

0104C 15033

b13455606 6EPA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
DEPARTAMENT DE LLENGUATGES I SISTEMES INFORMÀTICS
PROGRAMA DE DOCTORAT EN INTEL·LIGÈNCIA ARTIFICIAL

TESI DOCTORAL



APRENENTATGE I GENERACIÓ DE PLANS PER A SISTEMES EXPERTS

Memòria presentada per M. Beatriz López Ibáñez per aconseguir el títol de Doctor en Informàtica. El treball contingut en aquesta memòria ha estat realitzat a l'Institut d'Investigació en Intel·ligència Artificial (Centre d'Estudis Avançats de Blanes-CSIC)

Director: Enric Plaza i Cervera

Març 1993

MAGATZEM, 16

15.03.93

TEC (JP)

UNIVERSITAT ROVIRA I VIRGILI
BIBLIOTECA



1700072458

NREG. 857.346

Resum

Per esdevenir expert, un principiant aprèn a partir de la seva pròpia experiència en la resolució de problemes. L'aprenentatge per experiència ha estat utilitzat en intel·ligència artificial per aprendre conceptes i heurístiques, però s'han realitzat pocs esforços per a l'adquisició dels coneixements estratègics que han adquirit els experts per la pràctica i que resulta necessari per planificar les tasques que els sistemes experts han de dur a terme. Els plans fan eficient la resolució de problemes, controlant els recursos que es necessiten i adaptant el procés de resolució de problemes als canvis que es produeixen en l'entorn. En els dominis mèdics en particular, on la informació que coneix un agent és incerta i incompleta, i la informació va canviant amb el temps, la planificació reactiva permet a un sistema reaccionar ràpidament als canvis de l'entorn. En la recerca actual en sistemes basats en coneixements (SBC) les arquitectures metanivell fan explícita la distinció entre un nivell de planificació i un nivell de domini. Ambdós nivells col·laboren en la resolució d'un problema: el nivell de planificació raona sobre l'estat del procés de resolució de problemes i controla les inferències que s'han de realitzar en el nivell del domini. Tanmateix, adquirir coneixements estratègics pel nivell de planificació és complicat i consumeix temps de l'expert i de l'enginyer de coneixements.

La nostra recerca s'ha focalitzat en l'adquisició de coneixements estratègics i la planificació reactiva en arquitectures metanivell. En primer lloc, hem desenvolupat un planificador basat en casos: BOLERO. BOLERO aprèn coneixements estratègics bé a partir de casos que un mestre proporciona, bé a partir de la seva pròpia experiència. Després d'aprendre des de zero, BOLERO és capaç de construir plans. La qualitat dels plans generats per BOLERO depèn de la completesa de la informació disponible per a un problema donat. Quan més rica és la informació disponible, més precís serà el pla generat.

En segon lloc, hem integrat BOLERO en una arquitectura metanivell: BOLERO-SBC. BOLERO juga el paper del nivell de planificació, i construeix plans d'acord amb els coneixements estratègics que aprèn. Un SBC, SBC^B , consitueix el nivell de domini de l'arquitectura. SBC^B té el seu propi coneixement representat per fets, objectius i regles. En l'arquitectura metanivell, ambdós sistemes col·laboren en la resolució de problemes: els plans construïts per BOLERO organitzen els objectius de SBC^B ; SBC^B assoleix un a un els objectius proporcionats en el pla de BOLERO a partir de les seves regles i fets. En BOLERO-SBC, doncs, la generació i l'execució de plans s'alternen: cada cop que SBC^B coneix nova informació com a conseqüència de l'execució d'un pla, la informació sobre un problema s'enriqueix i BOLERO pot generar un pla nou. Així, el sistema integrat BOLERO-SBC mostra un comportament reactiu.

BOLERO ha estat aplicat a l'aprenentatge del procediment de com realitzar una diagnosi d'una pneumònia. L'avaluació experimental de BOLERO mostra que la planificació basada en casos és un bon mètode per a l'adquisició de plans en els dominis mèdics, i que les arquitectures metanivell són adequades per aconseguir una planificació reactiva. Tots els resultats obtinguts són satisfactoris i encoratgen l'ús de mètodes basats en casos per fer front al coll d'ampolla que representa l'adquisició de coneixements estratègics per al desenvolupament de SBC.

