

Table des matières

1	La programmation fonctionnelle	1
1.1	Généralités sur les langages de programmation	1
1.1.1	Logiciel, code, langage et algorithme	1
1.1.2	Les paradigmes de programmation	2
1.1.3	Le choix d'un langage	6
1.2	Les langages fonctionnels	6
1.2.1	Expressions, fonctions et transparence référentielle . . .	6
1.2.2	Principe de la programmation fonctionnelle	8
1.2.3	Les théories liées à la programmation fonctionnelle . . .	9
1.3	Panorama des langages fonctionnels	10
1.3.1	Les langages de type LISP	11
1.3.2	Les langages de type ML	12
1.3.3	Autres langages fonctionnels	12
1.4	Utilisation des langages fonctionnels	13
1.4.1	Domaines d'application	13
1.4.2	Quelques exemples notables	16
2	Notions de base d'Haskell	19
2.1	Découverte d'Haskell	19
2.1.1	Le langage Haskell	19
2.1.2	Exécuter du code	21
2.2	Syntaxe de base	24
2.2.1	Commentaires de code	24
2.2.2	Expressions	25
2.2.3	Variables	25
2.2.4	Fonctions	28
2.2.5	Parenthèses	30
2.2.6	Mots-clés	31
2.2.7	Indentation	32
2.2.8	Entrées/sorties	32

2.2.9	Arguments de la ligne de commande	35
2.3	Introduction aux types et aux classes	35
2.3.1	Notion de type	35
2.3.2	Types élémentaires	36
2.3.3	Types composés	40
2.3.4	Types de fonction	43
2.3.5	Types polymorphes	44
2.3.6	Notion de classe de types	46
2.3.7	Classes de types standards	48
2.3.8	Classes de types et inférence	52
2.4	Fonctions	53
2.4.1	Fermeture de fonction	53
2.4.2	Fonction lambda	54
2.4.3	Fonction d'ordre supérieur	55
2.4.4	Curryfication	56
2.4.5	Évaluation partielle	56
2.4.6	Composition de fonction	57
2.4.7	Notation « point-free »	58
2.4.8	Syntaxe des opérateurs binaires	59
2.5	Conditions, filtrage par motif, gardes	60
2.5.1	Expressions conditionnelles simples	61
2.5.2	Expressions conditionnelles multiples	62
2.5.3	Filtrage par motif (pattern matching)	62
2.5.4	Gardes	64
2.6	Fonctions récursives	64
2.6.1	Principe de la récursivité	64
2.6.2	Récursivité terminale	65
2.6.3	Récursivité sur des listes	66
2.7	Traitements de listes	67
2.7.1	Mapping	67
2.7.2	Filtrage	68
2.7.3	Réduction	69
2.8	Autres fonctionnalités sur les listes	70
2.8.1	Listes infinies	70
2.8.2	Liste en compréhension	71
2.8.3	Appairage de listes (zip)	73
2.9	Modules	74
2.9.1	Définir des modules	74
2.9.2	Importer des modules	75
2.9.3	Bibliothèques de modules	76

3	Types algébriques et classes de types	79
3.1	Types algébriques	79
3.1.1	Synonymes de types	80
3.1.2	Énumérations	80
3.1.3	Constructeurs de valeurs	81
3.1.4	Constructeurs de types	83
3.1.5	Types récursifs	84
3.1.6	Enregistrements	87
3.1.7	Quelques types algébriques standards	87
3.1.8	Type algébrique et module	91
3.2	Classes de types	92
3.2.1	Instancier une classe de types	93
3.2.2	Définir une nouvelle classe de types	95
3.2.3	Instanciation par défaut et dérivation de classe	96
3.3	Sortes	97
3.3.1	Sorte d'un type	97
3.3.2	Sorte d'une classe	99
3.4	Monades	99
3.4.1	Définition et lois des monades	100
3.4.2	Exemple de la monade <code>Maybe</code>	101
3.4.3	Implémentation et vocabulaire	102
3.4.4	Notation <code>do</code>	103
3.4.5	Quelques monades standards	104
3.4.6	Programmer avec des monades	108
3.4.7	Exemple de la monade <code>State</code>	116
3.5	Foncteurs et foncteurs applicatifs	122
3.5.1	Foncteurs	122
3.5.2	Foncteurs applicatifs	126
3.6	Introduction à l'analyse syntaxique	133
3.6.1	Parsing fonctionnel	134
3.6.2	La bibliothèque <code>Megaparsec</code>	140
3.7	Pour aller plus loin avec Haskell	145
3.7.1	Exemples de code Haskell	145
3.7.2	Monades	146
3.7.3	Autres classes de types	146
3.7.4	Extensions de langage	146
3.7.5	Performances de code	146

4	Exemples d'applications	149
4.1	Mise en place d'un projet en Haskell	149
4.1.1	Description du projet	149
4.1.2	Configurer un projet avec Cabal	151
4.1.3	Gérer les dépendances avec Stack	152
4.1.4	Gérer les dépendances avec Nix	153
4.1.5	Générer une documentation de code	154
4.1.6	Exécuter des tests automatisés	155
4.2	Applications simples	159
4.2.1	Parcourir une matrice en spirale	159
4.2.2	Casser le chiffrement de César	164
4.3	Applications des types et des classes	167
4.3.1	Intelligence artificielle pour le jeu de tic-tac-toe	168
4.3.2	Marche aléatoire discrète	175
4.4	Applications web ou réseaux	179
4.4.1	Un client-serveur d'écho, avec des sockets réseaux	180
4.4.2	Un blog basique, avec le framework Scotty	184
4.5	Compilateurs	190
4.5.1	Une calculatrice basique en notation préfixée	190
4.5.2	Un convertisseur Markdown vers HTML	193
5	Le style fonctionnel dans d'autres langages	199
5.1	Le style fonctionnel en C++	200
5.1.1	Pattern matching et déconstruction	200
5.1.2	Immuabilité	201
5.1.3	Fonctions et lambdas	202
5.1.4	Fonctions d'ordre supérieur	203
5.1.5	Mapping, filtrage et réduction	203
5.1.6	Curryfication et évaluation partielle	205
5.1.7	Récursivité	205
5.1.8	Fonctions pures et transparence référentielle	206
5.1.9	Fermetures	207
5.1.10	Évaluation paresseuse et mémoïsation	208
5.2	Le style fonctionnel en JavaScript	209
5.2.1	Pattern matching et déconstruction	210
5.2.2	Immuabilité	211
5.2.3	Fonctions et lambdas	211
5.2.4	Fonctions d'ordre supérieur	212
5.2.5	Mapping, filtrage et réduction	213
5.2.6	Curryfication et évaluation partielle	214

5.2.7	Récurtivité	215
5.2.8	Fonctions pures et transparence référentielle	216
5.2.9	Fermetures	216
5.2.10	Monades	217
5.3	Un exemple d'application	219
5.3.1	Implémentation en Haskell	220
5.3.2	Implémentation en C++	223
5.3.3	Implémentation en JavaScript	226
5.3.4	Implémentation en Python	228
5.3.5	Implémentation en Julia	231
5.3.6	Implémentation en Rust	233
	Conclusion	239
	Index	243