Collection du Centre d'Études Pratiques
d'Informatique et d'Automatique (CEPIA)

Emploi des ordinateurs

Introduction au SOFTWARE

par

Jean-Claude FAURE
Docteur-Ingénieur
Ingénieur consultant à la S.T.E.R.I.A.

avec la collaboration de Bernard LORHO
Docteur-Ingénieur
Ingénieur de recherche à l'I.R.I.A.

Préface de

Monsieur le Professeur Michel LAUDET
Professeur à l'Université Paul Sabatier de Toulouse
Président d'Honneur du CEPIA

Présentation de

René MALGOIRE
Vice-Président, Directeur du CEPIA

DEUXIÈME ÉDITION

DUNOD
Paris-Bruxelles-Montréal

DANS LA MÊME COLLECTION

P. DE MIRIBEL. — *Principes des ordinateurs.*

J. C. FAURE. — *Emploi des ordinateurs.*

M. BARES et H. DUCASSE. — *Cobol : Initiation et Pratique.*

Library of Congress Catalog Card Number 73-93450
N° d'Éditeur : 021 874 0303

ISBN 2-04-008828-8

PRÉFACE

Il m'est particulièrement agréable de préfacer ce livre.

Tout d'abord parce qu'il s'agit d'un ouvrage du C.E.P.I.A. que dirige avec compétence, autorité et dynamisme M. René MALGOIRE. La création de ce Centre, en 1968, répondait à la mission de formation de l'IRIA et s'adressait plus spécifiquement aux personnels dont le niveau était défini par les responsabilités qu'ils assumaient au sein des administrations ou des entreprises. Conjointement, trois autres tâches se sont ajoutées à cette mission : l'une, de recherche pédagogique, l'autre, de conseil et la troisième de formation des analystes en Informatique de Gestion.

Le succès actuel du C.E.P.I.A. a confirmé, d'une part, que la création de ce Centre correspondait à un besoin réel de formation et de sensibilisation à l'Informatique d'un nombre de plus en plus grand de personnels, d'autre part, que les enseignements qui y sont donnés étaient appréciés.

Ma satisfaction vient aussi du fait que J. C. FAURE l'auteur de cet ouvrage, et son collaborateur B. LORHO, ont été élèves de l'École Nationale Supérieure d'Électronique d'Informatique et d'Hydraulique de Toulouse (E.N.S.E.I.H.T.) et sont actuellement l'un Ingénieur consultant à la S.T.E.R.I.A., l'autre Ingénieur de recherche à l'I.R.I.A. Grâce à leur énergie et leur persévérance ils sont devenus d'excellents spécialistes du « Software ».

Ce livre de M. J. C. FAURE est destiné aux personnes qui, dans le domaine du « Software », désirent, soit compléter leur formation, soit acquérir des connaissances nouvelles. Pour ces derniers il constituera une excellente introduction à des ouvrages plus spécialisés.

Il s'agit essentiellement de poser le problème « Software », d'en donner une définition pragmatique et d'en montrer l'aboutissement sur des exemples concrets.

Cet ouvrage aborde un grand nombre d'aspects de cette discipline.

Le premier chapitre pose le problème de la communication entre l'homme et l'ordinateur. La formalisation d'un problème et la détermination de l'algorithme de résolution correspondant sont étudiés à partir d'exemples simples.

Les langages de programmation font l'objet du chapitre suivant. De courts programmes, cités en exemple, permettent de mieux assimiler les notions exposées et servent de base à une analyse critique.

Les principes généraux de fonctionnement des traducteurs sont ensuite étudiés. M. B. LORHO, qui a rédigé le chapitre, est un spécialiste de ce domaine; il a réussi à exposer très simplement, tout en conservant une très grande rigueur technique, les difficiles problèmes liés à la traduction des diverses classes de langages de programmation.

La notion de fichier est étudiée à l'occasion du chapitre 4. Les principales organisations sont exposées à propos d'un cas courant et simple. Une évaluation comparée des différentes méthodes de gestion de fichiers permet au lecteur, au prix bien sûr d'un effort sérieux, de s'initier aux techniques de l'évaluation des performances d'un programme.

