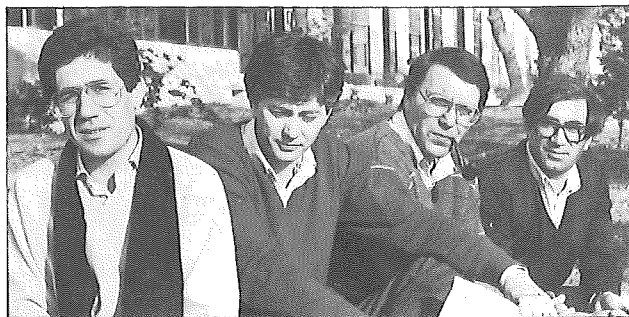


Prolog

FRANCIS GIANNESINI HENRY KANOUI ROBERT PASERO MICHEL VAN CANEGHEM
Préface de ALAIN COLMERAUER

Prolog (Programmation en Logique) est un langage de programmation de type déclaratif, conçu en 1972 au Groupe Intelligence Artificielle de Marseille - Luminy et placé au cœur de ce qu'il est convenu d'appeler le projet d'ordinateurs de cinquième génération. Destiné aux applications de l'intelligence artificielle: langues naturelles, démonstration automatique, reconnaissance de la parole, systèmes experts, Prolog est particulièrement bien adapté au traitement des problèmes non numériques comme la compilation. Ce livre, qui est le fruit de dix ans d'expérience d'enseignement, comporte deux parties. Dans la première, les auteurs exposent les principes d'utilisation du langage, en procédant, sur de nombreux exemples, par raffinements successifs d'un programme initial. Cette progression leur a permis d'exposer les diverses possibilités du langage, et surtout de donner au débutant les principes de la bonne programmation en Prolog. Une deuxième partie est consacrée aux applications, où sont abordées, à un niveau élevé, diverses applications de l'Intelligence Artificielle.



Les auteurs sont enseignants-chercheurs au Groupe Intelligence Artificielle de Luminy de l'Université Aix-Marseille II. Ils ont tous, à des titres divers, participé à la création, au développement et à la diffusion du langage Prolog et particulièrement de la version Prolog II utilisée dans ce livre.

La version Prolog II est disponible sur les matériels suivants:
Apple II, Apple Lisa, Macintosh ; IBM-PC et compatibles sous MS-DOS 2 ;
Hewlett-Packard (HP-150, HP-1000, HP-9000) ; VAX/VMS.
Elle est commercialisée par la société PrologIA, 278 rue Saint-Pierre,
13005 Marseille (France).

 **InterEditions**
87 AVENUE DU MAINE 75014 PARIS

ISBN 2 7296 0076 0

Prolog

F. GIANNESINI H. KANOUI R. PASERO M. VAN CANEGHEM

ATELIER PATRICK O'HEGUERTY

LC Call n.
QA76



60076

Prolog

F. GIANNESINI
H. KANOUI
R. PASERO
M. VAN CANEGHEM

PREFACE DE
A. COLMERAUER

INFORMATIQUE
ia
INTELLIGENCE ARTIFICIELLE

InterEditions

Prolog

F. GIANNESINI
H. KANOU
R. PASERO
M. VAN CANEGHEM

PREFACE DE
A. COLMERAUER

*Publié avec le concours
du Centre National des Lettres*



InterÉditions
87 AVENUE DU MAINE 75014 PARIS

Ce livre a été directement composé par les auteurs au moyen du logiciel \TeX sur un ordinateur VAX 730 et une imprimante LN01.

