

IRIA

CPL/1

1^{ère} partie :
**ANALYSEUR
GÉNÉRATEUR**



**sept. 1971
cahier n° 6**

Institut de Recherche
d'Informatique et d'Automatique
domaine de Voluceau
78 - Rocquencourt
France

BIBLIOTHEQUE DU CERIST

IRIA

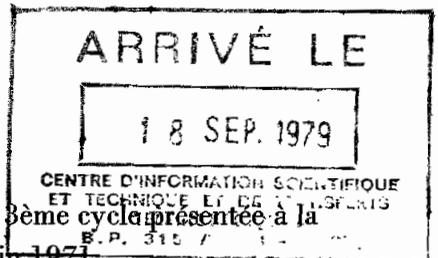
CPL/1

1ère partie :

ANALYSEUR-GÉNÉRATEUR*

par

B. LORHO et M. VATOUX



*Ce mémoire a fait l'objet d'une thèse de 3ème cycle présentée à la
 Faculté des Sciences de Toulouse le 7 juin 1971.

BIBLIOTHEQUE DU CERIST



PREFACE

BIBLIOTHEQUE DU CERIST

L'informatique s'est développée, au cours des vingt dernières années de son existence, plus comme une technique que comme une science, et l'on a pu voir, dans ces premiers temps, des mathématiciens connus se pencher sur les problèmes très concrets de la construction de nouveaux calculateurs.

Progressivement, cependant, un souci de rigueur et de méthode a succédé à l'empirisme initial et il en est résulté, dans certains domaines au moins, des progrès spectaculaires. Une formalisation appropriée a permis de trouver un fil conducteur dans la masse des astuces qui servaient jusque là de connaissances au spécialiste informaticien, et certains travaux complexes ont ainsi pu être réduits à l'application de méthodes de routine, tandis que des applications jugées inaccessibles devenaient tout à coup possibles.

L'un de ces domaines privilégiés où prend corps une informatique théorique concerne les langages de programmation. Au moment où l'IRIA se créait, la théorie des langages " context free " était bien établie, et l'étude méthodique de sous-classes des CF avait conduit à dégager toute une série de procédés d'analyse syntaxique appuyés sur un solide arrière plan mathématique, tout en manifestant bien entendu l'étendue des lacunes résiduelles.

Il était donc naturel que l'un des axes de recherche de l'IRIA s'orientât vers ce domaine passionnant et riche. Il fut décidé de démarrer un projet de recherche que nous pouvons baptiser " méta-langage " , avec les objectifs suivants:

dans un premier temps, former une équipe à l'occasion d'un problème concret de réalisation de compilateur, en vue de dégager clairement les principes et les méthodes de l'analyse syntaxique sans perdre de vue les préoccupations pratiques d'emploi.

- ensuite, de s'attaquer à la genèse automatique d'analyseurs syntaxiques pour des classes de plus en plus larges de langages pratiquement utilisables.

- et simultanément, définir des méthodes pratiques pour automatiser la description de la sémantique associée à une syntaxe du type précédent, dans l'espoir d'automatiser aussi la phase de génération.

Ce ne sont pas là des sujets vierges, heureusement d'ailleurs, car le domaine est si vaste que le travail de nombreuses équipes ne suffira pas à l'épuiser avant des années.

Mais l'apport de l'équipe de l'IRIA, solidement ancrée dans la réalité par un objectif concret et néanmoins au fait des plus récents travaux dans ce domaine, représente une part importante de l'effort de recherche français.

Le présent volume est consacré à la première phase du travail, la construction effective d'un compilateur conversationnel pour le langage PL/1. Le choix de PL/1 ne résulte pas d'un hasard: PL/1 est le premier langage à ambitions universelles, conçu par des programmeurs pour des programmeurs, et recouvrant une variété de concepts assez grande pour satisfaire toutes leurs exigences. Faire un compilateur pour PL/1, assez efficace pour que les programmeurs s'en servent, est donc un travail difficile; et dans ce compilateur, l'analyseur syntaxique doit reconnaître une grande variété de formes syntaxiques qui se rattachent souvent assez mal à l'un ou l'autre des modèles connus.

Ainsi, en choisissant de réaliser comme première étape du projet un compilateur PL/1, l'IRIA d'une part s'assurait un outil capable de répondre par la suite aux besoins les plus ambitieux de ses chercheurs, et d'autre part donnait à l'équipe de recherche sur les langages un objectif ambitieux et l'occasion d'approfondir toutes les recherches antérieures en cette matière.