Le dernier chapitre permet de suivre le déroulement d'un programme sous le contrôle d'un « moniteur ». Bien que l'étude soit relative à un cas particulier, les raisonnements sont suffisamment généraux pour que le lecteur soit en mesure de comprendre l'organisation et le rôle d'un système d'exploitation.

L'ouvrage de M. J. C. FAURE constitue une excellente initiation à l'étude du software. Mais, bien que s'adressant à un large public, sa lecture exige une attention soutenue.

Je souhaite le plus vif succès à ce livre. Je suis sûr qu'il permettra à la collection du C.E.P.I.A. de remplir pleinement l'un de ses objectifs, aider à la formation approfondie en l'Informatique.

Professeur à l'Université
Paul Sabatier de Toulouse
Président d'Honneur du C.E.P.I.A.

PRÉSENTATION
de la 2e édition

C'est un vif plaisir pour moi que de présenter la 2e édition de l'ouvrage de M. Jean-Claude FAURE car cela prouve que son travail a été favorablement accueilli par ses lecteurs. Qu'il en soit ici vivement remercié.

M. Jean-Claude FAURE, en collaboration avec M. Bernard LORHO, a bien voulu rédiger le cours de « Software » qu'il donne depuis de nombreuses années aux auditeurs du C.E.P.I.A. L'un et l'autre ont apporté, dans ce livre, le fruit de leur expérience d'ingénieur, de chercheur et d'enseignant. Et c'est précisément grâce à cela que leur travail peut intéresser les auditeurs du C.E.P.I.A. qui ont nécessairement besoin de suivre au plus près l'actualité et d'aborder ces matières à un niveau à la fois pratique, parce qu'ils ont des responsabilités réelles, et théorique parce que leur métier d'analyste les oblige à prendre le recul nécessaire. Or cet ouvrage est bien équilibré et devrait non seulement leur être utile, de ces points de vue, mais aussi leur donner la possibilité, dans un deuxième temps, d'aborder des textes plus complets au gré de leurs besoins et de leurs orientations.

L'ouvrage de M. FAURE s'inscrit donc bien dans la lignée des objectifs du C.E.P.I.A. (1). On sait en effet que le Centre, association de la loi de 1901, placé sous l'égide du Délégué à l'Informatique et de l'Institut de Recherche d'Informatique et d'Automatique, a pour vocation l'enseignement « pratique » de l'informatique de gestion. Il s'efforce donc, par ses interventions, de faciliter l'insertion de l'informatique dans le milieu qui en fait usage, d'améliorer la promotion des applications et, enfin, de rentabiliser au mieux les investissements correspondants.

---

(1) CEPIA - Centre d'Études Pratiques d'Informatique et d'Automatique, Domaine de Voluceau, Rocquencourt. 78150 - Le Chesnay. Tél. : 954-90-20 - 954-56-00 +.

Cette tâche est lourde, ceux qui vivent les développements actuels de l'informatique ne l'ignorent pas. Aussi des travaux tels que ceux mis en œuvre par MM. FAURE et LORHO ne peuvent que leur apporter une aide précieuse.

R. MALGOIRE
Administrateur des P.T.T.
Chargé de mission du Directeur
Général des Postes,
Vice-Président du C.E.P.I.A.

SOMMAIRE

I Introduction à l'informatique

A — UTILISATION D'UN ORDINATEUR

1 Principe d'un ordinateur

Comme toute machine, on peut considérer un ordinateur sous deux points de vue. Le point de vue constructeur nous apprendra quels sont les composants de cette machine et son fonctionnement interne. C'est ce que l'on désigne par « hardware ». Le point de vue de l'utilisateur considère le mode d'emploi de cette machine, ceci est désigné par « software » dans le langage des informaticiens (personnes s'occupant de l'informatique). L'informatique regroupe les préoccupations propres au « hardware » et au « software ». Dans la suite de ce livre, nous allons essentiellement nous intéresser au mode d'emploi d'un calculateur, c'est-à-dire au « software ».

