

# Table des matières

<i>Préface</i> .....	1
<i>Motivations</i> .....	3
Est-ce que le fragment de programme A réalise la tâche T ? .....	3

## Partie 1

### Modéliser pour vérifier et développer des programmes

<i>Leçon 1. Test et vérification de programmes</i> .....	9
Pré requis .....	9
Objectifs .....	9
1. Présentation de la démarche de modélisation pour la vérification et le test ...	10
1.1. Démarche : modéliser pour vérifier ou tester .....	10
1.2. Vérification de la cohérence entre un modèle descriptif .....	13
et une implémentation .....	13
2. Notion de test .....	15
3. Quelques éléments de stratégie de test .....	16
4. Comment établir un test de programme ? .....	17
5. Modéliser pour tester des programmes .....	19
6. Notion de vérification, différence entre vérification et test, limites des méthodes de test .....	20
7. Démarche de vérification .....	21
8. Résumé .....	21
9. Exercice .....	22
<i>Leçon 2. Vérification de programmes par exécution symbolique</i> .....	25
Objectifs .....	25
1. Notion de valeur symbolique de variables .....	26
2. Notion d'exécution avec des valeurs symboliques .....	27
3. Exemple d'exécution symbolique .....	28
4. Résumé .....	31
5. Exercices .....	31
<i>Leçon 3. Un langage de programmation générique</i> .....	35
Objectifs .....	35
1. Syntaxe du langage de programmation .....	35
2. Restrictions du langage de programmation .....	36
3. Sémantique du langage de programmation .....	38

4. Exemples de programmes	38
4.1. Factorielle $n$	38
4.2. Recherche dichotomique	39
4.3. Tri bulle	40
5. Résumé	41
6. Exercices	42
<i>Leçon 4. La logique des prédicats du premier ordre – le langage de modélisation du premier ordre</i>	43
Pré requis	43
Objectifs	43
1. Syntaxe de la logique des prédicats du premier ordre	44
2. Sémantique et propriétés de la logique des prédicats du premier ordre	46
2.1. Interprétation des prédicats	46
2.2. Validité d'un prédicat	47
3. Exemples de prédicats	49
4. Réécriture de prédicats quantifiés	49
5. Stratégies de preuve de validité de prédicats	50
6. Résumé	51
7. Exercice	52
8. Formules Satisfiables, formules Valides <i>versus</i> Tautologies – commentaires	52
9. Notations d'équivalences – commentaires	53
<i>Leçon 5. La logique de Hoare – le système de vérification</i>	55
Pré requis	55
Objectifs	55
1. Rappel de la notion de système formel	56
2. Rappel de la notion de preuve et de théorème	57
3. Comment présenter les preuves ?	57
4. Logique de Hoare	59
5. Propriétés de la logique de Hoare	61
6. Interprétation des règles de la logique de Hoare	62
7. Résumé	70
8. Exercices	70
<i>Leçon 6. Quelques éléments de stratégie de vérification de programmes</i>	73
Pré requis	73
Objectifs	73
1. Vérifier c'est prouver.	73
2. Stratégie de preuve d'une séquence de deux instructions	74
3. Stratégie de preuve d'une conditionnelle	76
4. Stratégie de preuve d'une itération	77
5. Stratégie de preuve d'un programme	78
6. Comment trouver l'ordre des preuves des sous formules ?	80
7. Comment trouver les invariants d'itération ?	82
8. Correction partielle et totale	83
9. Bilan	84
10. Exercice	86

*Leçon 7. Exemple de découverte d'erreurs à la vérification* . . . . . 87

1. Motivations . . . . .	87
2. Exemple 1 : somme de deux polynômes . . . . .	88
2.1. Énoncé du problème et spécification . . . . .	88
2.2. Programme solution du problème . . . . .	89
2.3. Preuve du programme . . . . .	89
2.4. Correction du programme . . . . .	91
3. Exemple 2 : racine carrée par dichotomie . . . . .	92
3.1. Énoncé du problème et spécification . . . . .	92
3.2. Programme solution . . . . .	92
3.3. Preuve du programme . . . . .	92
3.4. Correction de la spécification . . . . .	94
3.5. Correction du programme . . . . .	94
4. Résumé et conclusion . . . . .	95

