

les bases de la programmation

Jacques ARSAC

DUNOD

informatique



2-005-965-1

مصلحة الإقتناء

les bases de la programmation

par
Jacques ARSAC

Correspondant
de l'Académie des Sciences



DUNOD

informatique

Table des matières

Prologue	1
1. La programmation itérative	9
1.1. Un programme faux	9
1.2. L'instruction d'affectation	10
1.3. Assertions et instructions	11
1.3.1. Affectation	11
1.3.2. Sélection	13
1.3.3. Branchement	14
1.3.4. TANT QUE	15
1.3.5. Boucle	15
1.4. Interprétation d'un programme	16
1.5. Construction récurrente de programmes itératifs	18
1.6. Correction partielle	20
1.7. Renforcement de l'hypothèse de récurrence	23
1.8. Effet du choix de l'hypothèse de récurrence	24
1.9. Le tri en épi (heap sort)	28
2. La programmation récursive	33
2.1. Rappel sur la récursivité	33
2.2. Construction de définitions récursives	34
2.2.1. La factorielle	34
2.2.2. L'exponentielle	35
2.2.3. L'inverse d'une chaîne	36
2.2.4. Edition d'un entier dans une base b	37
2.2.5. Les nombres de Fibonacci	37
2.2.6. Le produit de deux entiers	38
2.3. Calcul d'une fonction définie récursivement	38
2.3.1. Définition monadique	38
2.3.2. Définition dyadique	39
2.4. Nature d'une définition récursive	40
2.5. Propriétés d'une fonction définie récursivement	41
2.5.1. Exemple simple	41
2.5.2. Précision des calculs	41
2.5.3. Complexité	44
2.6. Choix d'une forme récursive	45
2.6.1. Le problème de l'arrêt	45
2.6.2. Complexité	46
2.7. Transformation de définitions récursives	47

2.8.	Du bon usage de la récursivité	49
2.8.1.	La fonction 91	50
2.8.2.	L'appel par nécessité	52
2.9.	La programmation fonctionnelle	53
	Exercices	55
3.	La programmation récurrente	57
3.1.	Algorithmes directs	57
3.2.	Composition des algorithmes	59
3.3.	Relation de précédence	61
3.4.	Transformation en programme	61
3.5.	Evaluation retardée	63
3.6.	Algorithme récurrent	66
3.6.1.	Un exemple	66
3.6.2.	Forme générale	68
3.6.3.	Réurrences profondes	69
3.7.	Exécution d'un algorithme récurrent	70
3.8.	Transformation en programme itératif	72
3.8.1.	Cas général	72
3.8.2.	Exemple	73
3.8.3.	Réduction du nombre des variables	75
3.9.	Emploi de la récurrence pour la création d'un programme	75
3.10.	Donner un sens à un programme	81
	Exercices	84
4.	De la récursivité à la récurrence	87
4.1.	Forme récurrente associée à un schéma récursif	87
4.1.1.	Le schéma considéré	87
4.1.2.	Forme récurrente associée	88
4.1.3.	Programme itératif	89
4.1.4.	Exemple	90
4.2.	La pile	91
4.3.	Programmes sans pile	96
4.4.	Programme avec une seule boucle	98
4.5.	Interprétation du schéma récursif	99
4.6.	Autres simplifications	100
4.6.1.	Associativité	100
4.6.2.	Permutabilité	101
4.7.	Forme dégénérée du schéma récursif	103
4.7.1.	La forme dégénérée	103
4.7.2.	a est constant	104
4.7.3.	h est l'application identique	105
4.8.	Cas d'espèce	106
4.8.1.	Un schéma dyadique	106
4.8.2.	Loi non associative	107
	Exercices	110
5.	De la récursivité à l'itération	111
5.1.	Un exemple de transformation directe	111
5.1.1.	Invariant de boucle	111
5.1.2.	Construction du programme itératif	112
5.2.	Extension	113
5.2.1.	Généralisation	113

5.2.2. Test de fin	113
5.2.3. Progression	113
5.2.4. Initialisation	114
5.3. Exemples	114
5.3.1. Inverse d'une chaîne	114
5.3.2. Edition en base b	116
5.4. La synthèse de programmes	118
5.5. Une récursivité complexe	121
5.6. Les nombres heureux	124
5.7. La suite de Haming	130
Exercices	133
6. Les actions régulières	135
6.1. Présentation intuitive de la notion d'action	135
6.2. Système d'actions régulières	137
6.3. Interprétation d'un système d'actions régulières	139
6.3.1. Interprétation comme étiquette	139
6.3.2. Découpage d'un programme en actions	141
6.3.3. Les actions, procédures sans paramètres	142
6.4. Graphe d'un système d'actions	142
6.5. Substitution	143
6.5.1. Règle de copie	143
6.5.2. Transformation inverse	145
6.6. Transformation de la récursivité terminale en itération	145
6.6.1. Cas particulier	145
6.6.2. Cas général	147
6.7. Structuration d'un système d'actions régulières	150
6.7.1. Cas général	150
6.7.2. Exemple : recherche d'une sous-chaîne dans une chaîne	151
6.8. L'identification	155
Exercices	161
7. Transformations de programmes	163
7.1. Définitions et notations	163
7.2. Absorption simple et expansion simple	165
7.3. Double itération. Absorption de boucle	167
7.4. L'inversion propre	171
7.5. La répétition	174
7.6. Transformations sémantiques locales	175
7.6.1. Affectations	175
7.6.2. Sélections	177
7.7. Elaboration de transformations plus complexes	179
7.7.1. Boucles TANT QUE	179
7.7.2. Une transformation de Suzan Gerhardt	181
7.7.3. Remarque sur l'opérateur ET	185
7.8. Application à un problème d'arrêt	186
7.9. Changement de stratégie	189
Exercices	197
8. La transformation de sous-programmes récursifs en itératifs	199
8.1. Actions généralisées	199
8.1.1. Présentation intuitive	199

8.1.2. Définition	200
8.1.3. Interprétation	201
8.2. Régularisation	203
8.2.1. Cas général	203
8.2.2. Utilisation de prédicats	203
8.2.3. Propriété d'invariance	204
8.2.4. Compteur d'activation	205
8.2.5. Compteur n -aire	205
8.3. Exemple	207
8.4. Schémas d'Irlik	208
8.5. Paramètres formels et variables globales	213
8.5.1. Variables locales	213
8.5.2. Paramètres formels	214
8.6. Un schéma récursif connu	215
Exercices	221
9. La programmation analytique	223
9.1. La programmation analytique	223
9.2. Une vieille connaissance	224
9.2.1. Création du sous-programme récursif	225
9.2.2. Régularisation	226
9.2.3. Première réalisation de la pile	227
9.2.4. Deuxième implémentation de la pile	228
9.2.5. Variantes	231
9.3. Un problème de permutations	232
9.3.1. La procédure récursive	232
9.3.2. Première forme itérative	233
9.3.3. Deuxième forme itérative	237
9.4. Les tours de Hanoï	243
9.4.1. La procédure récursive	243
9.4.2. Première forme	244
9.5. Un résultat général	251
9.5.1. La pile	251
9.5.2. Recalcul des arguments	252
9.6. Les tours de Hanoï, deuxième forme	252
9.7. Remarques sur les deux solutions des tours de Hanoï	256
Exercices	258
Liste des programmes discutés dans le livre	261
Notations	263
Bibliographie	265
Index	269