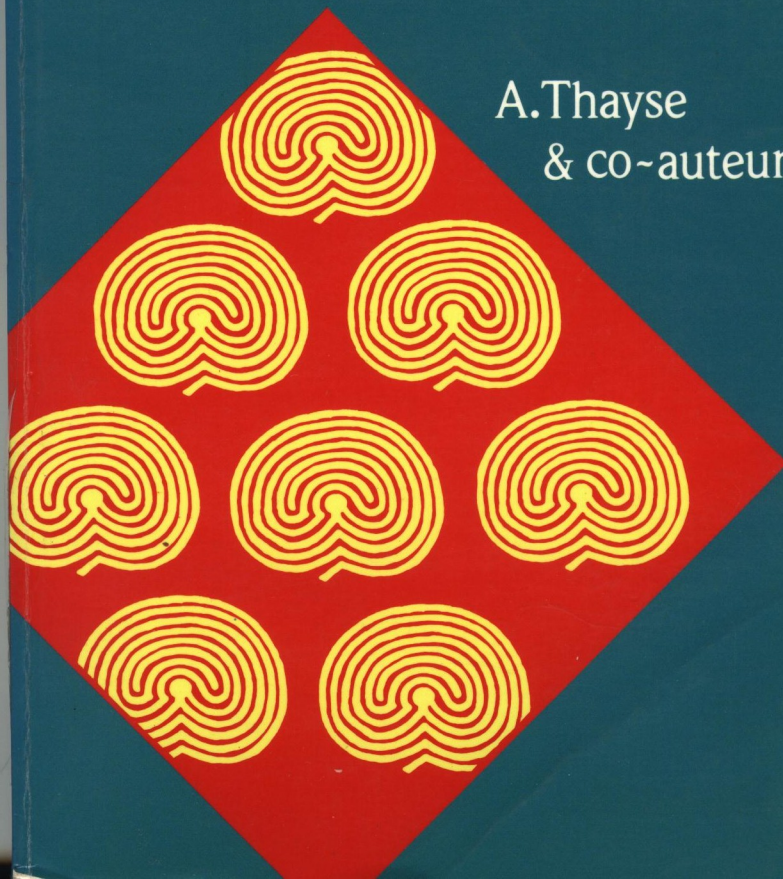


# Approche logique de l'intelligence artificielle

4. De l'apprentissage artificiel  
aux frontières de l'IA

A.Thayse  
& co-auteurs



DUNOD  
informatique

# Table des matières

<b>1</b>	<b>Apprentissage artificiel</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Architecture générale d'un système d'apprentissage . . . . .	6
1.3	Règles d'inférence pour l'apprentissage symbolique . . . . .	7
1.3.1	Déduction . . . . .	8
1.3.2	Induction . . . . .	9
1.3.3	Analogie . . . . .	16
1.3.4	Abduction . . . . .	18
1.4	L'apprentissage par instruction . . . . .	19
1.4.1	Acquisition d'une structure conceptuelle initiale . . . . .	20
1.4.2	Transformation de représentation . . . . .	22
1.4.3	Affinage de la structure conceptuelle . . . . .	23
1.5	L'apprentissage empirique de concept par exemples . . . . .	27
1.5.1	Principe général . . . . .	28
1.5.2	Approche guidée par les exemples . . . . .	31
1.5.3	Approche guidée par des concepts approximatifs . . . . .	43
1.5.4	Conclusion . . . . .	47
1.6	Le regroupement conceptuel . . . . .	48
1.6.1	Principe général . . . . .	49
1.6.2	Approche non incrémentale . . . . .	51
1.6.3	Approche incrémentale . . . . .	56
1.6.4	Conclusion . . . . .	62
1.7	L'apprentissage analytique par explication . . . . .	63
1.7.1	Principe général . . . . .	66
1.7.2	Généralisation par explication . . . . .	68
1.7.3	Apprentissage par analogie . . . . .	78
1.7.4	Conclusion . . . . .	93