Un ordinateur est une machine capable d'effectuer des opérations simples telles que des additions, des soustractions, des multiplications, des divisions, mais aussi des opérations non arithmétiques permettant, par exemple, de traiter les caractères d'un texte. Sur un ordinateur moderne, une opération élémentaire est effectuée en quelques micro-secondes (1 micro-seconde = 1 millionième de seconde) ou parfois moins. Le mode d'emploi habituel d'une machine (une caisse enregistreuse par exemple) est basé sur la manipulation de boutons en réponse à des indications fournies par l'intermédiaire de voyants ou de lampes. Une telle procédure ne permet pas, dans le cas d'un calculateur, une utilisation efficace. En effet, le temps de réponse d'un humain à un signal visuel est de l'ordre de la seconde, pendant ce temps l'ordinateur serait capable d'effectuer un million d'opérations. Pour profiter de cette vitesse « électronique », on sera amené à :

— décrire la suite des opérations à effectuer en prévoyant tous les cas possibles pour le problème à résoudre. Cette description se fera à l'aide d'un ensemble de symboles, de règles et de conventions appelé : langage de programmation.

— enregistrer dans la mémoire de l'ordinateur cette suite d'opérations ou programme (après une éventuelle traduction comme nous le verrons au chapitre 3).

— donner l'ordre au calculateur d'exécuter le programme qui se trouve dans sa mémoire. Les opérations à effectuer peuvent alors être fournies à une cadence suffisante, pour éviter toute attente entre deux ordres successifs.

Nous voyons donc que du point de vue utilisateur, un ordinateur se composera de deux parties essentielles : une unité centrale, qui est la partie de l'ordinateur capable d'effectuer un certain nombre d'opérations très rapidement, une mémoire centrale qui permet de stocker les séquences d'ordres et qui peut alimenter l'unité centrale très rapidement. Cette mémoire est divisée en cases (ou mots machine), chaque case est repérée par un numéro ou adresse.

Pour compléter notre vision schématique d'un ordinateur, nous devons rajouter un certain nombre de dispositifs d'entrée (lecteurs de cartes ou de rubans perforés...) permettant de rentrer des informations dans la mémoire centrale et des dispositifs de sortie (imprimantes, écrans de visualisation...) pour récupérer les résultats. Nous devons aussi signaler certains dispositifs de mémorisation (bandes, disques, tambours magnétiques...) qui permettent d'augmenter la taille de la mémoire centrale et que l'on appelle mémoires auxiliaires. On peut donc schématiser cette machine comme sur la figure 1-1. La description détaillée de la constitution et du fonctionnement de ces diffé-

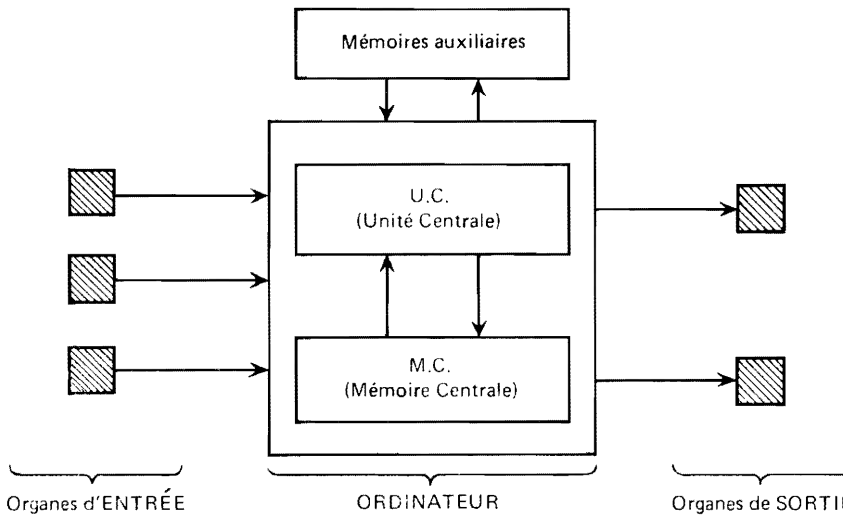

FIG. 1-1. — Schéma théorique d'un ordinateur.

rentes parties d'un ordinateur doit être examiné dans le cadre d'un livre sur le hardware (technologie des ordinateurs). Nous n'aborderons pas cette étude ici.

