autre cité, ou un pays qui lorgne vers les richesses d'un autre pays, elle est devenue polymorphe. Déjà du temps des rois Philippe, les grandes compagnies jouaient un rôle ambigu. Le concept de foire d'empoigne s'est, depuis, largement répandu en diffusant à travers toute la civilisation. Un petit exemple: Michelin déclare des milliards de bénéfice puis sous le prétexte vague, mais signifiant, d'accroître sa compétitivité internationale, s'apprête à jeter 7000 employés à la rue. Pourtant ces employés ont participé à l'accroissement de la fortune Michelin.

Le concept de cité barbare qui pirate la cité laborieuse ne marche plus sous une forme aussi brutale et simpliste que dans ces temps reculés. Les pirates font désormais partie intégrante de la population, il y sont intimement

mêlés. Certains même, je pense, ne se rendent guère compte du rôle qu'ils jouent. Observons les affaires de l'Erika, de l'élevage du porc de Bretagne, du sang contaminé, des vaches folles. Même lors de l'affaire Tchernobyl, on a vu se mettre en place spontanément une chaîne du mensonge. Le nuage radioactif ne pouvait traverser la frontière! On a l'habitude du bourrage de crâne de l'administration, seulement en l'occurrence le mensonge débouchait sur l'absence de précautions dont la moindre des conséquences était un nombre de cancers de la thyroïde et autres sarcomes, qui auraient pu être évités.

AGREGATION

La cité s'est transformée, elle est devenue le pays, le pays est devenu la communauté, qu'on l'appelle "Européenne", "Soviétique" tournée C.E.I, Etats Unis, ou autre, en attendant la mondialisation. Deux sortes de systèmes politiques se sont développés à l'échelle des nations, surtout à partir de la fin du XVIIIème siècle; la dictature, et la démocratie. Et ces deux systèmes, en apparence antagonistes, sont en fait, porteurs des mêmes conséquences économiques; et en parallèle des conséquences bien trop souvent dramatiquement humaines. Certes, les méthodes et les théâtres d'opérations diffèrent.

Le nouveau mécanisme de création de la "communauté barbare" est si simple et paraît si naturel, qu'il passe volontairement ou involontairement inaperçu. Il est fondé sur le concept de "réussite sociale", concept qui n'a d'ailleurs rien de social.

Une petite parenthèse s'impose ici sur ce fameux concept de "social", parfaitement ambigu au demeurant. Quand on dit "sécurité sociale", ou loi sociale dont le but est de préserver certains intérêts de l'ensemble des citoyens, le sens est clair. Par contre, les termes de bénéfices sociaux, dettes sociales, capital social n'ont de sens que rapportés à la notion étroite de société telle qu'elle est définie dans le code civil. Et un "bénéfice social" peut n'avoir rien de social au sens du premier terme, s'il se fonde, comme chez Michelin sur la mise à la porte de 7000 employés.

PHASE ASCENDANTE.

Observons alors l'évolution d'une société située dans le temps entre deux catastrophes, puisque le sort de nos civilisations semble rythmé par de périodiques bouleversements d'origine humaine. Partons d'une situation de fin de calamité, la chute de Napoléon, ou bien la chute de Hitler. En ce point de ruine, observons les sociétés françaises, allemandes ou anglaises. Dégagé d'une oppression insupportable, bercé par l'illusion des lendemains qui chantent, tout le monde se met au travail. Plein emploi. En quelques

années les pays se redressent, mais cette période de forte croissance économique est particulièrement profitable aux composantes de la "communauté barbare". Certains apparaissent spontanément. D'autres s'étaient déjà enrichi en collaborant avec l'organisation Todt, ou en pratiquant le marché noir, les B.O.F(*) .

Par exemple, André Morice est le constructeur du mur de la "forteresse Europe", qu'on retrouve pour la barrière électrifiée qui sépare l'Algérie de la Tunisie dans les années soixante. Mais, tous s'enrichissent outrageusement.

Les lois démocratiques n'empêchent nullement les enrichissements éhontés. Et pendant que des mouvements de revendication très durs arrachent quelques lois sociales, les holdings s'agrègent de plus en plus lourdement transgressant aisément des frontières normalement opaques au simple citoyen. Car il est nécessaire que les capitaux de diverses natures puissent converger facilement vers le coffre fort central de l'Europe, où il se verrouille dans des comptes secrets.

Là intervient la technologie.

Si complexe que puisse paraître notre société, on peut extirper des principes simples qui sont à la base de son organisation, nous venons d'en voir une partie, nous compléterons notre exploration dans le prochain éditorial.

A suivre.

Edmond Bianco

(*): B.O.F. signifie Beurre Oeufs Fromages, denrées ultra rares pendant la guerre car quasiment toutes confisquées pour l'armée allemande, avec l'accord de "Vichy". Le peu qui restait faisait l'objet du marché noir.

CHAPITRE 8

LA MACHINE UNIVERSELLE DE LA PROCEDURE FORMELLE

Edmond Bianco

(suite du N° 53 de Juin 1999)

LA PROCEDURE

Dans le chapitre qui lui est consacré, nous avons bien vu que la Machine Formelle est construite autour de la notion de procédure. En fait, elle est conçue à partir d'un problème précis et large à la fois, celui que pose l'utilisation systématique d'un algorithme standard. La programmation est un acte coûteux, humainement coûteux. Il est donc raisonnable de façonner soigneusement et une fois pour toutes, des calculs largement utiles qu'on pourra reprendre partout où on peut en avoir besoin, sans plus être obligé de les réécrire. Mais cela implique quelques règles précises qui permettent la communication entre ces algorithmes définitifs et la mouvance du contexte dans lequel on les insère. En précisant ces règles nous pourrions voir apparaître la solution du problème posé plus haut. En l'occurrence, l'insertion de la procédure sans avoir ni à en modifier, ni à en recopier le texte. Je vais donc compléter le langage Procédure Formelle avant d'en concevoir la machine universelle qui contient précisément la solution en question.