1.8	Autres stratégies d'apprentissage . . . . .	96
1.9	Conclusion . . . . .	100
<b>2</b>	<b>La définition d'un langage formel pour la spécification conceptuelle des systèmes informatiques</b>	<b>115</b>
2.1	Introduction . . . . .	115
2.2	Spécification conceptuelle de systèmes informatiques . . .	117
2.2.1	La conception de système . . . . .	117
2.2.2	Langages de spécification conceptuelle . . . . .	121
2.2.3	Spécification conceptuelle et approche logique . . .	124
2.3	Le langage de spécification conceptuelle ERAE . . . . .	125
2.3.1	Introduction . . . . .	125
2.3.2	Logique temporelle . . . . .	126
2.3.3	Logique temporelle typée . . . . .	135
2.3.4	Logique partielle . . . . .	143
2.3.5	Catégories prédéfinies . . . . .	148
2.3.6	Contrôle de variabilité . . . . .	153
2.3.7	Syntaxe graphique . . . . .	156
2.3.8	Intersection de types . . . . .	158
2.3.9	Surcharge de foncteurs . . . . .	167
2.3.10	Héritage de foncteurs . . . . .	170
2.3.11	Métrique temporelle . . . . .	172
2.3.12	Mécanismes de structuration . . . . .	178
2.3.13	Autres aspects d'ERAÉ . . . . .	188
2.4	Exemple . . . . .	189
2.4.1	Organisation des assertions . . . . .	189
2.4.2	Assertions . . . . .	191
2.5	Comparaison . . . . .	193
2.5.1	Expression de contraintes temporelles . . . . .	194
2.5.2	Utilisation d'une logique typée . . . . .	194
2.5.3	Fonctions partiellement définies . . . . .	195
2.5.4	Syntaxe graphique . . . . .	195
2.5.5	Mécanismes de structuration . . . . .	195

2.6	Conclusion . . . . .	196
<b>3</b>	<b>Les extensions de la sémantique de Montague</b>	<b>205</b>
3.1	Introduction . . . . .	205
3.2	Formalisation logique de la langue naturelle . . . . .	208
3.2.1	Les grammaires catégorielles . . . . .	208
3.2.2	La logique intensionnelle . . . . .	215
3.2.3	La sémantique de Montague . . . . .	225
3.2.4	La sémantique booléenne de Keenan et Faltz . . . . .	229
3.3	Sémantique booléenne d'une langue intensionnelle étendue	232
3.3.1	Structure de l'ensemble des dénnotations possibles pour un type booléen . . . . .	232
3.3.2	Enrichissement de la logique de Montague . . . . .	237
3.4	Logique intensionnelle étendue et formalisation de la langue	246
3.4.1	Connecteurs logiques . . . . .	246
3.4.2	Opérateurs modaux . . . . .	252
3.5	Grammaire de Montague flexible . . . . .	257
3.5.1	Introduction . . . . .	257
3.5.2	La grammaire flexible . . . . .	259
3.5.3	Exemples . . . . .	266
3.6	La logique intensionnelle dynamique . . . . .	270
3.6.1	Introduction . . . . .	270
3.6.2	Une grammaire dynamique élémentaire . . . . .	272
3.6.3	Connecteurs et quantificateurs dynamiques . . . . .	276
<b>4</b>	<b>Vérification de preuve dans les langages typés</b>	<b>281</b>
4.1	Introduction . . . . .	281
4.2	Langages typés d'ordre supérieur . . . . .	283
4.3	Les propositions vues comme des types . . . . .	287
4.4	Théorie constructive de types . . . . .	290
4.5	Textes corrects dans le langage $\Lambda\Delta\Pi$ . . . . .	294
4.6	Vérification de preuve . . . . .	303
4.7	Tactiques . . . . .	311
4.8	Conclusion . . . . .	324

<b>5</b>	<b>Une approche algébrique de la méta-programmation en Prolog</b>	<b>327</b>
5.1	Introduction . . . . .	327
5.2	Les algèbres de mots pour la spécification de la traduction de programmes . . . . .	330
5.2.1	Algèbres typées . . . . .	332
5.3	Approche algébrique de la méta-programmation en Prolog	333
5.4	Unification non standard . . . . .	336
5.5	Comptage des inférences logiques . . . . .	340
5.6	La composition de "flavours" . . . . .	346
5.7	Forme normale . . . . .	349
5.8	Comparaison d'efficacité . . . . .	352
5.9	Remarques finales . . . . .	354