Paraules clau: adquisició de coneixements, aprenentatge, raonament basat en casos, planificació, planificació reactiva, arquitectures metanivell, arquitectures reflexives.

Abstract

In order to become an expert, a novice learns from its own experience in solving problems. Learning from experience has been used in Artificial Intelligence to learn concepts and heuristics, but few efforts have been done to the acquisition of the strategic knowledge that human experts have acquired through practice and that is required to plan the tasks that expert systems carry out. Plans make problem solving efficient, controlling the resources needed and adapting the problem solving process to changes in the environment. Particularly in medical domains, where the information known by an agent is uncertain and incomplete and the information is changing over time, reactive planning allows a system to answer quickly to changes in the world. In the current research on knowledge bases system (KBS) the meta-level architecture approach makes explicit the distinction of a planning level and a domain level. Both levels collaborate in the solution of a problem: the planning level reasons about the state of the problem solving process and controls the inferences to be made in the domain level. To acquire the strategic knowledge for the planning level, however, is complicated and time consuming for the expert and the knowledge engineer.

Our research has been focused on the acquisition of the strategic knowledge and reactive planning in meta-level architectures. Firstly, we have developed a case-based planner, BOLERO. BOLERO learns strategic knowledge either from cases that a teacher provides or from its own experience. After learning from scratch BOLERO is able to build plans. The quality of plans generated by BOLERO depends on the completeness of the information available for a given problem. The richer the information BOLERO has about a problem, the more precise the plan generated will be.

Secondly, we have integrated BOLERO into a meta-level architecture, BOLERO-SBC. BOLERO plays the role of the planning level, and builds plans according to its learned strategic knowledge. A KBS, SBC^B , constitutes the domain level of the architecture. SBC^B has its domain knowledge represented by facts, goals and rules. In the meta-level architecture, both systems collaborate to problem solving: the plans built by BOLERO organize the goals that the KBS must follow; SBC^B reaches one by one all the goals provided in a BOLERO's plan from its rules and facts. Thus, in BOLERO-SBC, generation and execution of plans is interleaved: each new information SBC^B knows as a consequence of the execution of a plan enriches the information about a problem and BOLERO can generate a new plan. So the integrated system BOLERO-SBC shows a reactive behavior.

BOLERO has been applied to learn the procedure of how to perform the diagnosis of pneumonias. The experimental evaluation of BOLERO tells us that case-based planning is a good method for acquiring planning knowledge in medical domains, and that meta-level architectures are adequate to reach reactive planning. All the results obtained are satisfactory and encourage the use of case-based methods to cope with the strategic knowledge acquisition bottleneck.

Keywords: knowledge acquisition, learning, case-based reasoning, planning, reactive planning, meta-level architectures, reflective systems.

Agraïments

Les idees que han permès el desenvolupament d'aquest treball són el producte de discussions entre tots els membres de l'IIIA. En particular, vull agrair al director d'aquest treball de tesi, Enric Plaza, les seves aportacions, que han apuntalat tot el treball de recerca. A Carles Sierra vull agrair la seva ajuda desinteressada en les modificacions de MILORD, mercès a les quals he pogut realitzar les proves experimentals de la tesi. A Lluís Godó agraeixo la seva cooperació en la definició de les funcions que en aquest treball s'utilitzen, a més a més de la seva disponibilitat per donar-me un cop de mà sempre que m'ha fet falta. També estic amb deute amb Ramón López de Màntaras, que com a cap del nostre grup de recerca ha col·laborat en moltes discussions i ha contribuït amb suggeriments molt profitosos. A Walter Van de Velde li voldria agrair el seu especial interès mostrat en el treball durant la seva estada en l'IIIA. Gràcies també a tota la resta de companys de l'IIIA que heu estat al meu costat, en particular a en Jordi Levy i en Gabriel Valiente, que m'han ajudat a resoldre els meus problemes amb TeX.