SOMMAIRE

Préface	9
Avant-propos	11
I Présentation informelle	
1. Premier exemple: au restaurant	16
2. Où l'on revoit les choses de plus près	25
2.1. Les arbres	25
2.2. Contraintes sur les arbres	27
2.3. Equations sur les arbres	27
2.4. Inéquations sur les arbres	29
3. Structure d'un programme Prolog	31
3.1. Le principe d'effacement	31
3.2. Les règles prédéfinies	34
3.3. Le fameux "/"	34
4. Récapitulation	37
Enoncés des exercices	39
II Listes	
I Listes et suites finies	
1. Codage d'une liste	43
2. Propriété fondamentale	44
3. Accès aux éléments d'une liste	46
3.1. Accès à un élément connu	49

©1985, InterÉditions, Paris

Tous droits réservés. Aucun extrait de ce livre ne peut être reproduit, sous quelque forme ou par quelque procédé que ce soit (machine électronique, mécanique, à photocopier, à enregistrer ou toute autre) sans l'autorisation écrite préalable de l'Éditeur.

ISBN 2-7296-0076-0

3.2. Enumération des éléments d'une liste	51
3.3. Deuxième version du programme <i>menu</i>	53
4. Récursivité	54
5. Construction d'une liste	59
5.1. Construction directe d'une liste	58
5.2. Construction au moyen d'une liste auxiliaire	62
5.3. La concaténation de deux listes	66
6. Conclusion	69
II Listes et sous-suites	
1. Arbres représentatifs	70
2. Systèmes d'équations	72
3. Troisième version du programme <i>menu</i>	73
III Application: les analyseurs	
1. Approche naive	76
2. Approche par les graphes	80
3. Conclusion	84
Enoncés des exercices	87
III Termes préfixés - N-uplets	
I Termes préfixés	
1. Codage	91
2. Exemple d'utilisation: calcul booléen	92
3. Première solution: approche naturelle	93
4. Amélioration du programme	97
5. Nouvelle amélioration et version finale	99
6. La négation	101
6.1. Une autre approche de <i>valeur-booleenne</i>	101
6.2. Les règles de la négation	103
6.3. La double négation	105
6.4. Négation et vérité	106
7. Conclusion	107
II Les N-uplets	
1. N-uplets et ensembles	108
2. N-uplets et termes préfixés	108
3. Exemple d'utilisation: arbres syntaxiques	110
3.1. Reconnaissance d'un arbre syntaxique	111
3.2. Génération des arbres syntaxiques	112
3.3. Analyse et arbre syntaxique	112
4. Liste des feuilles d'un arbre	114
4.1. Approche directe	114

4.2. Modification du programme	115
5. Arbres infinis	116
Enoncés des exercices	120
IV Les règles prédéfinies de Prolog II	
1. La syntaxe de Prolog II	124
1.1. La syntaxe des variables et des identificateurs	124
1.2. Les symboles fonctionnels	124
1.3. La syntaxe des règles	125
2. Le contrôle de l'effacement des buts	127
2.1. Le "slash"	129
2.2. Block, block-exit	130
2.3. Bound, free	132
2.4. Dif	132
2.5. Freeze	135
2.6. Les arbres infinis	137
3. Les entrées et sorties	137
4. Opérations prédéfinies sur les données	144
4.1. Les tests de type	144
4.2. Les opérations arithmétiques	144
4.3. Opérations sur les chaînes	148
4.4. Composition et décomposition d'objets	149
5. Structuration et modification des règles	151
6. L'environnement	155
7. La mise au point des programmes	157
7.1. Pas de résultat	158
7.2. Programme qui boucle	159
7.3. Un résultat faux	160
7.4. Trop de résultats	162
V Applications	
I Logique et banques de données	
1. Une banque de données sur les liens de parenté	164
2. Interrogation par évaluation de formules	169
2.1. Définition syntaxique	169
2.2. Définition sémantique	170
2.3. Calcul de la valeur sémantique: première façon	170
2.4. Calcul de la valeur sémantique: deuxième façon	175
2.5. Définition des questions	177
2.6. Définition des réponses	177
2.7. Quelques exemples d'interrogation	178