Je rajoute donc deux instructions, l'une qui sert à déclarer les listes de variables, l'autre est une insertion de procédure.

Dans l'instruction de déclaration on précise le nom de la procédure, le nombre de paramètres formels, le nombre d'index, le nombre de variables locales avec la liste de leur dimension.

L'insertion de procédure permet de préciser le nom de la procédure appelée, et la liste des paramètres effectifs.

Déclaration: procédure nom de procédure ;
 paramètres entier ;
 index entier ;
 variables locales entier : liste d'entiers;

Insertion: insérer nom de procédure (liste d'entiers) ;

nom de procédure ::= identificateur (Cf. Algol)
entier ::= chiffre | entier chiffre (Suite de chiffres)
chiffre ::= 0 | 1 | 2 | ... | 9
liste d'entiers ::= entier | liste d'entiers , entier

Dans ce langage, comme dans tout langage de machine on ne traite que des entiers, mais on y utilise deux sortes de variables: les variables simples et les tableaux. Les "index" sont des entiers réservés à l'usage d'indice de tableau. Dans la déclaration des variables locales, on déclare d'abord le nombre total de variables, et ensuite la liste des dimensions de ces variables. Si la variable est simple on note: 1, si c'est un tableau on note son volume, le nombre de valeurs qui le constituent.

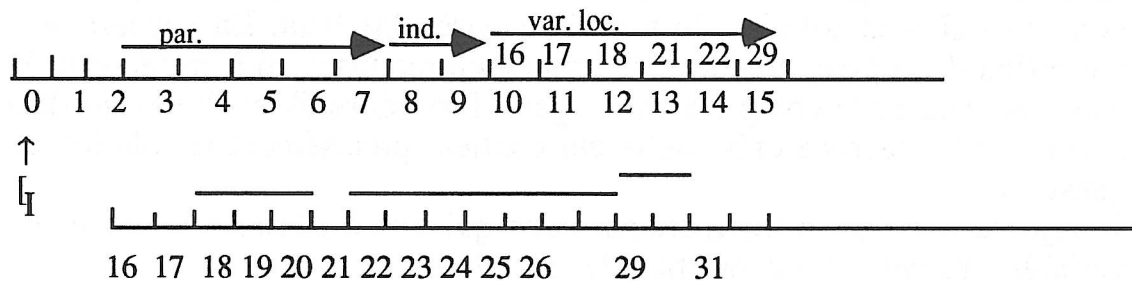
CONFIGURATION DE LA PROCEDURE

Telle qu'elle est présentée, l'instruction de déclaration de procédure, me permet de définir une structure de la configuration qui lui correspond. Pour cela je prends un exemple:

```

procédure  CALC ;
paramètres 5 ;
index 2 ;
variables locales 6 : 1 , 1 , 3 , 1 , 7 , 2 ;
  
```

L'origine de la configuration qui correspond à cette déclaration sera naturellement définie par le [I].



Les trois premières cases sont réservées pour la machine universelle qui s'en sert au moment du calcul d'une insertion puis d'un retour systématique.

Les variables déclarées occupent les cases qui suivent, dans l'ordre. D'abord les paramètres, une case par paramètre. Dans l'exemple ce sont les cases de 3 à 7. Ensuite les index, toujours une case par index, dans l'exemple, ce sont les cases 8 et 9. Ensuite on réserve autant de cases qu'il y a de variables locales, dans l'exemple ce seront les cases 10 à 15. Puis viennent les cases destinées à contenir les valeurs de ces variables locales, ainsi pour l'exemple:

La valeur de la	première variable, de volume	1	occupe la case	16,
" " "	deuxième " " "	1	"	17,
" " "	troisième " " "	3	"	les cases de 18 à 20,
" " "	quatrième " " "	1	"	la case 21,
" " "	cinquième " " "	7	"	les cases de 22 à 28,
La valeur de la	sixième variable, de volume	2	occupe les cases	29,30.

Les cases 10 à 15 contiennent les "noms des variables", c'est-à-dire les adresses par rapport à l'origine relative, de chaque ensemble de valeurs, ici, les adresses: 16, 17, 18, 21, 22, 29.

NOTE

Dans le chapitre 4 nous avons étudié la manière dont les paramètres se transmettent d'une configuration à l'autre au cours des insertions et des retours systématiques. La réservation de trois cases en tête de toute configuration, est la seule chose qui soit présentement modifiée. La mécanique du transfert des paramètres demeure rigoureusement la même, mais nous allons voir comment elle est rendue automatique grâce à la machine universelle.

CODE DU LANGAGE

Je choisis un jeu d'instructions assez complet ainsi décrit:

début

fin

procédure N ; LP ; X ; VL : LV ;

insérer N (LP) ; N est un nom de procédure,

LP une liste de paramètres,

X une liste d'index,

VL une liste de variables locales (v.l.),

LV le volume des valeurs des v.l.

x := y

x := y A y A représente les opérateurs: + , - , * , ÷

si y R y vers étiqu R représente: = , ≠ , > , ≥ , < , ≤

vers étiqu

x représente les 4 possibilités: [a , [a,b , [I , [w

et y représente toutes les possibilités de x plus la constante.