També vull expressar el meu agraïment als meus amics, sense el suport dels quals aquest treball no hauria estat possible. També tinc un deute amb els ecòlegs, oceanògrafs, administratius, personal de manteniment, astrofísics i en general tots els companys del CEAB per les seves paraules d'encoratjament i la seva solidaritat. Agraeixo als col·legues de la Unitat Docent d'Electrònica i Informàtica de l'Escola d'Enginyers Tècnics de la Universitat Rovira i Virgili de Tarragona per permetre'm finalitzar el treball de tesi amb comoditat.

Per acabar, agraeixo a l'Andrés tot el seu suport en aquests anys, i especialment la seva paciència.



Aquesta tesi ha rebut els suport de les institucions i els projectes següents:

- ESPRIT II #2148: Valid Project
- Project AMP CYCIT 90/801
- Ministeri d'Educació i Ciència (beca de formació de personal investigador PN90)

Taula de continguts

Introducció	1
1 Preliminars	5
1.1 Planificació, informació incompleta i reactivitat	5
1.1.1 Planificació	5
1.1.2 El problema de la informació incompleta	7
1.1.3 Planificadors reactius	8
1.2 Planificació en els sistemes basats en coneixements	9
1.2.1 Les arquitectures metanivell	10
1.3 Adquisició de plans	14
1.3.1 Mètodes basats en l'enginyer de coneixements	14
1.3.2 Eines d'explicitació de coneixements	15
1.3.3 Aprenentatge automàtic	15
1.3.4 Sistemes híbrids	18
1.4 Raonament analògic, raonament basat en casos i planificació	19
1.4.1 La planificació basada en casos	22
1.5 Visió de conjunt	22
2 L'aproximació de BOLERO: planificació basada en casos	25
2.1 L'arquitectura metanivell BOLERO-SBC	27
2.2 Aprenentatge de plans per observació	28
2.2.1 Etapes en l'aprenentatge per observació	31
2.3 Construcció de plans amb informació incompleta	33
2.4 Avaluació de plans	36
2.5 Aprenentatge a partir de plans reïxits i plans imprecisos	37
2.6 Conclusions. Planificació reactiva basada en casos	38
3 L'arquitectura metanivell BOLERO-SBC	39
3.1 El nivell objecte	39
3.2 El metanivell	41
3.3 Regles de reflexió	42
3.4 El comportament reactiu de BOLERO-SBC	45
3.5 El guany de la integració	47

4	El sistema BOLERO	49
4.1	Representació dels coneixements	50
4.1.1	Fets	50
4.1.2	Situacions	52
4.1.3	Accions	52
4.1.4	Plans	54
4.1.5	Episodis	57
4.1.6	Traces	57
4.1.7	Casos	57
4.1.8	Memòria de plans	60
4.2	Visió global	62
4.3	Aprentatge per observació	63
4.3.1	Incorporació d'un cas	64
4.3.2	Generalització de fets	65
4.3.3	Aprentatge de condicions de terminació	67
4.4	Construcció de plans	69
4.4.1	Recuperació de plans	69
4.4.2	Adaptació de plans	72
4.4.3	Terminació de l'execució	72
4.5	Avaluació dels plans	76
4.5.1	Correctesa dels plans	76
4.5.2	Precisió dels plans	77
4.6	Aprentatge per l'experiència pròpia	79
4.6.1	Aprentatge a partir del reeiximent del pla	79
4.6.2	Aprentatge a partir d'un pla imperfecte	79
4.6.3	Aprentatge de la importància estratègica de les generalitzacions	79
5	Avaluació experimental de BOLERO	85
5.1	L'aplicació	85
5.1.1	Contingència	86
5.1.2	El problema de les classes alternatives	87
5.1.3	Fets irrelevants i coneixements irrelevants	89
5.2	Mètode d'avaluació	90
5.3	Resultats	92
5.3.1	Identificació	92
5.3.2	Predicció	94
5.3.3	Temps d'execució	99
5.3.4	Mesures relatives a l'arbre de plans	102
5.4	Variacions	105
5.4.1	Avaluació de plans multiexpert	105
5.4.2	Aprentatge per observació versus aprenentatge per experiència	107
5.4.3	Importància estratègica dels fets	111
5.4.4	Aprentatge de la importància estratègica dels fets	113
5.4.5	Efectes del paràmetre mdm	114
5.4.6	Efectes del biaix conservador	116
5.4.7	Reducció del conjunt de fets	118
5.5	Independència del domini	124