II	Analyse et synthèse d'un langage	
1.	Application aux langues naturelles	183
1.1.	Définition des structures syntaxiques	183
1.2.	Définition des structures sémantiques	192
2.	Réalisation d'une interface en français	200
III	Arbres infinis et compilation	
1.	Une grammaire des grammaires	208
2.	L'arbre infini associé à une grammaire	212
3.	L'analyseur associé à une grammaire	216
4.	Le compilateur associé à une grammaire	216
5.	Conclusion	219
IV	Démonstration automatique et systèmes experts	
1.	Logique propositionnelle: terminologie	221
2.	Forme clausale	221
3.	Notion d'interprétation	222
4.	Principe de résolution	223
5.	Saturation	223
6.	SL-résolution	224
7.	Application aux systèmes experts	225
8.	Un exemple de système expert réalisé en Prolog	249
	Solutions des exercices	253
	Annexe A La syntaxe de Prolog II	293
	Annexe B Les principales règles prédéfinies de Prolog II	295
	Annexe C Correspondance entre Prolog II et Micro-Prolog	301
	Annexe D Correspondance entre Prolog II et Prolog Edimbourg	307
	Bibliographie	313
	Index	315

PREFACE

Le groupe de recherche en Intelligence Artificielle de Marseille est célèbre pour avoir conçu Prolog, mais célèbre aussi pour avoir très peu publié sur ce sujet. Ce livre comble cette dernière lacune et je félicite les quatre auteurs de nous présenter le savoir qu'ils ont accumulé, digéré et enseigné pendant dix ans.

On demande souvent ce qui différencie Prolog des autres langages de programmation et d'où vient sa "soi-disant" puissance. Prolog est né d'un pari : créer un langage de très haut niveau, même inefficace au sens des informaticiens de l'époque. L'efficacité consistait alors à faire exécuter très rapidement par une machine, des programmes écrits laborieusement. Le pari était donc de pouvoir écrire rapidement des programmes, quitte à ce que la machine les exécute laborieusement.

Une fois libérés de cette obsession d'efficacité, on a pu faire appel aux formalismes et aux mécanismes d'inférences de la logique mathématique, mécanismes inefficaces mais tellement puissants ! C'est de là que Prolog tire son nom : "programmation en Logique". Maintenant, si l'on regarde ce qui se passe au niveau de la machine, on constate que celle-ci doit être beaucoup plus intelligente que d'habitude : elle doit être non-déterministe, c'est-à-dire capable d'explorer plusieurs possibilités et qu'elle doit pouvoir résoudre des milliers et des milliers de petites équations.

Ces équations font intervenir des inconnues qui ne sont rien d'autre que les variables du programme. On est loin des variables habituelles qui désignent des adresses dans la machine ! Le programmeur qui passe d'un langage classique à Prolog, éprouve à peu près la même révélation que l'écolier passant de l'arithmétique aux premiers rudiments de l'algèbre. Il peut désigner les entités dont il cherche les valeurs par des inconnues, établir un certain nombre de relations entre ces inconnues, et, sans avoir à détailler, laisser la machine considérer tous les cas de figure possibles et lui fournir toutes les solutions possibles ...

Que demander de plus ?

Alain Colmerauer

AVANT-PROPOS

C'est en 1981 que les projets internationaux de développement de l'Intelligence Artificielle ont brusquement projeté sur le devant de la scène un langage de programmation d'un type nouveau, Prolog, fondé sur la logique mathématique.

En fait, Prolog est né en 1972 au Groupe d'Intelligence Artificielle de Luminy (Marseille), dirigé par Alain Colmerauer, et n'était jusque-là connu que du cercle étroit de quelques chercheurs. Aujourd'hui, ce langage est largement répandu, non seulement dans les universités et les laboratoires de recherche, mais aussi, et de plus en plus, au sein du monde industriel.

Dix ans d'enseignement de Prolog nous ont convaincus que ce langage était à la fois facile et difficile. En effet, le débutant parvient très vite à écrire ses premiers programmes, souvent spectaculaires. Mais la réalisation de gros programmes nécessite une rigueur de conception et une maîtrise du langage difficiles à acquérir. Ainsi le débutant est-il très vite désorienté, d'autant que la littérature sur ce sujet est très rare.