Pour construire la machine universelle il faut procéder à l'opération de codage. Il est nécessaire de retenir l'information signifiante et seulement cette information car c'est ce que va lire la machine universelle.

On va raisonner sur un exemple:

[2,3 := [4 * [6,1

Il est important de savoir que le traitement à faire subir aux trois opérandes est décrit par le couple ':=,*'. Ensuite il faudra retenir la suite des indices 2, 3, 4, 6, 1. Mais cette liste n'est pas suffisante, car elle pourrait aussi bien représenter une instruction de la forme:

$$\begin{aligned} [2] &:= [3,4] * [6,1] && \text{ou même} \\ [2,3] &:= [4,6] * [1] \end{aligned}$$

qui n'ont pas du tout le même sens.

Je vais rajouter une information dont le rôle est d'indiquer le type de l'opérande dans les cas où il peut y avoir ambiguïté, ce qui est le cas des instructions d'affectation. Aux quatre cas possibles:

$$[a], [a,b], [I], [w]$$

on rajoute la constante, pour la partie droite de l'affectation. Je choisis arbitrairement le code suivant:

opérande:	[a	[a,b	[I	[w	Const
code:	0	1	2	3	4

Si je désigne par n1, n2, n3 les codes attachés aux trois opérandes, pris , chacun dans cette liste, pour les trois exemple qui précèdent j'obtiens les valeurs:

$$\begin{aligned} [2,3] &:= [4] * [6,1] && n1=1 \quad n2=0 \quad n3=1 \\ [2] &:= [3,4] * [6,1] && n1=0 \quad n2=1 \quad n3=1 \\ [2,3] &:= [4,6] * [1] && n1=1 \quad n2=1 \quad n3=0 \end{aligned}$$

Et pour une instruction de la forme:

$$[I] := [w] * 25 \quad n1=2 \quad n2=3 \quad n3=4$$

LE CODE DE L'INSTRUCTION

Chacun des indices des opérandes, dans l'instruction d'affectation est représenté par un entier de la valeur de l'indice. On distingue des opérateurs différents, chacun représenté par un entier arbitraire.

opérateur	code	opérateur	code
<u>début</u>	0	<u>:=+</u>	8
<u>fin</u>	1	<u>si=</u>	9
<u>procédure</u>	2	<u>si≠</u>	10
<u>insérer</u>	3	<u>si<</u>	11
<u>:=</u>	4	<u>si≤</u>	12
<u>:=+</u>	5	<u>si></u>	13
<u>:=-</u>	6	<u>si≥</u>	14
<u>:=*</u>	7	<u>vers</u>	15

Ce choix de codes n'a d'intérêt que pour l'exemple de machine universelle, et on pourra éventuellement en changer pour la construction d'un processeur qui matérialiserait cette machine universelle. De toute manière,

pour la clarté de l'exposé, j'utiliserai les formes présentées dans la colonne opérateur, en les encadrant de "quote":

"deb", "fin", "pro", "ins", ":", ":", ":", ... "vers".

CONFIGURATION DE LA MACHINE UNIVERSELLE

Pour définir cette configuration se pose tout de suite un choix de système, entendons système de gestion de l'ordinateur. En effet, selon ce choix, les choses se présentent de manière différente. Je choisis pour commencer de considérer la machine universelle comme une procédure qui s'insère dans le système, procédure la plus extérieure.

Je définis les cinq paramètres nécessaires:

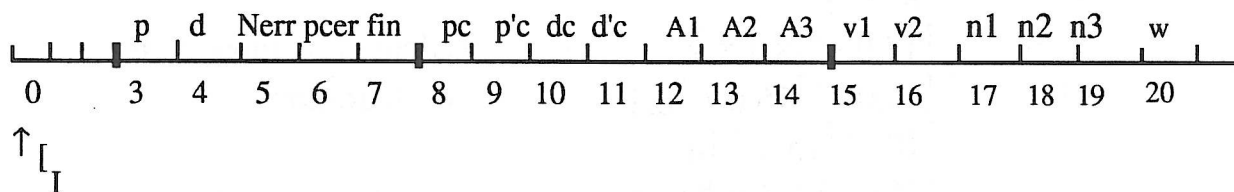
p l'accès au code du programme de la procédure déroulée,
d l'accès à la configuration de cette procédure,
Nerr un code d'erreur détectée, zéro signifie: pas d'erreur,
pcer accès au code erroné,
fin code indiquant si le "fin" de la procédure déroulée a été atteint.

Il faut quatre index:

pc, p'c, dc, d'c;

et enfin cinq variables locales:

v, v1, n1, n2, n3.



Dans le choix de système que je fais, je rajoute une propriété que je justifierai plus loin au chapitre qui traite de l'autojectivité. Simplement j'organise l'ensemble système-machine universelle, pour que celle-ci ne déroule qu'une seule instruction à la fois. C'est cette organisation qui exige la présence du paramètre "fin", car le système insérant la machine universelle pour chacune des instructions ne peut plus savoir quand se termine le calcul. C'est donc du rôle de la machine universelle de fournir cette information sur la rencontre de l'instruction "fin" de la procédure déroulée.

