

**Samy Ait-Aoudia**

S'IL FAIT TROP CHAUD  
J'ACTIVERAI  
LA CLIMATISATION

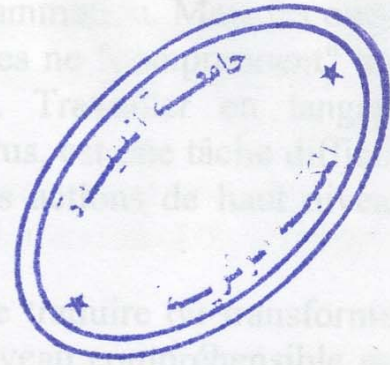
# Compilation

**Cours et Exercices Corrigés**



OFFICE DES PUBLICATIONS UNIVERSITAIRES

**Samy AIT-AOUDIA**



# COMPILATION

**Cours et Exercices Corrigés**

**2<sup>ème</sup> Édition**



**OFFICE DES PUBLICATIONS UNIVERSITAIRES**



# Sommaire

CHAPITRE I. Introduction aux Compilateurs .....	5
I.1. Traduction.....	5
I.2. Structure d'un compilateur .....	6
I.2.2. Analyse lexicale.....	6
I.2.3. Analyse syntaxique .....	7
I.2.4. Analyse sémantique .....	8
I.2.5. Génération de code intermédiaire .....	8
I.2.6. Optimisation .....	9
I.2.7. Génération de code .....	9
I.2.8. La table des symboles .....	10
I.3. Outils d'écriture de compilateurs .....	12
CHAPITRE II. Analyse lexicale.....	17
II.1. Introduction .....	17
II.2. Les expressions régulières .....	18
II.2.1. Chaînes et langages.....	18
II.2.2. Définition d'une expression régulière.....	19
II.3. Automate d'états finis .....	21
II.3.1. Formalisme .....	21
II.3.2. Automate d'états finis déterministe (AFD) .....	21
II.3.3. Automate d'états finis non déterministe (AFN).....	23
II.3.4. Représentation d'un automate.....	23
II.4. Transformation automate non déterministe en automate déterministe .....	24
II.4.1. Algorithme de transformation .....	24
II.4.2. Minimisation du nombre d'états de l'AFD .....	27
II.5. Transformation expression régulière en automate non déterministe .....	30

II.5.1. Construction de Thompson.....	30
II.5.2. Algorithme de Glushkov.....	33
II.6. Transformation d'une expression régulière en automate déterministe .....	38
II.7. Transformation automate déterministe en expression régulière	41
II.8. Analyseur lexical .....	43
II.8.1. Contrôle par AFD vs. Contrôle par programme .....	43
II.8.2. Mots-clés vs. Identificateurs .....	45
II.8.3. Fonctionnement d'un analyseur lexical .....	47
II.9. Utilisation de Lex dans l'analyse lexicale .....	49
II.9.1. Introduction à LEX.....	49
II.9.2. Expressions régulières de LEX .....	51
II.9.3. Structure d'un programme LEX.....	54
II.9.4. Quelques fonctions et variables de LEX.....	58
II.9.5. Exemple de programmes LEX .....	60
II.9.6. Utilisation des états d'analyse : exemple .....	63
II.10. Exercices .....	64
CHAPITRE III. Principes de l'Analyse Syntaxique .....	69
III.1. Introduction .....	69
III.2. Les grammaires non-contextuelles.....	70
III.3. Dérivations et langages .....	70
III.3.1. Dérivations .....	70
III.3.2. Langages.....	72
III.3.3. Arbre syntaxique .....	72
III.3.4. Ambiguïté ou non ambiguïté d'une grammaire .....	73
III.4. Transformations de Grammaires non-contextuelles.....	75
III.4.1. Propriétés particulières d'une grammaire .....	75
III.4.2. Elimination des symboles inutiles.....	75
III.4.3. Elimination des productions unitaires .....	77
III.4.4. Rendre une grammaire $\epsilon$ -libre .....	78



III.4.5. Substitution des non-terminaux .....	79
III.5. Principe des analyses descendantes et ascendantes .....	80
III.6. Backus–Naur Form (BNF).....	81
III.7. Exercices .....	82
<b>CHAPITRE IV. Analyse syntaxique descendante .....</b>	<b>85</b>
IV.1. Introduction.....	85
IV.2. Analyse syntaxique LL.....	85
IV.2.1. Analyse syntaxique LL(1).....	86
IV.2.2. Analyse syntaxique LL(k).....	97
IV.3. Analyse syntaxique par descente récursive .....	99
IV.3.1. Conditions préalables à une descente récursive .....	99
IV.3.2. Ecriture des procédures.....	99
IV.3.3. Exemple d'analyse .....	101
IV.4. Exercices .....	105
<b>CHAPITRE V. Analyses syntaxiques ascendantes .....</b>	<b>109</b>
V.1. Introduction.....	109
V.2. Analyse par précedence d'opérateurs .....	109
V.2.1. Grammaire d'opérateurs .....	109
V.2.2. Relations de précedence d'opérateurs.....	110
V.2.3. Grammaire de précedence d'opérateurs .....	110
V.2.4. Algorithme d'analyse de précedence d'opérateurs .....	112
V.2.5. Exemple d'analyse .....	112
V.2.6. Analyse et Critiques de la méthode.....	113
V.3. Analyse par précedence simple.....	114
V.3.1. Relations de précedence simple.....	114
V.3.2. Grammaire de précedence simple .....	
V.3.3. Algorithme d'analyse de précedence simple.....	115
V.3.4. Exemple d'analyse .....	116
V.3.5. Optimisation de la taille de la table des relations .....	118
V.3.6. Transformations de grammaires simples .....	121

