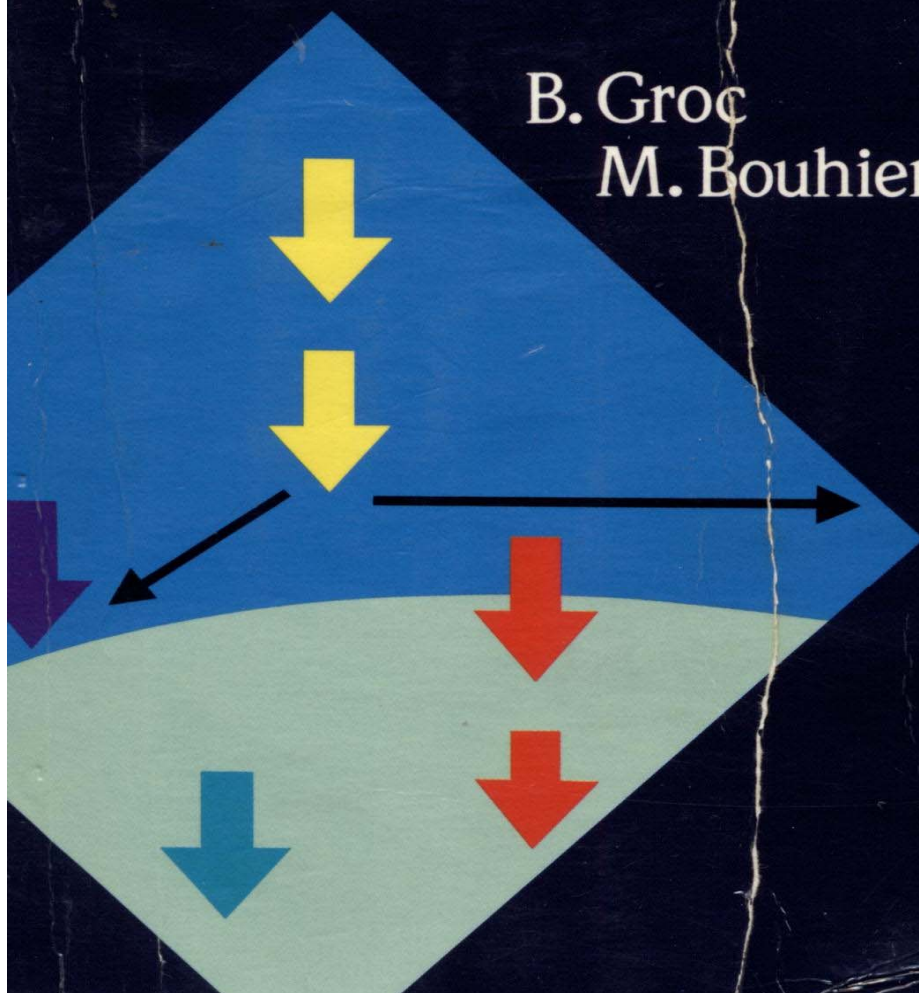


La programmation par syntaxe

Des grammaires à la compilation

B. Groc
M. Bouhier



DUNOD
informatique

2-005-233-1

2-005-233-1



La programmation par syntaxe

Des grammaires à la compilation

BERNARD GROC

Professeur

MARCEL BOUHIER

Maître de conférence

*au département informatique
de l'IUT2 de Grenoble*

DUNOD
informatique

SOMMAIRE

1. Généralités sur les langages	1
1.1. Structure d'un langage	1
1.2. Grammaire et langage engendré par une grammaire	3
1.2.1. <i>Les composants d'une grammaire</i>	4
1.2.2. <i>Dérivation-réduction de chaînes</i>	5
1.2.3. <i>Langage engendré par une expression, par une grammaire</i>	6
1.3. Classification de Chomsky des grammaires et des langages	7
1.4. Expressions régulières	9
1.5. Métalangues usuelles d'écriture des grammaires	11
1.5.1. <i>Notation de Backus (BNF)</i>	11
1.5.2. <i>Diagrammes syntaxiques</i>	12
1.6. Arbre de dérivation d'une phrase	14
1.7. Priorité des opérateurs	15
1.8. Ambiguïtés de langages apportés par la grammaire	16
1.8.1. <i>Ambiguïtés de lecture de la grammaire</i>	16
1.8.2. <i>Ambiguïtés syntaxiques</i>	17
1.9. Exercices	19
2. Analyse descendante	23
2.1. Classification des méthodes d'analyse	24
2.2. Compléments sur les grammaires	25
2.2.1. <i>Règles et grammaires récursives</i>	25
2.2.2. <i>Grammaire propre et factorisée</i>	26
2.2.3. <i>Expressions génitrices d'une grammaire</i>	26
2.3. Grammaire = Schéma de programme	27
2.3.1. <i>Principe de traduction de règles de grammaire</i>	28
2.3.2. <i>Exemple : Traduction d'une grammaire en un programme de synthèse</i>	29