procédure MUPF ;
par 5 ;
ind 7 ;
v1 6: 1 , 1 , 1 , 1 , 1 , 1 ;

début

ENTREE: si [3,0 = "deb" vers DEB
si [3,0 = "fin" vers FIN
si [3,0 = "proc" vers DECL
si [3,0 = "ins" vers INSER
si [3,0 = ":@" vers AFF
si [3,0 = ":@" vers DPEAR
si [3,0 = ":@" vers DPEAR
si [3,0 = ":@" vers DPEAR
si [3,0 = ":@" vers DPEAR
si [3,0 = "si=" vers COND
si [3,0 = "si≠" vers COND
si [3,0 = "si<" vers COND
si [3,0 = "si≤" vers COND
si [3,0 = "si>" vers COND
si [3,0 = "si≥" vers COND
si [3,0 = "vers" vers VERS
 vers ERR18

DECL : [8 := [3
 [15,0 := [8,2 + 3 Calcul du volume de la
 [15,0 := [15,0 + [8,3 configuration.
 [10 := [15,0 + [4
 [15,0 := [15,0 + [8,4 Par + ind + v1 + 3 en 15,0 :
 [9 := [8 + 5
 [16,0 := [8,4

DECLH : si [16,0 = 0 vers DECB Mise en place des adresses
 [10,0 := [15,0 des variables locales.
 [15,0 := [15,0 + [9,0
 [9 := [9 + 1
 [16,0 := [16,0 - 1
 [10 := [10 + 1
 vers DECLH

DECB : [4,0 := [15,0
vers ENTREE

Le volume total en case 0 de la configuration.

INSER : [10 := [4,0 + [4
[10,2 := [3 + [3,2
[10,2 := [10,2 + 3
[10,1 := [4,0
[16,0 := [3,2
[11 := [10 + 3
[9 := [3 + 3

Préparation de la configuration de la procédure à insérer.
Calcul du retour dans le code.
Retour dans la configuration.

INSH : si [16,0 = 0 vers INSB
si [9,0 ≥ [4,0 vers ERR16
si [9,0 < 3 vers ERR17
[11,0 := [9,0 - [4,0
[11 := [11 + 1
[9 := [9 + 1
[16,0 := [16,0 - 1
vers INSH

Chargement des paramètres effectifs, dans la configuration insérée.

INSB : [3 := [3 + [3,1
[4 := [4 + [4,0
vers ENTREE

AFF : [8 := [3
[9 := [8 + 2
[17,0 := [8,1 ÷ 2⁶
[18,0 := [8,1 * 2^{H-6}
[18,0 := [18,0 ÷ 2^{H-3}

Calcul des n1 et n2.

AFF1 : si [17,0 = 0 vers OPAF0
si [17,0 = 1 vers OPAF1
si [17,0 = 2 vers OPAF2
si [17,0 = 3 vers OPAF3
vers ERR10

Sélection du type du premier opérande.

OPAF0 : si [9,0 ≥ [4,0 vers ERR11 Opérande de la forme: [a .
 si [9,0 < 3 vers ERR12
 [12,0 := [4 + [9,0
 [9 := [9 + 1
 vers AFF2

OPAF1 : si [9,0 ≥ [4,0 vers ERR11 Opérande de la forme: [a,b .
 si [9,0 < 3 vers ERR12
 [12 := [4 + [9,0
 [12 := [12,0 + [4
 [12 := [12 + [9,1
 [9 := [9 + 2
 vers AFF2

OPAF2 : [12 := [20 [20 est l'adresse de la valeur qui simule [w
 vers AFF2

OPAF3 : [12 := 4 Car 4 est l'adresse qui contient d.
 vers AFF2

AFF2 : si [18,0 = 0 vers OPAF01 Sélection du type du
 si [18,0 = 1 vers OPAF11 second opérande.
 si [18,0 = 2 vers OPAF21
 si [18,0 = 3 vers OPAF31
 si [18,0 = 4 vers OPAF41
 vers ERR13

OPAF01 : si [9,0 ≥ [4,0 vers ERR14 Opérande de la forme [a .
 si [9,0 < 3 vers ERR15
 [13 := [4 + [9,0
 [9 := [9 + 1
 vers AFF3

OPAF11 : si [9,0 ≥ [4,0 vers ERR14 Opérande de la forme [a,b .
 si [9,0 ≥ 3 vers ERR15
 [13 := [4 + [9,0
 [13 := [13,0 + [4
 [13 := [13 + [9,1
 [9 := [9 + 2
 vers AFF3

OPAF21 : $[13 := [20$ $[20$ est l'adresse de la valeur qui simule $[w$
vers AFF3

OPAF31 : $[13 := 4$ Car 4 est l'adresse qui contient d.
vers AFF3

OPAF41 : $[13 := [9$ C'est l'adresse de la constante
 $[9 := [9 + 1$ dans le code qui est utilisée.
vers AFF3

AFF3 : $[12,0 := [13,0$ Calcul de l'affectation.
 $[3 := [9$
vers ENTREE

DPEAR : $[8 := [3$ Séparation des $n1, n2$ et $n3$.
 $[9 := [8 + 2$
 $[17,0 := [8,1 \div 2^6$
 $[18,0 := [8,1 * 2^{H-6}$ H est le nombre de bits
 $[18,0 := [18,0 \div 2^{H-3}$ que contient la case
 $[19,0 := [8,1 * 2^{H-3}$ de la machine.
 $[19,0 := [19,0 \div 2^{H-3}$

DPEA1 : si $[17,0 = 0$ vers OPDE0 Sélection du type de $n1$.
 si $[17,0 = 1$ vers OPDE1
 si $[17,0 = 2$ vers OPDE2
 si $[17,0 = 3$ vers OPDE3
vers ERR1