V.3.7. Limites de l'analyse par précedence simple .....	121
V.4. Analyse par précedence faible .....	123
V.4.1. Grammaires de précedence faible .....	123
V.4.2. Analyse par précedence faible.....	123
V.4.3. Exemple d'analyse .....	125
V.4.4. Optimisation .....	127
V.5. Analyse par la méthode LR(k).....	128
V.5.1. Analyse Contextuelle .....	128
V.5.2. Méthode pratique pour la construction d'analyseurs LR ....	140
V.5.3. Utilisation des grammaires ambiguës .....	157
V.5.4. Gestion des erreurs en analyse LR.....	159
V.5.5. Analyse et Classification des grammaires.....	160
V.6. Exercices .....	162
CHAPITRE VI. Traduction Dirigée par la Syntaxe .....	167
VI.1. Introduction.....	167
VI.2. Formes intermédiaires .....	168
VI.2.1. Notation post-fixée .....	169
VI.2.2. Code à trois adresses .....	169
VI.2.3. Arbres abstraits.....	173
VI.2.4. Bytecode .....	174
VI.3. Principe de la traduction dirigée par la syntaxe .....	176
VI.3.1. Analyseur syntaxique en tant que "noyau" de la traduction	176
VI.3.2. Attributs des symboles de la grammaire .....	177
VI.3.3. Analyses n'utilisant que des attributs synthétisés.....	177
VI.3.4. Analyses utilisant des attributs hérités.....	178
VI.3.5. Schémas de traduction .....	179
VI.4. Traduction descendante.....	182
VI.4.1. Emplacement des routines sémantiques .....	182
VI.4.2. Grammaires naturelles récursives gauches.....	183
VI.4.3. Grammaires naturelles non factorisées .....	184



VI.4.4. Conception d'un traducteur descendant.....	185
VI.5. Traduction ascendante .....	195
VI.5.1. Emplacement des routines sémantiques .....	195
VI.5.2. Elimination des actions intérieures .....	195
VI.5.3. Exemples de schémas de traduction .....	197
VI.6. YACC .....	203
VI.6.1. Grammaires YACC.....	203
VI.6.2. Structure d'un programme YACC.....	203
VI.6.3. Variables et commandes de YACC .....	204
VI.6.4. Exemple de programme YACC.....	205
VI.6.5. Fonctionnement de l'analyseur .....	206
VI.7. Exercices .....	212
CHAPITRE VII. Environnements d'exécution .....	217
VII.1. Introduction.....	217
VII.2. Organisation de l'espace mémoire.....	217
VII.2.1. Taille d'un exécutable sur mémoire secondaire.....	217
VII.2.2. Répartition de la mémoire à l'exécution .....	218
VII.3. Procédures et activations .....	219
VII.3.1. Procédures.....	219
VII.3.2. Bloc d'activation .....	220
VII.3.3. Appels de procédure.....	221
VII.4. Allocation de la mémoire .....	222
VII.4.1. Allocation statique .....	222
VII.4.2. Allocation en pile .....	222
VII.4.3. Allocation dans le tas.....	225
VII.5. Noms et adresses mémoires dans le langage C .....	226
VII.5.1. Règles de portée .....	226
VII.5.2. Structure de blocs.....	227
VII.5.3. Déclaration de procédures .....	229
VII.5.4. Classes de stockage en C .....	231

VII.5.5. Allocation dynamique .....	233
VII.6. Exercices .....	234
CHAPITRE VIII. Production de Code.....	237
VIII.1. Introduction.....	237
VIII.2. Machine Cible .....	238
VIII.3. Blocs de base et graphes de flot de contrôle .....	240
VIII.3.1. Blocs de Base .....	240
VIII.3.2. Transformations sur les blocs de Base .....	242
VIII.3.3. Construction du graphe de flot de contrôle .....	244
VIII.4. Un générateur de Code simple.....	245
VIII.4.1. Informations d'utilisation ultérieure .....	245
VIII.4.2. Descripteurs de registres et d'adresses.....	246
VIII.4.3. Algorithme de production de code.....	247
VIII.4.4. Production de code à partir de DAG.....	249
VIII.4.5. Assignation globale de registres .....	261
VIII.5. Exercices .....	265
Correction d'exercices sélectionnés.....	267
Bibliographie .....	287
Annexe A. Classification des Grammaires.....	289
Annexe B. Formes Normales des Grammaires.....	291
Sommaire.....	295