

Table des matières

Préface	1
Gilles DOWEK et Catherine DUBOIS	
Avant-propos	3
Chapitre 1. Du matériel au logiciel	5
1.1. L'ordinateur vision bas niveau	5
1.1.1. Traitement de l'information	5
1.1.2. Mémoires	6
1.1.3. Microprocesseur	8
1.1.4. Périphériques	11
1.2. L'ordinateur vision haut niveau	13
1.2.1. Modélisation du calcul	13
1.2.2. Langages de haut niveau	14
1.2.3. Du code source à l'exécutable	15
Chapitre 2. Introduction à la sémantique des langages	19
2.1. Environnement, mémoire, état	19
2.1.1. Environnement d'évaluation	19
2.1.2. Mémoire	22
2.1.3. État	24
2.2. Évaluation d'expressions	24
2.2.1. Syntaxe	25
2.2.2. Valeurs	26
2.2.3. Sémantique d'évaluation	28
2.3. Définition, affectation	30
2.3.1. Définition d'un identificateur	30
2.3.2. Affectation	33
2.4. Exercices	35

Chapitre 3. Sémantique des traits fonctionnels	39
3.1. Aspects syntaxiques	39
3.1.1. Syntaxe du noyau d'un langage fonctionnel	39
3.1.2. Arbre syntaxique	40
3.1.3. Raisonnement par induction sur les expressions	43
3.1.4. Déclaration de variables, variables muettes, variables libres	43
3.2. Sémantique d'exécution : fonction d'évaluation	46
3.2.1. Erreurs lors de l'évaluation	46
3.2.2. Valeurs	47
3.2.3. Interprétation des opérateurs	49
3.2.4. Fermeture	50
3.2.5. Évaluation des expressions	51
3.3. Sémantique d'exécution : sémantique opérationnelle	58
3.3.1. Expressions simples	59
3.3.2. Appel par valeur	60
3.3.3. Fonctions récursives, fonctions mutuellement récursives	64
3.3.4. Appel par nom	65
3.3.5. Appel par valeur <i>versus</i> appel par nom	67
3.4. Fonction d'évaluation <i>versus</i> relation d'évaluation	68
3.4.1. Statut de la fonction d'évaluation	68
3.4.2. Induction sur les arbres d'évaluation	69
3.5. Propriétés sémantiques	73
3.5.1. Expressions équivalentes	74
3.5.2. Environnements équivalents	75
3.6. Exercices	76
Chapitre 4. Sémantique des traits impératifs	81
4.1. Syntaxe du noyau d'un langage impératif	81
4.2. Évaluation des expressions	85
4.3. Évaluation des définitions	90
4.4. Sémantiques opérationnelles	93
4.4.1. Sémantique à grands pas	93
4.4.2. Sémantique à petits pas	97
4.4.3. Expressivité des sémantiques opérationnelles	99
4.5. Propriétés sémantiques	100
4.5.1. Programmes équivalents	100
4.5.2. Terminaison de programmes	102
4.5.3. Déterminisme de l'exécution des programmes	104
4.5.4. Grands pas <i>versus</i> petits pas	108
4.6. Procédures	113
4.6.1. Blocs	113
4.6.2. Fonctions en impératif : les procédures	116

4.7. Autres approches	122
4.7.1. Sémantique dénotationnelle	122
4.7.2. Sémantique axiomatique, logique de Hoare	133
4.8. Exercices	138
Chapitre 5. Typage	141
5.1. Quand et comment vérifier les types ?	143
5.1.1. À quel moment vérifier les types ?	143
5.1.2. Comment vérifier les types ?	144
5.2. Typier informellement un programme Exp_2	145
5.2.1. Un premier exemple	145
5.2.2. Typier une expression conditionnelle	146
5.2.3. Typier sans contraintes de type	146
5.2.4. Polymorphisme	147
5.3. Règles de typage de Exp_2	148
5.3.1. Types, schémas de type, environnements de typage	148
5.3.2. Généralisation, substitution, instanciation	150
5.3.3. Règles et arbres de typage	155
5.4. Algorithme d'inférence de type de Exp_2	158
5.4.1. Type principal	159
5.4.2. Ensembles de contraintes et unification	159
5.4.3. Algorithme d'inférence de type	163
5.5. Propriétés	172
5.5.1. Propriété du typage	172
5.5.2. Propriétés de l'algorithme d'inférence	172
5.6. Extension du typage aux constructions impératives	173
5.6.1. Algèbre de types	173
5.6.2. Règles de typage	174
5.6.3. Typage des définitions polymorphes	176
5.7. Sous-typage et surcharge	177
5.7.1. Sous-typage	178
5.7.2. Surcharge	180
Chapitre 6. Types de données	183
6.1. Types de base	183
6.1.1. Booléens	183
6.1.2. Entiers	185
6.1.3. Caractères	190
6.1.4. Flottants	191
6.2. Tableaux	195
6.3. Chaînes de caractères	198

6.4. Définitions de types	199
6.4.1. Abréviations de type	199
6.4.2. Enregistrements	200
6.4.3. Types énumérés	205
6.4.4. Types somme	206
6.5. Conditionnelle généralisée	209
6.5.1. <code>switch/case</code> à la C	210
6.5.2. Filtrage	212
6.6. L'égalité	221
6.6.1. Égalité physique	221
6.6.2. Égalité structurelle	223
6.6.3. Égalité entre fonctions	225
Chapitre 7. Pointeurs et gestion de la mémoire	229
7.1. Adresses et pointeurs	229
7.2. Boutisme, <i>endianness</i>	231
7.3. Pointeurs et tableaux	231
7.4. Passage de paramètre par adresse	232
7.5. Références	235
7.5.1. Références en C++	235
7.5.2. Références en Java	240
7.6. Gestion de la mémoire	240
7.6.1. Allocation de mémoire	240
7.6.2. Libération de la mémoire	244
7.6.3. Gestion automatique de la mémoire	246
Chapitre 8. Exceptions	251
8.1. Notifier et propager des erreurs	251
8.1.1. Variable globale	253
8.1.2. Définition d'enregistrement	253
8.1.3. Passage par adresse	254
8.1.4. Introduction des exceptions	254
8.2. Une formalisation simple : les exceptions à la ML	256
8.2.1. Syntaxe abstraite	256
8.2.2. Valeurs	256
8.2.3. Algèbre de types	257
8.2.4. Sémantique opérationnelle	257
8.2.5. Typage	258
8.3. Exceptions dans d'autres langages	259
8.3.1. Exceptions en OCaml	259
8.3.2. Exceptions en Python	260
8.3.3. Exceptions en Java	261
8.3.4. Exceptions en C++	262

Conclusion	265
Annexe. Solution des exercices	267
Liste des notations	295
Liste des implantations des sémantiques	297
Bibliographie	301
Index	303
Sommaire de <i>Concepts et sémantique des langages de programmation 2</i>	307