5.5.1	RENOIR	124
5.6	Conclusions	125
6	Treballs relacionats	127
6.1	Raonament basat en casos	127
6.1.1	PROTOS	130
6.1.2	CASEY	133
6.2	Planificació basada en casos	136
6.2.1	CHEF	136
6.2.2	SMART	139
6.3	Planificació reactiva	143
6.3.1	Theo-Agent	143
6.4	Adquisició de plans	145
6.4.1	ASK	146
6.4.2	PRODIGY/EBL	148
6.4.3	SAGE	151
6.5	Conclusions	153
7	Contribucions i treball futur	155
7.1	Avaluació de la recerca	155
7.2	Contribucions	157
7.3	Treballs futurs	158
A	Símbols utilitzats en la representació de coneixements de BOLERO	161
A.1	Fets	161
A.2	Situacions	161
A.3	Accions	161
A.4	Plans	162
A.5	Episodis	162
A.6	Traces	162
A.7	Casos	162
A.8	Nodes	162
B	T-normes i T-conormes	163
C	PNEUMON-IA: l'expert i el principiant	165
C.1	PNEUMON-IA: l'expert mestre	165
C.2	PNEUMON-IA*: el principiant	167
	Bibliografia	169

1	1. INTRODUCCIÓ
2	2. OBJECTIUS I JUSTIFICACIÓ
3	3. METODOLOGIA
4	4. RESULTATS
5	5. CONCLUSIONS
6	6. BIBLIOGRAFIA
7	7. ANNEXOS
8	8. RESUMEN
9	9. SUMMARY
10	10. ABSTRACT
11	11. RESUME
12	12. ZUSAMMENFASSUNG
13	13. SUMMARY
14	14. SUMMARY
15	15. SUMMARY
16	16. SUMMARY
17	17. SUMMARY
18	18. SUMMARY
19	19. SUMMARY
20	20. SUMMARY
21	21. SUMMARY
22	22. SUMMARY
23	23. SUMMARY
24	24. SUMMARY
25	25. SUMMARY
26	26. SUMMARY
27	27. SUMMARY
28	28. SUMMARY
29	29. SUMMARY
30	30. SUMMARY
31	31. SUMMARY
32	32. SUMMARY
33	33. SUMMARY
34	34. SUMMARY
35	35. SUMMARY
36	36. SUMMARY
37	37. SUMMARY
38	38. SUMMARY
39	39. SUMMARY
40	40. SUMMARY
41	41. SUMMARY
42	42. SUMMARY
43	43. SUMMARY
44	44. SUMMARY
45	45. SUMMARY
46	46. SUMMARY
47	47. SUMMARY
48	48. SUMMARY
49	49. SUMMARY
50	50. SUMMARY
51	51. SUMMARY
52	52. SUMMARY
53	53. SUMMARY
54	54. SUMMARY
55	55. SUMMARY
56	56. SUMMARY
57	57. SUMMARY
58	58. SUMMARY
59	59. SUMMARY
60	60. SUMMARY
61	61. SUMMARY
62	62. SUMMARY
63	63. SUMMARY
64	64. SUMMARY
65	65. SUMMARY
66	66. SUMMARY
67	67. SUMMARY
68	68. SUMMARY
69	69. SUMMARY
70	70. SUMMARY
71	71. SUMMARY
72	72. SUMMARY
73	73. SUMMARY
74	74. SUMMARY
75	75. SUMMARY
76	76. SUMMARY
77	77. SUMMARY
78	78. SUMMARY
79	79. SUMMARY
80	80. SUMMARY
81	81. SUMMARY
82	82. SUMMARY
83	83. SUMMARY
84	84. SUMMARY
85	85. SUMMARY
86	86. SUMMARY
87	87. SUMMARY
88	88. SUMMARY
89	89. SUMMARY
90	90. SUMMARY
91	91. SUMMARY
92	92. SUMMARY
93	93. SUMMARY
94	94. SUMMARY
95	95. SUMMARY
96	96. SUMMARY
97	97. SUMMARY
98	98. SUMMARY
99	99. SUMMARY
100	100. SUMMARY