2.4. Analyse descendante LL non déterministe	30
2.4.1. <i>Spécifications des fonctions d'analyse</i>	31
2.4.2. <i>Principes de traduction de la partie droite d'une règle</i>	32
2.4.3. <i>Des exemples d'analyseurs LL</i>	35
2.5. Formes de grammaires mal adaptées à l'analyse LL	37
2.6. Transformations de grammaires	38
2.7. Analyse descendante LL(1)	40
2.7.1. <i>Principes de l'analyse LL(1)</i>	41
2.7.2. <i>Ensembles de symboles associés à une expression génitrice</i>	44
2.7.3. <i>Principes de traduction de la partie droite d'une règle</i>	51
2.7.4. <i>Marche à suivre pour la traduction d'une grammaire en un analyseur LL(1)</i>	52
2.7.5. <i>Exemples d'écriture d'un analyseur LL(1)</i>	53
2.8. Grammaire LL(1)	61
2.8.1. <i>Reconnaître à première vue une grammaire non LL(1)</i>	62
2.8.2. <i>Exemples de vérification des conditions LL(1)</i>	63
2.9. Programmation d'un analyseur LL(1) déterministe	65
2.9.1. <i>Structure générale d'un analyseur LL(1) déterministe</i>	65
2.9.2. <i>Principe de traduction de la partie droite d'une règle</i>	66
2.9.3. <i>Cas de simplifications de la traduction</i>	67
2.9.4. <i>Exemples d'analyseur LL(1) déterministe</i>	68
2.10. Exercices	69

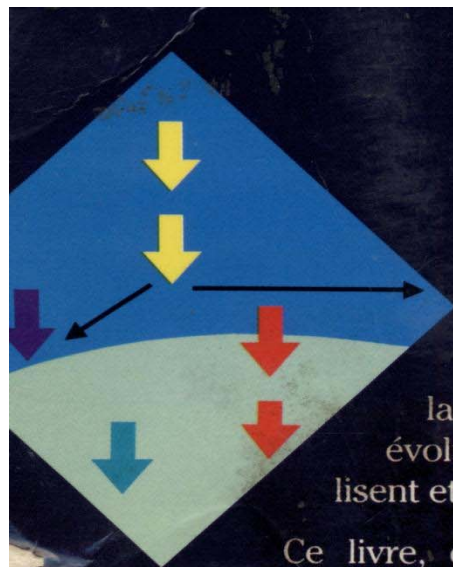
3. Analyse avec lexicographie 71

3.1. Lexicographie, Syntaxe d'un langage et d'une phrase	71
3.2. Structure de l'analyseur	74
3.2.1. <i>Analyse lexicographique d'un symbole</i>	75
3.2.2. <i>Ensembles premiers, suivants et directeurs</i>	82
3.2.3. <i>Analyseur syntaxique</i>	82
3.3. Exercices	84

4. Actions sémantiques 87

4.1. Représentation abstraite d'une phrase	88
4.2. Notion d'action sémantique	89
4.2.1. <i>Exemple : construction de la forme postfixée d'une expression</i>	89
4.2.2. <i>Exemple : construction de l'arbre abstrait d'une expression</i>	91
4.3. Spécification des actions sémantiques dans la grammaire	93
4.3.1. <i>Grammaires avec actions sémantiques</i>	94
4.3.2. <i>Traduction des grammaires avec action sémantique</i>	96
4.4. Exercices	99

5. Grammaires et programmation	101
5.1. Programmation par syntaxe	101
5.2. Recherche associative séquentielle	102
5.2.1. <i>Recherche dans une structure séquentielle non triée</i>	102
5.2.2. <i>Recherche dans une structure séquentielle triée</i>	105
5.3. Parcours de données calculées	106
5.4. Algorithme de réécriture	108
5.5. Tracé d'une courbe polygonale	111
5.6. Exercices	113
6. Introduction à la compilation	115
6.1. Organisation globale de la mémoire	116
6.2. Adressage dans les environnements accessibles	121
6.3. Organisation de l'enregistrement d'activation d'une routine	123
6.4. Allocation de mémoires temporaires	124
6.5. Passages de paramètres	128
6.6. Appel de sous-programme	133
6.7. Mise en œuvre de l'organisation mémoire	135
6.8. Traduction des expressions algébriques	137
6.9. Code objet d'un programme	142
6.10. Evaluation sémantique	152
7. Exemple de compilation d'un langage évolué	155
7.1. Langage source - Classes lexicales	155
7.2. Langage objet	161
7.3. Traitement des erreurs	163
7.4. Descripteurs de mémoires	164
7.5. Descripteurs de routine	165
7.6. Descripteurs de types - Sémantique des opérateurs	166
7.7. Table des symboles	171
7.8. Etat de l'analyseur	178
7.9. Analyse lexicographique	182
7.10. Primitives de génération de code	185
7.11. Analyse syntaxique	197
7.12. Compilateur écrit en Pascal	203
8. Correction des exercices	219
Chapitre 1	219
Chapitre 2	223
Chapitre 3	233
Chapitre 4	237
Chapitre 5	239
Références bibliographiques	243



LA PROGRAMMATION PAR SYNTAXE

Les techniques d'analyse grammaticale et de compilation sont indispensables à l'utilisateur de langages évolués qui doit connaître la façon dont les machines lisent et comprennent les programmes.

Ce livre, d'un abord simple, comprend une partie formelle définissant les notions abordées (grammaires, analyses LL, – ou gauche/droite descendantes –, actions sémantiques...) et une partie pratique qui est l'application de la méthode à quelques problèmes classiques puis à l'écriture d'un compilateur. Celui-ci fonctionne selon les principes d'une des techniques d'analyse les plus élémentaires à mettre en œuvre : l'analyse LL1. Cette technique, dite de programmation par syntaxe, consiste en une traduction quasi automatique de la grammaire définissant un langage en un programme qui reconnaît les phrases de ce langage.

Cet ouvrage, qui s'adresse en priorité aux étudiants en informatique (IUT, maîtrise...), pourra donc aussi être lu avec profit par tous les professionnels soucieux de compléter leurs connaissances.



ISBN 2-04-019837-7