

OUI

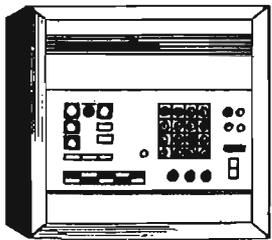
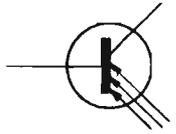


NON

1 + 1 = 10
10 + 10 = 100
1000 - 100 = 100
11 x 11 = 1001

ET

OU



INITIATION AU CALCUL ELECTRONIQUE

BASIC • ALGOL • FORTRAN

(Suite voir N° 1 330)

LES perspectives de l'emploi ne se présentent pas à l'heure actuelle sous un jour optimiste : les demandes non satisfaites suivent une courbe ascendante qui préoccupe les Pouvoirs Publics.

Un seul secteur échappe à cette ambiance maussade : l'informatique, dont le chiffre d'affaires, à l'issue de dix années d'expansion sauvage, représente plus de 10 milliards de francs, avec des effectifs dépassant 150 000 personnes. En 1975, le chiffre d'affaires de la profession s'élevait à 30 milliards de francs ; une telle évolution impliquerait la formation de 25 à 30 000 personnes par an au cours des prochaines années et de 50 000 à partir de 1975.

Il y aura, d'après les prévisions françaises, 18 000 ordinateurs en 1975 et 42 000 en 1980. Quant au software, il représente 8 % du chiffre d'affaires du secteur informatique.

Quant aux programmeurs, dont les annonces de journaux proposent des carrières alléchantes, ils ont un avenir incertain : les hypothèses avancées encore récemment (220 000 informaticiens dont 30 % de programmeurs) commencent à être remises en cause par l'apparition de langages nouveaux et softwares généraux.

LES LANGAGES

Un langage est un assemblage organisé de signes ; cet assemblage assure la communication entre l'homme et la machine, ou entre deux machines. L'ensemble de tous les signes est appelé un alphabet.

A titre d'exemple, le langage ALGOL est tout entier basé sur 116 de ces signes :

- 26 lettres majuscules
- 26 lettres minuscules
- 10 chiffres
- 24 symboles complexes tels que DEBUT, FIN, VRAI, JUSQU'A...
- Enfin des signes : +, -, x, /, =, >, <, ...

Il faut bien remarquer que les symboles complexes forment un tout : il s'agit d'éléments de l'alphabet que la machine connaît au même titre que les lettres et les chiffres. Par exemple, il faut voir dans DEBUT, non pas 5 lettres de notre alphabet classique, mais une seule et unique lettre de l'alphabet ALGOL. C'est là, l'une des premières difficultés que le débutant doit bien comprendre.

Ce ne sera malheureusement pas la seule !

Cet alphabet permet de donner des instructions à la machine :

« Lire sur la première carte perforée un nombre, que l'on appellera A ».

« Mettre A en mémoire ».

« Lire sur la seconde carte perforée un nombre, que l'on appellera B ».

« Ajouter B à A et mettre le résultat en mémoire ».

L'ensemble des instructions que la machine accepte constitue un langage.

LE PROGRAMME

Chaque instruction donnée à la machine sert à transformer l'état de la mémoire de la machine : on y introduit un nombre, on lit un nombre déjà en mémoire pour le transformer dans un registre de calcul (on lui ajoute, par exemple, un autre nombre), etc.

Il faut considérer la mémoire de l'ordinateur comme un assemblage de petits casiers ; une instruction consiste à ouvrir l'un des casiers et à transformer son contenu.

Chaque casier porte le nom de registre.

Chaque casier est repéré par son adresse.

Une instruction peut jouer sur plusieurs registres ; ainsi, pour faire l'opération toute simple $A + B = C$, il faudra pouvoir dire à la machine ce qui suit :

« Prendre le contenu du casier A ».

« Mettre ce contenu dans un registre spécial d'accumulation S ».

« Prendre le contenu du registre B ».

« Le mettre dans le registre S ».

« Mettre le contenu du registre S dans le casier C ».

Dans ce casier spécial d'accumulation, que nous avons appelé S, on fait toutes les additions. Chaque fois qu'on y met un nombre, celui-ci s'ajoute automatiquement au contenu précédent du registre.

Par contre, dans les casiers A, B, C, l'introduction d'un nombre efface automatiquement le contenu précédent de ces registres.

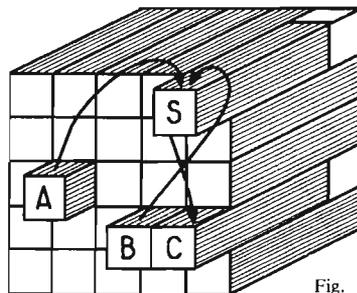


Fig. 1

La machine ne comprend pas notre langage ; nous allons donc coder chaque opération.

Prenons donc le code suivant : 001000 sera le code de l'instruction : « transfert du contenu du registre A dans le registre d'accumulation S ».

001001 sera le code de l'instruction : « transfert du contenu du registre B dans le registre d'accumulation S ».