*Leçon 8. Étude de cas – modélisation et vérification d'un programme de calcul de la racine carrée entière par division* . . . . . 97

1. Énoncé du problème . . . . .	97
2. Spécification . . . . .	97
3. Solution . . . . .	98
3.1. Documents de présentation de la solution . . . . .	98
3.2. Quelques éléments historiques . . . . .	99
3.3. Présentation de la solution algorithmique . . . . .	100
4. Programme . . . . .	101
5. Preuve de solution . . . . .	103
5.1. Preuve de l'invariant [28] . . . . .	104
5.2. Preuve que $y_i < 2r_i + 1$ . . . . .	105
6. Preuve du programme . . . . .	106
6.1. Preuve que l'invariant est vrai avant l'itération . . . . .	106
6.2. Preuve que l'itération conserve l'invariant . . . . .	107
6.3. Preuve que la séquence des deux phases est correcte . . . . .	109
6.4. Preuve de la post condition $r^2 + y = n$ . . . . .	109
6.5. Preuve que $y < 2r + 1$ est invariante . . . . .	110

*Leçon 9. Développer des programmes corrects par construction à partir de modèles* . . . . . 113

1. Construire en prouvant . . . . .	113
2. Exemple du produit de deux nombres . . . . .	115
2.1. Produit par addition . . . . .	115
2.2. Produit par décalage . . . . .	115
3. Exemple du calcul de la racine carrée entière par défaut . . . . .	118
3.1. Calcul de racine carrée par incrémentation . . . . .	120
3.2. Calcul de racine carrée par pas de longueur décroissante . . . . .	120
4. Exemple du drapeau tricolore . . . . .	122
4.1. Spécification du problème . . . . .	122
4.2. Construction d'un programme . . . . .	122
5. Résumé, conclusion et perspectives . . . . .	126
6. Exercice . . . . .	127

# Automatisation de la vérification et de la génération de tests à partir de modèles

<i>Leçon 10. Modélisation de programmes en B</i> .....	131
1. Introduction .....	131
2. Langage B .....	131
2.1. Exemple – Système d'alimentation .....	132
2.2. Typage en B .....	134
2.3. Prédicats et expressions en B .....	136
2.4. Actions en B .....	137
3. Machine Abstraite .....	138
3.1. Modèle de données .....	139
3.2. Spécification descriptive .....	140
3.3. Spécification opérationnelle .....	140
4. Vérification de cohérence .....	145
5. Bilan, conclusion et perspectives .....	146
6. Exercice .....	148
 <i>Leçon 11. Modéliser en B pour engendrer des tests boîte noire</i> .....	 149
1. Problématique et motivations .....	150
2. Exemple de modélisation pour engendrer des tests .....	153
2.1. Modélisation boîte noire de RDicho .....	153
2.2. Modélisation de RDicho tenant compte de la technologie de génération de tests .....	155
2.3. Modélisation de RDicho pour observer la réduction de l'intervalle de recherche .....	155
2.4. Cibles de test et pilotage .....	157
2.5. Modélisation de RDicho pour piloter finement la génération de tests ..	161
2.6. Résultats des expérimentations avec LTG .....	162
3. Méthode de spécification pour la génération automatique de tests .....	165
4. Traçabilité des tests .....	167
5. Exemple du Qui-Donc .....	167
5.1. Spécification informelle .....	168
5.2. Modélisation du Qui-Donc .....	168
5.3. Résultats de la génération de tests .....	175
6. Résumé, conclusion .....	176
7. Exercices .....	177
 <i>Leçon 12. Modéliser des programmes en B pour les vérifier</i> .....	 181
1. Introduction .....	181
2. Exemple introductif .....	183
3. Preuve d'un programme itératif en logique de Hoare .....	185
4. Machine B équivalente et obligations de preuve .....	185
5. Justification des obligations de preuve .....	187
5.1. Justification de $f_3$ .....	187
5.2. Justification de $f_1$ et $f_2$ .....	187