C'est cette lacune que nous avons voulu combler en écrivant cet ouvrage qui est non seulement un manuel de programmation, mais aussi un guide exposant notre expérience de la "bonne" programmation en Prolog. Dans ce but, nous avons choisi, ayant posé un problème, de donner la solution la plus naturelle d'abord, de critiquer cette solution, et, par raffinements, au travers de plusieurs essais successifs, de parvenir à la "meilleure" solution. Nous espérons ainsi que le lecteur ne sera

pas noyé sous un ensemble de règles et de recettes indigestes! Dans le même esprit nous avons abordé les concepts fondamentaux de Prolog de manière intuitive, laissant volontairement de côté l'étude du modèle théorique que le lecteur intéressé trouvera dans les articles cités dans la bibliographie.

Pour cet exposé, il a bien fallu se référer à une implantation précise de Prolog : il existe en effet plusieurs versions du langage qui diffèrent entre elles par la syntaxe et les fonctions primitives offertes. Nous avons choisi Prolog II, qui a été développé au GIA de Luminy. Cette version présente l'avantage de posséder un modèle théorique rigoureux, une syntaxe claire et des primitives originales qu'on ne trouve rassemblés nulle part ailleurs. Néanmoins, nous nous sommes efforcés de ne pas lier les trois premiers chapitres qui exposent les bases de la programmation en Prolog à une version précise. Les exemples cités pourront être transcrits sans difficulté dans n'importe quelle implantation du langage. D'ailleurs, les syntaxes des différents Prolog sont assez voisines et nous donnons en annexe des procédés qui permettent de passer de l'une à l'autre.

Le livre est divisé en cinq chapitres.

Le chapitre I est destiné à familiariser le débutant avec les concepts fondamentaux et les mécanismes de base de Prolog, de façon à lui permettre d'écrire ses premiers programmes. Le lecteur est guidé par un exemple concret, enrichi et complété au fur et à mesure de la progression de l'exposé.

Les chapitres II et III introduisent les structures complexes manipulées par Prolog: listes et arbres. A titre d'exemples, sont introduites les premières notions de traitement d'objets formels et d'analyse syntaxique qui seront longuement développées au chapitre V.

Comme tout langage, Prolog doit pouvoir communiquer avec son environnement. Cela s'effectue au moyen des "règles prédéfinies" qui sont étudiées au chapitre IV. Ce chapitre est spécifique de Prolog II. Cependant, une grande partie des règles prédéfinies qui y sont exposées existent sous une forme ou une autre dans la plupart des versions.

Enfin le chapitre V est consacré aux grandes applications de Prolog, à savoir la résolution de divers problèmes d'Intelligence Artificielle : traitement du français, banques de données structurées, systèmes experts, ou problèmes classiques comme la compilation. Ce chapitre ne doit pas être abordé sans une très bonne connaissance du langage, car les méthodes qui y sont exposées relèvent parfois du domaine de la recherche. C'est à dessein que nous y avons inclus de gros programmes, conscients du fossé qui sépare des exercices à but pédagogique, comme ceux du début de l'ouvrage, d'une application réelle. Il nous a été souvent impossible

de commenter en détail la totalité de ces programmes; aussi, nous nous sommes attachés à éclairer les parties les plus importantes et nous avons laissé au lecteur le soin de justifier lui-même les parties les plus simples. Le livre est complété par des exercices et leur solution commentée, par diverses annexes, un index et une bibliographie.

Conventions typographiques

Tout ce qui renvoie à un programme Prolog est écrit en *caracteres penches*.

Les commandes qui déclenchent l'exécution d'un programme Prolog sont précédées du caractère ">".

Les résultats affichés par l'exécution d'un programme sont écrits en *caracteres teletype*.

Bien entendu, programmes et résultats ne comportent pas de caractères accentués. Signalons enfin, pour le regretter, que les impératifs de diffusion internationale nous ont contraints à utiliser l'anglais pour les mots réservés et les messages d'erreur de Prolog II.

Adresse des auteurs :

Groupe Intelligence Artificielle

Faculté des Sciences de LUMINY Case 901

70, Route Léon Lachamp

13288 MARSEILLE Cedex 9