OPDE0 : si $[9,0 \geq [4,0$ vers ERR2 Opérande de la forme: $[a$.
 si $[9,0 < 3$ vers ERR3
 $[12,0 := [4 + [9,0$
 $[9 := [9 + 1$
vers DPEA2

- OPDE1 : si [9,0 ≥ [4,0 vers ERR2 Opérande de la forme [a,b.
si [9,0 < 3 vers ERR3
[12 := [4 + [9,0
[12 := [12,0 + [4
[12 := [12 + [9,1
[9 := [9 + 2
vers DPEA2
- OPDE2 : [12 := [20 [20 est l'adresse de la valeur qui simule [w.
vers DPEA2
- OPDE3 : [12 := 4 Car 4 est l'adresse qui contient d.
vers DPEA2
- DPEA2 : si [18,0 = 0 vers OPDE02 Sélection du type de n2.
si [18,0 = 1 vers OPDE12
si [18,0 = 2 vers OPDE22
si [18,0 = 3 vers OPDE32
si [18,0 = 4 vers OPDE42
vers ERR6
- OPDE02 : si [9,0 ≥ [4,0 vers ERR4 Opérande de la forme: [a .
si [9,0 < 3 vers ERR5
[13 := [4 + [9,0
[9 := [9 + 1
vers DPEA3
- OPDE12 : si [9,0 ≥ [4,0 vers ERR4 Opérande de la forme: [a,b .
si [9,0 < 3 vers ERR5
[13 := [4 + [9,0
[13 := [13,0 + [4
[13 := [13 + [9,1
[9 := [9 + 2
vers DPEA3
- OPDE22 : [13 := [20 [20 est l'adresse de la valeur qui simule [w.
vers DPEA3
- OPDE32 : [13 := 4 Car 4 est l'adresse qui contient d.
vers DPEA3

OPDE42 : [13 := [9
 [9 := [9 + 1
 vers DPEA3

DPEA3 : si [19,0 = 0 vers OPDE03
 si [19,0 = 1 vers OPDE13
 si [19,0 = 2 vers OPDE23
 si [19,0 = 3 vers OPDE33
 si [19,0 = 4 vers OPDE43
 vers ERR9

Sélection du type de n3.

OPDE03 : si [9,0 ≥ [4,0 vers ERR7
 si [9,0 < 3 vers ERR8
 [14 := [4 + [9,0
 [9 := [9 + 1
 vers OPDEF

Opérande de la forme: [a .

OPDE13 : si [9,0 ≥ [4,0 vers ERR7
 si [9,0 < 3 vers ERR8
 [14 := [4 + [9,0
 [14 := [14,0 + [4
 [14 := [14 + [9,1
 [9 := [9 + 2
 vers OPDEF

Opérande de la forme: [a,b .

OPDE23 : [14 := [20
 vers OPDEF

[20 est l'adresse de la valeur qui simule [w.

OPDE33 : [14 := 4
 vers OPDEF

Car 4 est l'adresse qui contient d.

OPDE43 : [14 := [9
 [9 := [9 + 1
 vers OPDEF

OPDEF : si [8,0 = ':+ ' vers OPPL
 si [8,0 = ':- ' vers OPMS
 si [8,0 = ':* ' vers OPMUL
 si [8,0 = ':=÷ ' vers OPDIV

Sélection du type
d'opérateur.

<p>OPPL : [12,0 := [13,0 + [14,0 [20,0 := [w <u>vers</u> SORTDPE</p>	<p>Calcul de l'addition.</p>
<p>OPMS : [12,0 := [13,0 - [14,0 [20,0 := [w <u>vers</u> SORTDPE</p>	<p>Calcul de la soustraction.</p>
<p>OPMUL : [12,0 := [13,0 * [14,0 <u>vers</u> SORTDPE</p>	<p>Calcul de la multiplication.</p>
<p>OPDIV : [12,0 := [13,0 ÷ [14,0 SORTDPE: [3 := [9 <u>vers</u> ENTREE</p>	<p>Calcul de la division.</p>

<p>COND : [8 := [3 [9 := [8 + 2 [17,0 := [8,1 ÷ 2⁶ [18,0 := [8,1 * 2^{H-6} [18,0 := [18,0 ÷ 2^{H-3}</p>	<p>Séparation des n1 et n2.</p>
<p>COND1 : <u>si</u> [17,0 = 0 <u>vers</u> OPCO0 <u>si</u> [17,0 = 1 <u>vers</u> OPCO1 <u>si</u> [17,0 = 2 <u>vers</u> OPCO2 <u>si</u> [17,0 = 3 <u>vers</u> OPCO3 <u>si</u> [17,0 = 4 <u>vers</u> OPCO4 <u>vers</u> ERR1</p>	<p>Sélection du type de n1.</p>
<p>OPCO0 : <u>si</u> [9,0 ≥ [4,0 <u>vers</u> ERR2 <u>si</u> [9,0 < 3 <u>vers</u> ERR3 [12 := [4 + [9,0 [9 := [9 + 1 <u>vers</u> COND2</p>	<p>Opérande de la forme: [a .</p>