000011 sera le code de l'instruction : « transfert du contenu du registre d'accumulation S dans le registre C ».

Le programme à donner à l'ordinateur sera alors :

001000
001001
000011

Ce programme, dont le code est indiqué plus haut, permet à l'ordinateur d'effectuer une addition.

En général, un programme est exécuté en séquence : une instruction après l'autre. Mais sur certaines machines, on indique, après

chaque instruction, où il faut aller chercher la suivante.

Certaines instructions sont dites de saut (ou de branchement). Le saut peut être inconditionnel (il est effectué quel que soit le contenu de la mémoire) ou conditionnel (en utilisant un test).

Par exemple, adoptons le code suivant :

101000 : « transfert du contenu du registre A dans le registre d'accumulation S si le contenu de A n'est pas nul ; sinon passer à l'instruction suivante ».

101001 : « transfert du contenu du registre B dans le registre d'accumulation S si le contenu de B n'est pas nul ; sinon passer à l'instruction suivante ».

100011 : « transfert du contenu du registre d'accumulation S dans le registre C si le contenu de S n'est pas nul ; sinon passer à l'instruction suivante ».

000000 : « mettre zéro dans tous les registres ».

Le programme d'addition suivant :

000000
101000
101001
100011

contiendra un test à chaque instruction : la machine analysera le contenu de chaque registre, pour savoir s'il faut ou non faire un transfert.

Le langage précédent est le « langage machine », dont les inconvénients sont nombreux :

- Il est long et fastidieux.
- Il est difficile de le relire.
- Il est peu aisé de communiquer le programme à une tierce personne.
- Il varie d'une machine à l'autre.

Pour remédier à de tels inconvénients, on va utiliser la capacité de la mémoire de la machine en introduisant un programme qui se chargera de traduire un langage proche du langage humain en langage machine.

LANGAGE SYMBOLIQUE

Le premier perfectionnement

consiste à désigner les noms des registres de mémoire pour un symbole, et un autre symbole sera affecté aux opérations. Par exemple on pourra écrire :

ADD A, B, C

pour dire à la machine de mettre les contenus des registres A et B dans le registre d'accumulation, puis de transférer le résultat dans le registre C.

A partir de 1957, on a cherché à définir des langages plus proches du langage mathématique : IBM a fait apparaître son langage FORTRAN, dans lequel l'addition s'écrit :

$C = A + B$

D'autres langages sont apparus : l'ALGOL (Algorithmic Language) et le BASIC pour le calcul scientifique, le COBOL pour la gestion.

L'ALGOL

Il n'est guère possible de décrire ici complètement un langage. On se propose donc de donner des notions générales d'ALGOL, qui seront valables pour d'autres langages scientifiques. On fournira ensuite des notions de FORTRAN et de BASIC.

Définissons tout d'abord un nombre :

Un nombre est défini par :

- Une partie entière.
- Une partie décimale.
- Un facteur de cadrage.

Considérons, par exemple, le nombre 4 751,5. On peut écrire :

$$4\ 751,5 = 4,7515 \cdot 10^3$$

La partie entière est 4, la partie décimale est 7515 et le facteur de cadrage est 10^3 .

On aurait pu tout aussi bien écrire :

$$4\ 751,5 = 47\ 515,0 \cdot 10^{-1}$$

La partie entière est alors 47 515, la partie décimale est nulle et le facteur de cadrage : 10^{-1} .

En ALGOL :

- La virgule est remplacée par un point : 4 751.5
- Si le facteur de cadrage est 10^0 , il sera omis.
- S'il n'y a pas de partie décimale, on enlève le point : 47515
- Si la partie entière est nulle, on peut ne pas l'écrire : 0.47515 est identique à .47515.
- Un nombre est précédé du signe - s'il est négatif.

VARIABLES SIMPLES...

En ALGOL, chaque variable a un nom commençant obligatoirement par une lettre : X, X1, X1R, INTENSITE, CIRCUIT 1 sont des exemples de nom de variables.

Il faut prévenir la machine que, dans le cours du programme, telle ou telle variable sera utilisée : toute variable doit être **déclarée** avant d'être utilisée. Au début du programme, on écrit alors :

DEBUT

REEL INTENSITE;

On constate que tout programme ALGOL débute par :

DEBUT

et qu'après la déclaration de la variable par **REEL**, on inscrit un point-virgule. S'il y a plusieurs variables à déclarer, on les sépare par des virgules et on ne met le point-virgule que lorsque la déclaration des variables est terminée :

DEBUT

REEL X, Y1, A, INTENSITE;

Si certaines variables M, N1, P, ne prennent que des valeurs entières dans tout le programme, on les déclarera par la formule :

ENTIER M, N1, P;

la déclaration est close par un point-virgule.

Sur une même ligne, on peut faire plusieurs déclarations, pourvu qu'elles soient séparées par un point-virgule :

DEBUT

ENTIER M, N; REEL X1, Y1, Z1;

... ET INDICEES

Une variable indicée s'écrit sous la forme :

$A[M, N]$

Suivant les valeurs prises par les deux variables entières M et N, A varie. M et N sont des indices, A est l'identification du tableau $A[M, N]$.