Nous avons introduit précédemment deux notions qu'il nous faut détailler un peu plus. Qu'est-ce exactement qu'un langage de programmation et un programme?

2 *Notions sur les langages*

Avant de parler des langages de programmation, nous allons rappeler quelques notions de linguistique, en prenant pour exemple notre langage naturel, c'est-à-dire le français.

a) Définition d'un langage

Nous allons essayer de définir un langage en représentant le processus que notre esprit effectue pour bâtir des phrases. Si nous considérons la phrase :

LE CHAT MANGE LA SOURIS

on peut la décomposer en ses constituants élémentaires : les mots. Les mots n'ont pas été placés au hasard dans cette phrase mais selon certaines règles. Par exemple, LE a été placé en tête de la phrase car c'est un article. La figure 1-2 montre les différentes étapes de construction de cette phrase exemple. Pour visualiser ces étapes, nous avons construit un arbre dont les feuilles sont les mots de notre phrase. Les nœuds de cet arbre représentent les entités plus ou moins complexes qui entrent dans la composition de l'entité la plus élaborée de l'arbre, sa racine.

Une construction de ce type porte le nom d'arbre syntaxique de décomposition. Les feuilles sont appelées éléments terminaux, car de ces éléments ne peut partir aucune branche. Les nœuds et la racine sont, par opposition appelés éléments non terminaux. La racine est un élément non terminal particulier du langage, car aucune branche ne peut y aboutir : on l'appelle l'axiome.

Tous les éléments terminaux et non terminaux composent le vocabulaire du langage. Il est nécessaire, bien évidemment, de pouvoir distinguer les éléments terminaux des éléments non terminaux; nous avons effectué cette distinction par une écriture en lettres majuscules pour les éléments terminaux.

Nous avons parlé de règles de construction d'une phrase. L'ensemble des règles qui vont permettre d'écrire correctement dans un langage donné, s'appelle la syntaxe de ce langage.

La formulation de ces règles peut se faire dans un langage que l'on nomme un métalangage (langage qui permet de définir d'autres langages). Les règles d'un métalangage s'appellent les règles de production.