Aujourd'hui, le CPL/1 fonctionne, et sa mise au point a effectivement permis la création d'une équipe solide, qui s'est sérieusement attaquée à la seconde phase, mais qui saura discerner le point où une trop belle idée théorique doit être infléchie pour servir à quelque chose dans la réalité quotidienne.

MM. LORHO et VATOUX, qui animent cette équipe, ont commencé à Toulouse leur carrière de chercheurs en écrivant un compilateur ALGOL qui est aujourd'hui en service sur l'ordinateur CII 10.070. L'analyseur syntaxique du CPL/1 leur a fourni leur sujet de thèse de 3^e cycle et constitue l'objet de ce premier volume, tandis que le second décrira plus particulièrement l'interpréteur et les mécanismes d'allocation de mémoire associés à l'exécution des programmes.

L'IRIA se félicite d'avoir réalisé avec CPL/1, un outil de programmation réellement pratique, et constituant une contribution concrète au plan calcul français.

H. BOUCHER

Directeur scientifique

TABLE DES MATIERES

0. INTRODUCTION GENERALE	7
1. MODES DE COMPILATION	9
1.1 Compilation classique	9
1.2 Compilation incrémentale	10
1.3 Organisation choisie	11
2. LANGAGE SOURCE	13
2.1 Choix du langage	13
2.2 Description du sous-ensemble	14
2.3 Restrictions entraînées par la méthode	23
3. ANALYSEUR SYNTAXIQUE.CHOIX D'UNE METHODE	27
3.0 Généralités. Terminologie	27
3.1 Définitions	27
3.2 Analyse syntaxique descendante	29
3.3 Analyse syntaxique ascendante	31
3.4 Conclusions.Choix d'une méthode mixte	33
4. ANALYSEUR LEXICOGRAPHIQUE	35
4.1 Généralités	35
4.2 Principe	35
4.3 Instruction TTBS	37
4.4 Organisation	38
5. ANALYSEUR DESCENDANT	41
5.1 Généralités	41
5.2 Diagrammes de transition	41
5.3 Machine d'analyse	45
5.4 Représentation de Kahan	48
5.5 Inclusions sémantiques	50
5.6 Tests de début d'alternatives	51
5.7 Analyseur	53
5.8 Générateur	57
5.9 Conclusions	58

6. INTRODUCTEUR DE SYNTAXE	59
6.0 Généralités	59
6.1 Métalangage-représentation	59
6.2 Exemples	61
6.3 Limites du métalangage	61
6.4 Principes de fonctionnement	62
6.5 Métasyntaxe	63
6.6 Conclusions	64
7. ANALYSEUR ASCENDANT	67
7.0 Principe	67
7.1 Syntaxe- Exemple	67
7.2 Choix d'une méthode ascendante	68
7.3 Méthode des matrices de transition	68
7.4 Réalisation- Extension	74
7.5 Connexion avec l'analyseur descendant	75
7.6 Réalisation-Conclusions	76
8. TRAITEMENT DES ERREURS	77
8.0 Problème	77
8.1 Instructions de traitement d'erreur	77
8.2 Conclusions	79
9. TRAITEMENT DES ATTRIBUTS	81
9.1 Instruction DECLARE	82
9.2 Compatibilité, incompatibilité, implication	84
9.3 Traitement des attributs	85
9.4 Connexion avec l'analyseur descendant	94
10. TRAITEMENT DES STRUCTURES	101
10.1 Définitions	101
10.2 Déclaration de structure	105
10.3 Qualification	108
11. TABLES D'IDENTIFICATEURS	111
11.0 Problèmes	111
11.1 Quelques solutions	111
11.2 Organisation choisie	113
11.3 Traitement des mots-clés	117
11.4 Conclusions	119
BIBLIOGRAPHIE	121
ANNEXES	123

INTRODUCTION GENERALE

Le projet CPLI a débuté en octobre 1968 . Il a été motivé par un besoin d'offrir aux chercheurs de l'IRIA certaines facilités de mise au point de leurs programmes, tant au cours de leur rédaction qu'au cours de leur exécution .

Ce double souhait impose un traitement conversationnel que l'on peut scinder en deux parties :

- le programme source est introduit et traité ligne par ligne . Chaque ligne est analysée le plus complètement possible afin de détecter le maximum d'erreurs .
- une exécution d'une ou plusieurs lignes peut intervenir à tout moment et il faut qu'elle apparaisse comme étroitement liée à la forme du programme source .

Nous allons décrire complètement la partie analyse du texte source. La partie exécution et conversation sera décrite par ailleurs .

Ce travail n'est envisageable que grâce à un système d'exploitation conversationnelle . Le système ESOPE [1] dont l'étude et la réalisation ont débuté parallèlement à ce travail offre les plus larges facilités dans ce domaine .