Llista de figures

1.1	Components bàsics d'un planificador	6
1.2	Regla de tipus estímul-resposta	9
1.3	Esquema d'una arquitectura metanivell	11
1.4	Interaccions entre el nivell estratègic i el nivell del domini	12
1.5	Metaregla de planificació específica i metaregla de planificació de caràcter general	13
2.1	Sistema integrat resultat de la iteració entre BOLERO i un SBC	26
2.2	Dades dels casos b06v1 i g01v1	29
2.3	Representació dels casos b06v1 i g01v1 per episodis	31
2.4	Memòria de BOLERO després d'haver après per observació els casos b06v1 i g01v1	32
2.5	Representació esquemàtica de la resolució d'un cas	35
3.1	Sistema integrat resultat de la interacció entre BOLERO i un SBC	40
3.2	Regles del sistema basat en regles PNEUMON-IA	44
3.3	Seqüència de canvis de lloc d'acció en BOLERO-SBC	45
4.1	Esquema general de BOLERO	50
4.2	Relació d'inclusió entre les classes d'identificadors i identificadors de fets, i entre fets i identificadors en la aplicació PNEUMON-IA	51
4.3	Relacions del fet esput amb altres esdeveniments del domini de pneumònies	52
4.4	Jerarquia de plans	54
4.5	Exemples de plans	55
4.6	Traça i cas g22v1	58
4.7	Visió parcial dels coneixements del domini on es mostren diferents relacions entre diferents fets	66
4.8	Condicions de terminació adquirides de l'expert mestre per a la diagnosi de pneumònies	74
4.9	Variació del paràmetre mdm en funció del nombre de nodes terminals	75
5.1	Diferents classificacions de 4 casos.	88
5.2	Variació del nombre de nodes creats en memòria en funció del nombre de casos d'entrenament	93
5.3	Variació del percentatge de nodes en execució en funció del nombre de casos d'entrenament	94
5.4	Configuració bàsica: grau d'assoliment en la predicció	94
5.5	Nombre d'accions en memòria en funció del nombre de casos d'entrenament	95
5.6	Configuració bàsica: grau d'enfocament en la predicció	96