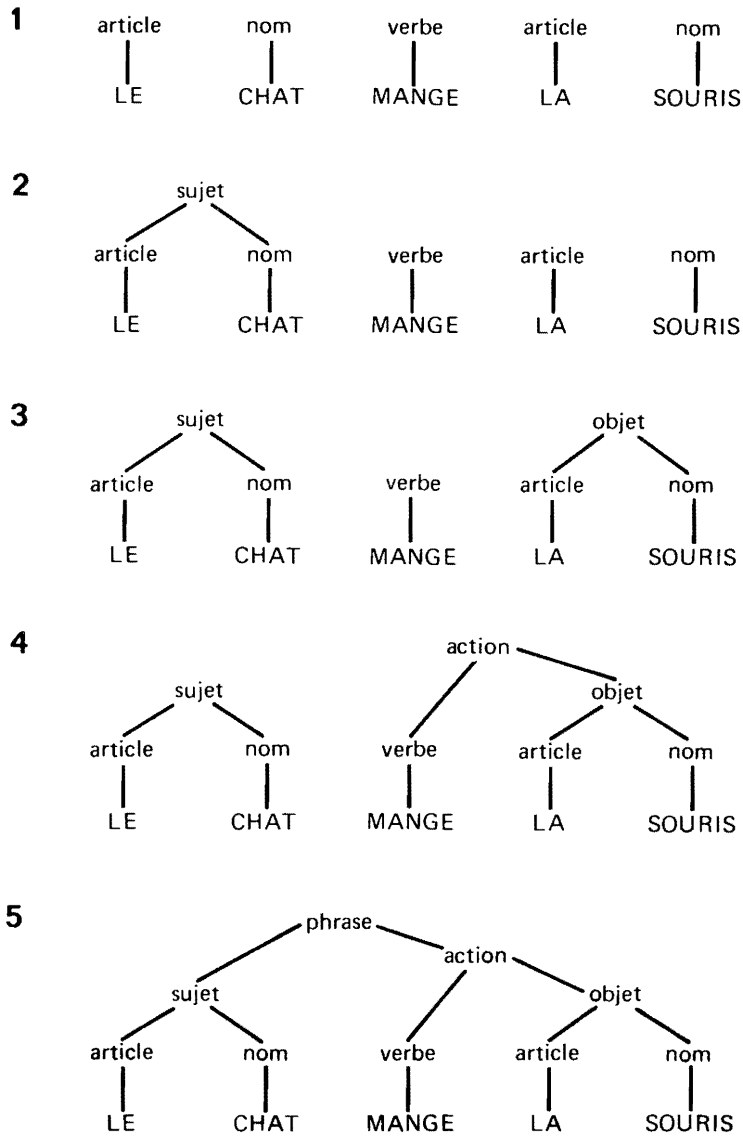

FIG. 1-2. — Étapes de construction d'une phrase.

Le métalangage le plus répandu est la Forme de Backus Naur.
Une règle de production se présente sous la forme suivante :

partie gauche : : = partie droite.

Il n'est pas dans notre propos de décrire complètement les syntaxes de tous

les langages. Pour les langages simples et en particulier les langages de programmation, la partie gauche d'une règle de production est un symbole non terminal et la partie droite une juxtaposition quelconque d'éléments non terminaux et d'éléments terminaux.

La figure 1-3 décrit la syntaxe de notre exemple.

phrase : : = sujet action
sujet : : = article nom
article : : = LE
nom : : = CHAT
action : : = verbe objet
verbe : : = MANGE
objet : : = article nom
article : : = LA
nom : : = SOURIS

Signification : Une règle de production, par exemple : sujet : : = article nom se lit : sujet EST DÉFINI par article SUIVI DE nom.

FIG. 1-3. — Description syntaxique d'une phrase.

On peut remarquer qu'il existe plusieurs règles ayant même partie gauche; par exemple, il y a deux règles ayant « article » pour partie gauche, de même pour « nom ». On peut regrouper ces deux règles en écrivant :

article : : = LE | LA
nom : : = CHAT | SOURIS

La barre verticale a le sens de OU. Par exemple :

article EST DÉFINI PAR « LE » OU « LA ».

La syntaxe d'un langage ne permet pas de définir complètement un langage. Elle permet seulement d'écrire des phrases correctes dans ce langage. Par exemple :

LA SOURIS MANGE LE CHAT
LA CHAT MANGE LE SOURIS

sont des phrases syntaxiquement correctes dans notre langage. Ces phrases ont-elles un sens? Ceci ne peut être déterminé par la syntaxe mais par la sémantique du langage que l'on peut définir comme l'ensemble des règles qui permettent de dire qu'une phrase syntaxiquement correcte a un sens.

On ne sait malheureusement pas décrire la sémantique d'un langage aussi simplement que la syntaxe. Il existe quelques langages qui permettent de décrire certains aspects sémantiques d'autres langages, mais on peut voir sur l'exemple :

LA SOURIS MANGE LE CHAT

que la difficulté est grande, si l'on veut dire que cette phrase n'est pas correcte sémantiquement car elle a un sens peu conforme à ce que le comportement respectif des chats et des souris laisse prévoir!