Soit n le nombre d'indices.

On dit que le tableau a pour dimension : n.

Un tableau de dimension 1 est un vecteur.

Un tableau de dimension 2 est une matrice.

Il est bien évident que les valeurs prises par A peuvent être ou non, entières.

Par exemple, soit à calculer, dans un programme la viscosité de l'air. La viscosité de l'air pourra être entrée dans l'ordinateur sous forme d'un tableau, avec pour paramètres la température et la pression.

On écrira VISCOSITE [TEMPERATURE, PRESSION], et à chaque valeur **entièr**e des indices TEMPERATURE et PRESSION, correspondra une valeur de viscosité.

Toute variable indicée est déclarée, en début de programme, en indiquant le domaine de variation des indices :

DEBUT

ENTIER TABLEAU VISCOSITE [300 : 400, 1 : 5]

Le tableau considéré est une matrice dont le premier indice (TEMPERATURE) varie, par valeurs entières, de 300 à 400, et le second indice (PRESSION), toujours par valeurs entières, varie de 1 à 5.

Dans le programme, on pourra utiliser la variable :

VISCOSITE [350, 4]

mais si l'on écrit VISCOSITE [250, 6] ou VISCOSITE [350.4, 4.5], la machine détectera une erreur.

On aurait éventuellement pu écrire :

DEBUT

REEL TABLEAU VISCOSITE [300 : 400, 1 : 5]

EXPRESSION

ARITHMETIQUE SIMPLE

Une expression arithmétique simple est une expression algébrique. Elle est construite par une suite de variables (indicées ou simples) et de nombres, séparés deux à deux par l'un des signes suivants :

- + addition
- soustraction
- * multiplication
- / division
- ÷ division entière (le résultat est la partie entière du quotient)
- ↑ élévation à une puissance

Pratiquement, on écrit des expressions arithmétiques simples comme on écrit des expressions mathématiques, ainsi :

$A + B * C^2 - (INT - 4) \uparrow B$ représente, en écriture ALGOL, ce que l'on écrit usuellement sous la forme mathématique suivante :

$$A + B.C^2 - (INT - 4)^B$$

On remarque qu'il y a possibilité d'introduire des parenthèses. Certains langages, autres que l'ALGOL, n'autorisent pas l'emploi de parenthèses : c'est le cas du programme connu sous le nom de MAGE et qui est employé avec l'ordinateur PALLAS.

INDICATEUR DE FONCTIONS

En Algol, il est possible de faire appel à un certain nombre de fonctions mathématiques standard :

sinus, cosinus, racine carrée,... On utilise des indicateurs de fonctions :

SIN pour le sinus
 COS pour le cosinus
 RAC2 pour la racine carrée
 LN pour le logarithme népérien
 ARCTAN pour arc tangente
 EXP pour l'exponentielle
 ABS pour la valeur absolue
 SIGNE pour signe
 ENTIER pour partie entière

On écrira, par exemple :
ENTIER (+ 3.4)

La machine calculera alors la partie entière de 3,4 et trouvera : 3.

Un indicateur de fonction sera constitué d'un nom de fonction suivi, entre parenthèses, du paramètre effectif (appelé argument de la fonction). L'argument est lui-même une expression arithmétique simple :

$$RAC2(B^2 - 4 * A * C)$$

AFFECTATION

Pour donner une valeur à une variable, on est amené à fournir une instruction d'affectation. En ALGOL, l'affectation s'écrit, d'une façon très générale, sous la forme :

VARIABLE :=

Expression arithmétique

Par exemple, on écrira, dans le cours d'un programme :

A := 3;

B := 5;

C := SIN(B^A);

X := RAC2(B^2 - 4 * A * C);

La machine commence par calculer l'expression arithmétique, avec les valeurs actuelles des variables, puis affecte, à la variable à gauche du signe d'affectation, la valeur de l'expression arithmétique.

De ce qui précède, il est évident que toutes les variables utilisées dans l'expression arithmétique à droite du signe d'affectation auront une valeur numérique, faute de quoi la machine ne peut pas calculer la valeur de l'expression arithmétique.

Dans ces conditions, on constatera que l'expression :

$$X := X + 1$$

a un sens autre que celui couramment utilisé en mathématiques.

En algèbre classique, l'équation $x = x + 1$ n'a pas de sens.

En ALGOL, x a initialement une valeur, par exemple x vaut 3.

La machine trouve l'instruction $x := x + 1$; elle calcule $x + 1$, et elle trouve $3 + 1 = 4$; ensuite, la machine dira que x vaut 4, et oubliera la valeur précédente.

C'est là un mode de calcul souvent rencontré en programmation. C'est dire son importance et surtout l'importance à bien comprendre la différence fondamentale qui existe entre le signe d'égalité et le signe d'affectation.

Le mois prochain, nous parlerons des branchements.

(à suivre)

Marc Ferretti.