6. Preuve d'un programme avec 2 itérations imbriquées - Application au tri bulle	188
7. Preuve par raffinement du tri bulle	190
7.1. Spécification abstraite du tri	190
7.2. Premier raffinement du tri	192
7.3. Second raffinement du tri	193
7.4. Résultats expérimentaux de preuve	193
8. Généralisation de la modélisation de programmes	193
9. Conclusion	196
10. Exercices	197
11. Annexe 1 : Preuve de l'extension du Calcul des Substitutions	204

*Solutions des exercices* ..... 207

Solution de l'exercice 1	<i>(Test et vérification de programmes)</i>	207
Solution de l'exercice 2	<i>(Raisonnement de cohérence programme annotation ; Tri par sélection)</i>	209
Solution de l'exercice 3	<i>(Comprendre un programme - Calcul de R)</i>	210
Solution de l'exercice 4	<i>(Concevoir un programme - Quick Sort)</i>	210
Solution de l'exercice 5	<i>(Arrêt de programme)</i>	211
Solution de l'exercice 6	<i>(Et bit à bit)</i>	212
Solution de l'exercice 7	<i>(Pgcd de a et b)</i>	214
Solution de l'exercice 8	<i>(Tri bulle)</i>	214
Solution de l'exercice 9	<i>(Variables libres et liées)</i>	215
Solution de l'exercice 10	<i>(Et bit à bit et pgcd - spécification)</i>	215
Solution de l'exercice 11	<i>(Preuve avec les Axiomes de l'affectation)</i>	216
Solution de l'exercice 12	<i>(Preuve par application de la Règle de la séquence)</i>	216
Solution de l'exercice 13	<i>(Preuve par application de la règle de l'itération)</i>	218
Solution de l'exercice 14	<i>(Annotation de programme)</i>	218
Solution de l'exercice 15	<i>(Trouver et présenter une preuve)</i>	218
Solution de l'exercice 16	<i>(Preuve de la procédure placer pour le tri rapide)</i>	219
Solution de l'exercice 17	<i>(Preuves en logique de Hoare)</i>	221
Solution de l'exercice 18	<i>(Preuves en logique de Hoare)</i>	222
Solution de l'exercice 19	<i>(Racine carrée entière par décrémentement)</i>	223
Solution de l'exercice 20	<i>(Modélisation B d'un robot de transport de pièces avec 2 évacuateurs et 3 types de pièces)</i>	224
Solution de l'exercice 21	<i>(Modélisation B de placer)</i>	225
Solution de l'exercice 22	<i>(Modélisation d'un système d'alimentation)</i>	231
Solution de l'exercice 23	<i>(Modéliser pour tester un programme de tri)</i>	235
Solution de l'exercice 24	<i>(Modéliser pour tester une spécification de la fusion de 2 séquences triées)</i>	237
Solution de l'exercice 25	<i>(Preuve avec B4free)</i>	240
Solution de l'exercice 26	<i>(Modéliser des programmes de calcul de la racine carrée par défaut pour les prouver)</i>	241
Solution de l'exercice 27	<i>(Modéliser des programmes réalisant la procédure placer du tri rapide pour les prouver)</i>	242
Solution de l'exercice 28	<i>(Modéliser un programme de calcul de la racine par pas pour le prouver)</i>	243
Solution de l'exercice 29	<i>(Modéliser une autre version de placer pour la prouver)</i>	244

Solution de l'exercice 30 ( <i>Modélisation par raffinement de racine carrée pour faire sa preuve</i> ) . . . . .	246
Solution de l'exercice 31 ( <i>Modéliser un programme utilisé par le tri bulle pour le prouver</i> ) . . . . .	247
Solution de l'exercice 32 ( <i>Modéliser pour vérifier un programme de fusion de deux séquences triées par ordre croissant</i> ) . . . . .	248
Solution de l'exercice 33 ( <i>Preuve d'affectations multiples</i> ) . . . . .	250
<i>Table des figures</i> . . . . .	253
<i>Table des définitions</i> . . . . .	256
<i>Table des exemples</i> . . . . .	257
<i>Table des idées clés</i> . . . . .	257
<i>Table des exercices</i> . . . . .	258
<i>Table des notations</i> . . . . .	259
<i>Glossaire</i> . . . . .	259
<i>Références bibliographiques</i> . . . . .	260
<i>Index</i> . . . . .	262