5.7	Variació del percentatge del nombre de nodes en execució en funció del nombre de casos d'entrenament	97
5.8	Evolució de la mitjana de nombre de canvis de pla que té lloc en la resolució d'un problema respecte al nombre de casos en memòria	98
5.9	Evolució del grau d'enriquiment i del grau de desencert en la predicció	99
5.10	Relació de les mitjanes dels temps necessaris per aprendre i resoldre un cas respecte al nombre de casos en memòria	100
5.11	Instantània de la memòria de BOLERO quan ha après per observació 79 casos	103
5.12	Instantània de la memòria de BOLERO després de la compactació	104
5.13	Nombre de nodes terminals en memòria, quan BOLERO aprèn per observació i quan aprèn per experiència, respecte al nombre de casos d'entrenament . . .	109
5.14	Temps necessari per aprendre un cas mitjançant l'aprenentatge per observació i l'aprenentatge per experiència	111
5.15	Variació del paràmetre mdf respecte a λ i en el nombre de casos en memòria	115
5.16	Variació del grau d'assoliment i el grau d'enfocament en la predicció en funció del paràmetre mdf	116
5.17	Configuració 6: Evolució de l'enfocament respecte al nombre de casos d'entrenament.	118
5.18	Variació del grau d'ordenació amb un conjunt estratègic de fets respecte al conjunt complet de fets	121
5.19	Diagrama de barres on es reflecteix la variació del nombre de fets per node en funció del nombre de casos en memòria	122
5.20	Temps de CPU consumit per BOLERO per aprendre un cas per observació depenent del nombre de casos d'entrenament, i si el conjunt de fets representats és complet o reduït	123
5.21	Temps de CPU consumit per a resoldre un cas en BOLERO-SBC en funció del nombre de casos en memòria, i el tamany del conjunt dels coneixements del domini	123
6.1	Coneixements de domini a PROTONS	130
6.2	Índexs per a la recuperació d'exemplars	131
6.3	Explicació causal generada per Heart Failure per diagnosticar un pacient . . .	134
6.4	Pla (recepta) per cuinar bròquil amb costelles	137
6.5	Regla d'aplicació específica sobre l'ingredient gamba	138
6.6	Representació dels coneixements a PRODIGY	140
6.7	Interacció entre el resolador de problemes per analogia NoLimit i el gestor de memòria SMART	141
6.8	Arbre de decisió per seleccionar una acció	144
6.9	Regla de tipus estímul-resposta	144
6.10	Exemples de regles d'estratègies en ASK	146
6.11	Representació d'un concepte objectiu a aprendre a PRODIGY	149
6.12	Representació d'esquemes de demostració a PRODIGY	150
6.13	Estratègia de caràcter general per resoldre el problema de les torres de Hanoi	151
6.14	Heurística per a detectar punts morts en la resolució de problemes	152
6.15	Estratègia per resoldre el problema de les torres de Hanoi	153
C.1	Nivells de domini i de control de Milord	166

C.2 BOLERO controla MILORD en l'aquitectura BOLERO-SBC	167
--	-----

Llista de taules

4.1	Algorisme de racionalització d'una traça	59
4.2	Algorisme global de BOLERO	63
4.3	Algorisme de resolució de problemes	64
4.4	Algorisme d'incorporació d'un cas en memòria	65
4.5	Fets de tres casos que s'han resolt amb el mateix pla final	68
4.6	Algorisme de recuperació de BOLERO	70
4.7	Algorisme d'aprenentatge de la importància estratègica dels fets	81
5.1	Temps necessari per resoldre un cas en la identificació i en la predicció	101
5.2	Grau d'enfocament en la predicció	106
5.3	Configuració 2: Grau d'enfocament en la predicció	108
5.4	Configuració 3: variació del grau d'enfocament en la predicció	112
5.5	Variació del grau d'assoliment i el grau d'enfocament en la predicció	113
5.6	Variació del grau d'enriquiment i el grau de desencert en la predicció	114
5.7	Configuració 7: Grau d'assoliment en la predicció	119
5.8	Configuració 7: Grau d'enfocament en la predicció	120

Introducció

Un dels focus d'interès de les ciències cognitives és entendre què és l'expertesa en la resolució de problemes, qualitat dels humans que és necessari capturar per tal de crear un model de comportament intel·ligent. Un principiant i un expert poden resoldre correctament un problema, però el segon és més eficient que el primer. L'expert recull les dades més útils o les que el poden guiar a saber quines dades són rellevants a un problema i, amb poques dades, genera les hipòtesis més versemblants. El principiant demana dades a cegues, necessita moltes dades i observacions, i genera moltes hipòtesis poc versemblants. Per tal com tots els experts han estat en el seu dia principiants, és lògic suposar que existeix un procés mitjançant el qual un principiant esdevé expert. Si analitzem la paraula expert en el diccionari, veiem que significa "versat en la coneixença d'una cosa per la pràctica" [34]. Per tant, la pràctica és la clau de l'expertesa. En efecte, segons un estudi realitzat en els Estats Units sobre l'ensenyament en medicina, s'ha observat que és realment en les pràctiques quan els alumnes aprenen. Concretament s'ha advertit que, de quatre anys d'estudi, dos de teòrics i dos més de pràctics, en els dos darrers anys els alumnes aprenen de nou tots els coneixements [143]. La resolució de problemes és, doncs, una manera d'adaptar els coneixements per ser aplicats adequadament. Per bé que un expert i un principiant disposen d'uns coneixements bàsics sobre un domini (teòric, basat en principis fonamentals, etc.), l'organització dels coneixements és diferent: l'expert té els seus coneixements organitzats mitjançant característiques que afavoreixen la solució dels problemes que usualment es troba, i el principiant organitza els seus coneixements segons característiques superficials, i.e. no directament vinculades a la resolució dels problemes. Per exemple, l'expert té en compte el cost dels recursos disponibles per trobar una solució, la urgència de trobar una solució (per exemple, si *el pacient s'està deshidratant* el metge ha de donar un diagnòstic ràpidament), etc. El principiant, en canvi, es fixa en detalls superficials com l'edat del pacient [52]. L'organització dels coneixements en l'expert és una conseqüència de la seva experiència a resoldre una important quantitat de problemes o casos. En psicologia cognitiva s'ha demostrat àmpliament la importància de l'aprenentatge per l'experiència ([37, 137, 131, 25, 35], etc.). En intel·ligència artificial (IA) McCarthy anunciava en 1958: "el nostre darrer objectiu és el de construir programes que aprenguin de la seva experiència tan efectivament com ho fan els humans" [42].