- OPCO1 : si [9,0 ≥ [4,0 vers ERR4 Opérande de la forme: [a,b.
si [9,0 < 3 vers ERR5
[12 := [4 + [9,0
[12 := [12,0 + [4
[12 := [12 + [9,1
[9 := [9 + 2
vers COND2
- OPCO2 : [12 := [20 [20 est l'adresse de la valeur qui simule [w.
vers COND2
- OPCO3 : [12 := 4 Car 4 est l'adresse qui contient d.
vers COND2
- OPCO4 : [12 := [9
[9 := [9 + 1
vers COND2
- COND2 : si [18,0 = 0 vers OPCO10 Sélection du type de n2.
si [18,0 = 1 vers OPCO11
si [18,0 = 2 vers OPCO12
si [18,0 = 3 vers OPCO13
si [18,0 = 4 vers OPCO14
vers ERR6
- OPCO10 : si [9,0 ≥ [4,0 vers ERR4 Opérande de la forme: [a.
si [9,0 < 3 vers ERR5
[13 := [4 + [9,0
[9 := [9 + 1
vers OPCOF
- OPCO11 : si [9,0 ≥ [4,0 vers ERR7 Opérande de la forme: [a,b.
si [9,0 < 3 vers ERR8
[13 := [4 + [9,0
[13 := [13,0 + [4
[13 := [13 + [9,1
[9 := [9 + 2
vers OPCOF

OPCO12 : [13 := [20 [20 est l'adresse de la valeur qui simule [w.
vers OPCOF

OPCO13 : [13 := 4 Car 4 est l'adresse qui contient d.
vers OPCOF

OPCO14 : [13 := [9
 [9 := [9 + 1
vers OPCOF

OPCOF : si [8,0 = 'si=' vers SIEG Sélection du type
 si [8,0 = 'si≠' vers SIDIF d'opérateur de comparaison.
 si [8,0 = 'si>' vers SIPG
 si [8,0 = 'si≥' vers SIPGE
 si [8,0 = 'si<' vers SIPP
 si [8,0 = 'si≠' vers SIPPE

SIEG : si [12,0 = [13,0 vers VRAI Calcul des comparaisons.
vers FAUX

SIDIF : si [12,0 = [13,0 vers FAUX
vers VRAI

SIPG : si [12,0 > [13,0 vers VRAI
vers FAUX

SIPGE : si [12,0 ≥ [13,0 vers VRAI
vers FAUX

SIPP : si [12,0 < [13,0 vers VRAI
vers FAUX

SIPPE : si [12,0 ≤ [13,0 vers VRAI
vers FAUX

VRAI : [3 := [3 + [9,0
vers ENTREE

FAUX : [3 := [9 + 1
vers ENTREE

VERS : [3 := [3 + [3,1
vers ENTREE

FIN : [3 := [4,2
[4 := [4 - [4,1
[7,0 := 'fin'
vers ENTREE

DEB: [3 := [3 + 1
vers ENTREE

ERR1: [6,0 := '1'
vers ENTREE
ERR2: [6,0 := '2'
vers ENTREE

ERR18: [6,0 := "18"
vers ENTREE

CALCUL DES OPERANDES

C'est un couple formé d'un code d'algorithme et d'une configuration sur laquelle il s'applique, qui est soumis à la machine universelle. Le rôle de cette dernière consiste à effectuer les calculs décrits par le code, et à reporter les résultats sur la configuration qui lui est attachée.

La configuration de la machine universelle a son origine définie par: [I et c'est par rapport à cette origine que le code est repéré par une adresse "p", et la configuration par "d". La configuration attachée au code déroulé est également repérée par un [I, mais cette origine est virtuelle par rapport à la précédente, en fait la machine universelle ne la connaît que par l'adresse "d".

Pour pouvoir effectuer les calculs comme ils sont décrits dans le code, la machine universelle doit ramener toutes les adresses par rapport à sa propre origine de configuration. Ainsi:

Pour l'opérande $[a$, atteindre à la case a de la configuration image, revient à calculer l'adresse $d + a$ valable, elle, pour la configuration de la machine universelle. Adresse calculée aux étiquettes:

OPAF0, OPAF01, OPDE0, OPDE02, OPDE03, OPCO0, OPCO10.

L'opérande $[a,b$ est un peu plus compliqué, il faut aller chercher dans la case a de la configuration image, une adresse image à reporter de d puis à corriger de b . Soit, calculer $x = d + a$, puis prendre $y = [x$, enfin obtenir l'adresse $d + y + b$. C'est ce qui est calculé aux étiquettes:

OPAF1, OPAF11, OPDE1, OPDE12, OPDE13, OPCO1, OPCO11.

CALCUL DES CONFIGURATIONS

Lorsqu'au cours du déroulement d'une procédure, un code d'instruction d'insertion de procédure est rencontré, la machine universelle va créer une configuration. La nouvelle configuration est placée à l'extrémité de l'actuelle, dont le volume est contenu dans sa case de N° 0. Or, le code de l'insertion contient deux sortes d'informations, le nom de la procédure à insérer, et la liste des paramètres effectifs.

Les paramètres effectifs sont des numéros de cases pris parmi les cases qui contiennent les variables de la configuration actuelle. Il s'agit pour la machine universelle de placer à partir de la quatrième case de la nouvelle configuration la suite des adresses qui vont permettre d'atteindre par rapport à son origine les variables de la configuration précédente.