La description de la syntaxe d'un langage par un métalangage n'est pas uniquement une satisfaction intellectuelle. Elle a deux utilisations importantes. Tout d'abord, quelqu'un qui veut écrire dans un certain langage doit connaître la syntaxe de ce langage et, au besoin, s'y référer s'il a des doutes sur la validité de certaines constructions syntaxiques. D'autre part, comme nous le verrons au chapitre 3, un traducteur doit vérifier qu'un texte qui lui est soumis, suit bien la syntaxe du langage. Il existe des méthodes qui utilisent directement la description de la syntaxe du langage, pour réaliser cette fonction de vérification.

b) Langages de programmation

Un langage de programmation est un symbolisme permettant une communication entre l'utilisateur et l'ordinateur. Le langage naturel ne peut pas servir de langage commun car sa syntaxe est très complexe (multiplicité des règles de grammaire et des exceptions), sa sémantique aussi. Un langage de programmation sera donc plus restrictif, on le définira en fournissant à l'utilisateur, le vocabulaire autorisé, sa syntaxe et sa sémantique (cette sémantique peut s'exprimer sous des formes relativement simples). Pour ces langages, une phrase s'appelle une instruction et correspond à la description d'une ou plusieurs opérations élémentaires de l'ordinateur. Nous verrons dans la suite de ce chapitre quelles sont les instructions indispensables à un langage de programmation pour pouvoir traiter des problèmes réels.

3 *Notion de programme*

a) Applications d'un ordinateur

Un programme correspond à la traduction pour l'ordinateur d'un problème utilisateur. Nous allons donc essayer d'examiner rapidement les différentes applications possibles d'un calculateur.

1° Calcul scientifique

Historiquement, ce fut la première application effectuée à l'aide d'un ordinateur. On a utilisé la rapidité de calcul pour résoudre par des procédés itératifs (impliquant un nombre faible d'opérations différentes, mais se répétant de nombreuses fois) les problèmes mathématiques que l'on rencontre en algèbre numérique et non numérique. Dans ce cas là, l'ordinateur est considéré comme une très puissante machine à calculer.

2° Traitement de données

Ceci correspond aux problèmes de la gestion classique. On cherche donc à traiter automatiquement des applications comme :

— la facturation
— la gestion des stocks
— la paie du personnel
— la gestion d'une bibliothèque
— l'édition de rapports d'états, de catalogues et de statistiques.

3° Analyse de projets

Actuellement l'ordinateur peut intervenir à deux niveaux pour aider à l'analyse d'un projet. Il peut effectuer tous les gros travaux de calcul scientifique, mais, aussi intervenir d'une manière plus intéressante, pour aider à l'optimisation de la solution retenue.

Nous pouvons prendre comme exemple de cette utilisation, la participation d'un ordinateur au tracé d'une autoroute. On peut donner dans ce cas les différents paramètres du tracé et faire afficher sur un écran cathodique (écran de télévision) la vue correspondant à une route. Il existe même actuellement des programmes qui permettent de visualiser ce que le conducteur verra au volant de sa voiture, en parcourant l'autoroute. On espère ainsi détecter les points dangereux pour l'automobiliste et les supprimer en modifiant le tracé. Tout ceci se passe au niveau de la conception, avant que des travaux soient entrepris réellement.

4° Contrôle de processus (en anglais « Process Control »)

Le but est de surveiller et de commander à l'aide d'un ordinateur le fonctionnement de divers dispositifs industriels ou non. On peut citer comme exemple, le contrôle d'une chaîne de production dans une usine chimique. Dans le domaine médical, on peut surveiller un malade à l'aide d'un ordinateur, il avertira une infirmière si certains paramètres du patient prennent des valeurs anormales et donnera éventuellement des indications sur l'intervention à effectuer.

5° Gestion mathématique

Ces problèmes sont assez récents. Ils correspondent à l'application dans le domaine de la gestion, de certains résultats mathématiques (sur les arbres et sur les graphes par exemple). On peut citer en particulier la méthode PERT, qui permet, étant donné un certain nombre de tâches liées par des contraintes, de trouver le planning d'exécution optimum de celles-ci par rapport à un

critère donné. Par exemple, ayant déterminé différents travaux élémentaires à effectuer dans la construction d'un barrage, chaque travail durant un certain temps et ne pouvant démarrer qu'au moment où d'autres seront finis (par exemple on ne peut commencer les travaux réels que quand les routes d'accès sont terminées), il sera possible de trouver l'enchaînement des tâches qui minimise le temps de construction total de ce barrage. On peut rechercher aussi, en incorporant des paramètres sur la main d'œuvre et le matériel nécessités par chaque tâche, une optimisation du coût de construction.