Exemple

Une procédure A, déclarée

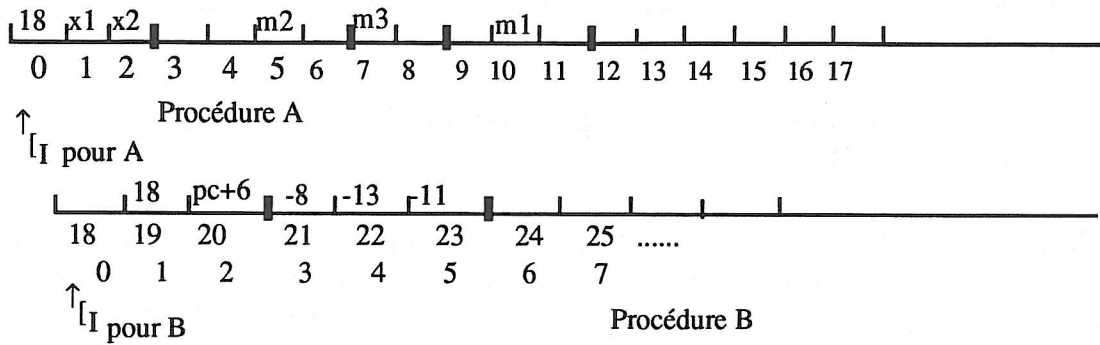
proc A ; par 4 ; ind 2 ; vl 3 : 3 , 1 , 2 ;

travaille sur une configuration de 18 cases, sa case 0 contient donc 18.

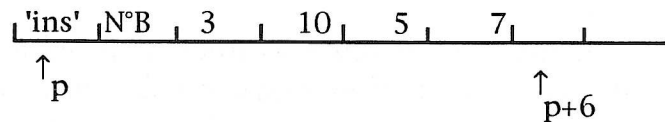
Cette procédure contient une instruction:

insérer B (10, 5, 7),

La procédure B a 3 paramètres, dont les valeurs vont donc apparaître dans les cases 3, 4, et 5 de sa configuration. Mais quand on traite cette instruction d'insertion on est encore dans la configuration de A, donc ces 3 cases y sont les cases de numéro 21, 22, 23.



Prenons le cas de l'instruction: insérer B (10 , 5 , 7);
dont le code est:



ce code indique en $p+2$ qu'il y a 3 paramètres effectifs qui sont: 10, 5, et 7, numéros des cases de la configuration de A, contenant 3 adresses m_1 , m_2 et m_3 .

La machine universelle transvase ces trois adresses dans les cases 21, 22 et 23, après les avoir corrigées pour qu'elles soient valables par rapport à l'origine de la configuration de B.

Le calcul est simple, il suffit au numéro de case pris dans le code de l'instruction, de soustraire le volume de la configuration insérante pris en case 0, soit ici 18. Cela donne: $10-18 = -8$, $5-18 = -13$, $7-18 = -11$.

L'algorithme correspondant se trouve à l'étiquette INSH entrée de la boucle qui tourne sur le nombre de paramètres récupérés dans le code de l'instruction.

Cette mise en place des paramètres achevée il faut calculer l'accès à la procédure insérée. Dans une première hypothèse j'admets le choix simple d'une insertion calculée comme une instruction VERS. Dans le code de l'instruction d'insertion on trouve un entier: N°B, censé mesurer le nombre de cases qui sépare celle qui contient le code 'ins', de la première case du code inséré. Ce calcul est réalisé à l'étiquette INSB.

L'instruction suivante déroulée par la machine universelle est précisément la première de la procédure insérée, ici B. Cette instruction est de code 'decl', et contient tout ce qu'il faut pour définir les variables de la configuration de B. En l'occurrence, le nombre de paramètres formels, le nombre d'index, le nombre de variables locales ainsi que le volume de chacune d'elles.

On vient de voir comment sont chargées les adresses des paramètres, les index sont des adresses calculées pendant que se déroule la procédure, il faut mettre en place les adresses des variables locales.

Supposons pour poursuivre l'exemple, une procédure B déclarée:

procédure B ; par 3 ; ind 4 ; vl 3 : 5 , 2 , 1 ;

Dans la boucle d'étiquette DECLH, on met en place dans la première case réservée aux variables locales, l'adresse de la première case de libre à droite, ainsi dans cet exemple, en case 10 on met l'adresse 13. Dans la case suivante des variables locales, on met l'adresse qui tient compte du volume de la première variable, soit $13+5 = 18$. Ainsi de suite.

REMARQUE 1

Le traitement de l'instruction d'insertion est directement lié à la notion de système. En effet c'est une articulation remarquable entre des éléments différents d'un programme. En ces points, dans un déroulement, un système peut être amené à intervenir pour contrôler des temps de déroulement, ou gérer de la place en mémoire. On pourrait donc envisager des modes de traitement plus complexes et plus souples. Un premier pas vers la maîtrise de l'occupation de la mémoire, consisterait à remplacer la distance que contient le code de l'instruction d'insertion, par une information ayant signification d'identificateur. Le calcul de l'ALLER-A vers la procédure insérée, passerait alors par l'exploration d'une table des procédures contenant pour chacune d'elles un descriptif complet, dont en particulier, une localisation momentanée.

C'est ainsi qu'on pourrait envisager de n'avoir pas toutes les procédures en mémoire centrale, quitte à les appeler lors d'une insertion, seulement quand leur présence est indispensable. Et les sortir dès qu'elles ne sont plus directement utiles. Cette gestion faisant bien partie des attributions du système.

REMARQUE 2

Dans la configuration de la procédure, j'ai choisi de traiter les trois types de variables, les paramètres, les index, les variables locales, sous la même forme. Dans les trois cas ce sont des adresses. Pour les paramètres, c'est évidemment indispensable. Il en est de même pour les index. Quant aux variables locales, il pourrait paraître plus économique de ne faire apparaître que leurs valeurs dans la configuration.

Mais cette économie est d'autant plus négligeable, qu'on a affaire à des variables qui sont des tableaux volumineux. Surtout, n'importe laquelle de ces variables pouvant être prise comme un paramètre pour une éventuelle procédure insérée, il est important que leur forme soit identique, pour en assurer un traitement homogène.