6° Système d'information

Dans ce cas, l'ordinateur est utilisé pour fournir rapidement à l'utilisateur, des informations qui sont enregistrées dans ses mémoires. Le volume de ces informations peut être très important. Nous pouvons prendre comme exemple un système de réservation de places dans une compagnie aérienne. Chaque fois qu'une personne demande un billet, l'ordinateur peut fournir au préposé, le remplissage de l'avion correspondant au vol demandé. Il autorise donc une réponse très rapide et très précise. On peut espérer ainsi optimiser le remplissage des avions en tenant compte immédiatement des défections éventuelles de clients par exemple. De tels systèmes sont chargés de remplacer la consultation manuelle de fichiers très volumineux, les informations traitées peuvent être textuelles ou numériques.

b) Notion d'algorithme

Un algorithme est un ensemble de règles permettant de réaliser mécaniquement (tout au moins en théorie) toutes les opérations particulières correspondant à un certain type de travail.

Pratiquement, en informatique, un algorithme sera la décomposition en opérations élémentaires admissibles par l'ordinateur d'un problème utilisateur.

Un programme correspondra à la codification, suivant un langage de programmation donné d'un algorithme. Les programmes sont écrits suivant une convention séquentielle, c'est-à-dire que les différentes instructions les composant seront écrites les unes à la suite des autres, dans leur ordre d'exécution. Éventuellement, comme nous le verrons, certaines instructions peuvent permettre de violer ce principe séquentiel.

D'après sa définition, nous voyons donc qu'un programme aura un nombre d'instructions très variable selon la complexité de l'algorithme dont il résulte, c'est-à-dire selon le problème traité.

c) Programme de base, programme d'application

Nous avons vu les différents travaux susceptibles d'être traités par un ordinateur. Pour toutes ces applications, un certain nombre de problèmes

sont communs. Afin d'éviter que chaque utilisateur retraite tous les problèmes, on a été amené à faire la différence entre le « software de base » qui est livré par le constructeur d'un ordinateur et le « software d'application ». Le « software de base » se compose de tous les programmes ayant un intérêt universel. Ces programmes seront étudiés dans les chapitres 3, 4 et 5 de ce livre. Le « software d'application » comprend les programmes écrits par les utilisateurs pour résoudre des problèmes spécifiques, ils sont d'une généralité moindre.

Du point de vue utilisateur, le problème qui se pose est : étant donné une application particulière, comment réaliser le programme correspondant en se servant d'un ordinateur et des programmes de base qui lui sont associés?

Dans le paragraphe suivant, nous allons donner quelques notions sur l'établissement de l'algorithme correspondant à un problème, puis, dans le chapitre 2, nous étudierons les possibilités de codage suivant les différents langages de programmation disponibles.

B — ALGORITHMES ET PROGRAMMATION

I Mise au point d'un programme

Un ordinateur est une machine capable d'exécuter des instructions données par l'homme à une très grande vitesse. La seule « expérience » d'un ordinateur consiste en programmes de base. Pour réaliser le programme correspondant à un certain problème, on devra indiquer à l'ordinateur toutes les opérations à effectuer dans tous les cas que l'on juge possibles. La préparation d'un problème en vue de son passage sur ordinateur va nécessiter plusieurs étapes (fig. 1-4).

a) Définition exacte du problème

On ne peut pas espérer résoudre à l'aide d'un calculateur, un problème qui ne serait pas complètement défini. Il faudra en particulier définir très clairement les objectifs que l'on souhaite atteindre et leurs limitations éventuelles. Cette étude fait partie de la phase d'analyse d'un problème, elle dépend étroitement de l'application que l'on veut traiter. Elle doit être effectuée par des spécialistes de l'application choisie ayant quelques notions d'informatique. Nous n'étudierons pas, ici, cette analyse générale d'un problème.

b) Choix de la méthode de résolution

La phase précédente a permis de déterminer l'allure générale du problème, il reste à déterminer quelles sont les méthodes de résolution possibles. En effet,