REMARQUE 3

Le traitement de l'instruction d'insertion qui se fait à l'étiquette INSER, prépare les trois premières cases de la configuration qui va appartenir à la procédure qu'on est en train d'insérer. Notamment ces deux instructions préparent l'adresse de retour à l'instruction qui suit l'instruction d'insertion qu'on est en train de traiter:

$$\begin{aligned} [10,2] &:= [3 + [3,2] \\ [10,2] &:= [10,2 + 3 \end{aligned}$$

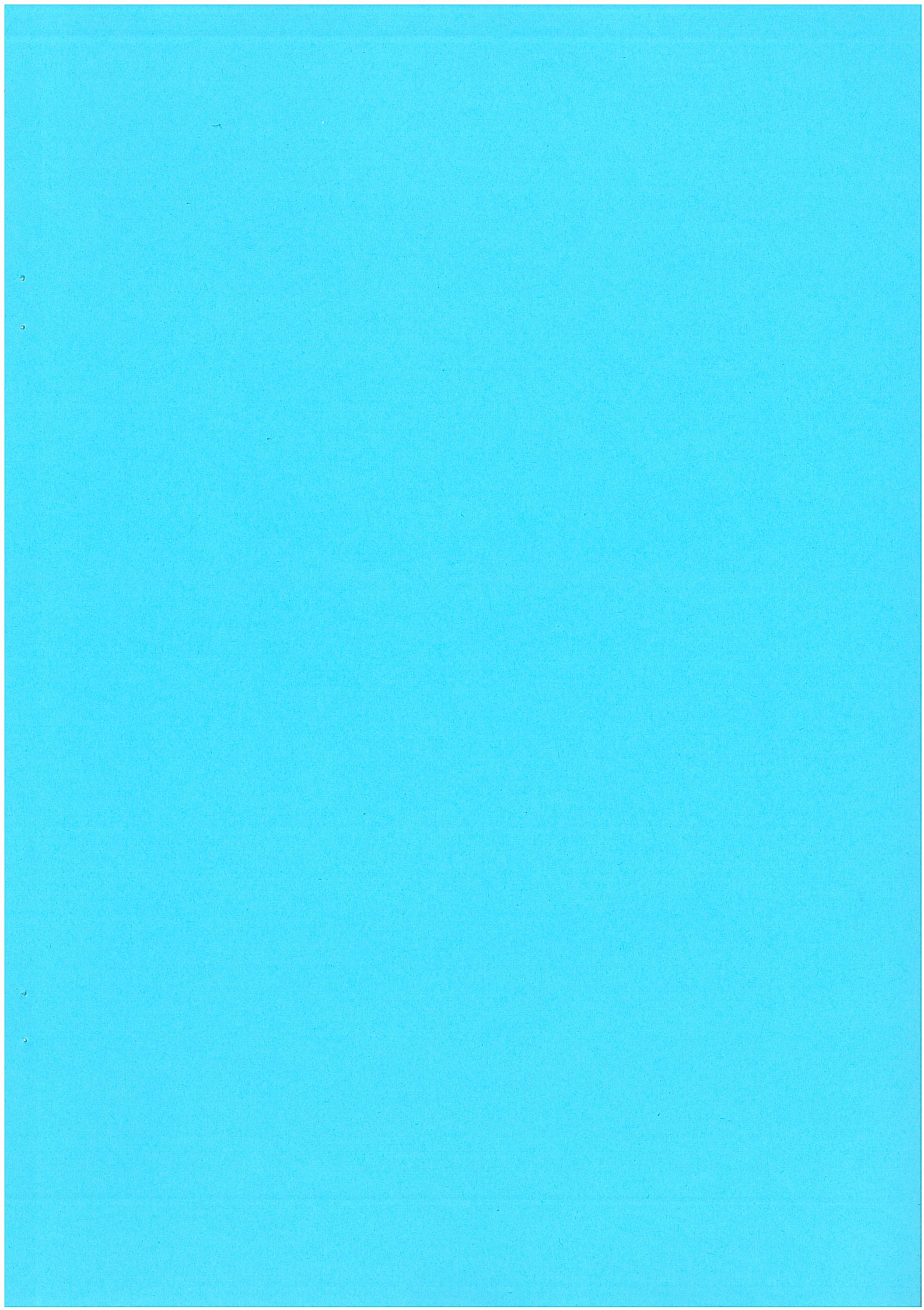
Lorsque la machine universelle rencontre le code de l'instruction FIN de cette procédure insérée, il lui faut retourner au code qui suit celui de son insertion. C'est ce que fait cette séquence:

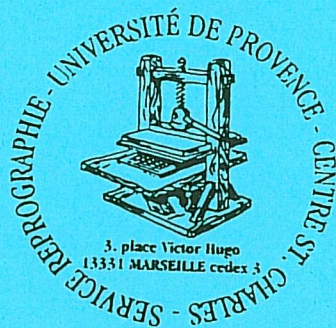
FIN : $[3 := [4,2$
 $[4 := [4 - [4,1$
 $[7,0 := 'fin'$
 vers ENTREE

Dans la troisième case de la configuration actuelle on récupère l'adresse de ce code qui y avait été placé par INSER, de même que le retour dans la configuration précédente contenu dans la deuxième case. En plus, comme je l'ai déjà signalé, la machine universelle informe le système qu'une procédure vient de s'achever.



Der Professor
Ft.
791





Université de Provence
Atelier de Reprographie
Centre Saint Charles
3, place Victor Hugo
F - 13331 Marseille Cedex 3