Des de 1958 la intel·ligència artificial ha evolucionat molt i s'han desenvolupat sistemes basats en coneixements (SBC) que realitzen tasques complexes, com per exemple, la diagnòstic mèdica. Per dur a terme la tasca que han de realitzar, els SBC necessiten dels coneixements que posseeixen els experts, i.e., els coneixements estratègics obtinguts mitjançant la pràctica. Els coneixements estratègics permeten als SBC generar plans d'actuació per tal de resoldre problemes. Com proporcionar plans als SBC i com usar-los és la finalitat d'aquest treball de tesi, que s'ha consolidat amb el desenvolupament d'un nou sistema: BOLERO.

La idea embrionària de BOLERO ha estat la de desenvolupar un sistema que aprengui

de l'experiència pròpia, de manera que a partir d'uns coneixements de principiant s'obtingui un sistema que mostri el comportament d'un expert mitjançant la pràctica. Però BOLERO no és un sistema aïllat, sinó que és fruit de molts anys de recerca a l'Institut d'Investigació en Intel·ligència Artificial (IIIA¹). L'experiència de l'IIIA en el desenvolupament de SBC s'inicià amb la construcció d'un shell de sistemes experts, MILORD [142], sobre el qual s'han desenvolupats diferents aplicacions mèdiques. El desenvolupament de MILORD va generar diferents línies de recerca, entre les quals tècniques d'aprenentatge de coneixements estratègics que permetessin realitzar una resolució de problemes més eficient [142]. L'objectiu de BOLERO és, doncs, l'aprenentatge i la generació de plans perquè es millori l'eficiència en la resolució de problemes d'un SBC desenvolupat en MILORD. En aquesta tesi es presenta el treball de recerca desenvolupat per aconseguir aquest objectiu.

Esquema dels capítols de la tesi

La tesi consta de set capítols i tres apèndixs. En el primer capítol es proporcionen coneixements preliminars que permeten entendre l'entorn d'investigació que ha motivat el desenvolupament d'un sistema per a l'aprenentatge i generació de plans. Així, es realitza una revisió de l'estat de la planificació, presentant el problema de la informació incompleta i incerta que els SBC han de manipular amb aplicacions reals, que conclou en el desenvolupament de planificadors reactius. També es mostra la presència de la planificació en els sistemes basats en coneixements, fent especial èmfasi en els sistemes d'arquitectura metanivell. Atès que la planificació requereix coneixements estratègics (i.e. els coneixements que tenen els experts humans), primerament es passa revista als mètodes d'adquisició de coneixements emprats fins avui en dia, que estan agrupats en quatre grans grups: mètodes basats en l'enginyer de coneixements, eines d'explicitació de coneixements, aprenentatge automàtic i sistemes híbrids. Entre els sistemes d'aprenentatge automàtic per a l'adquisició de coneixements, es destaca els sistemes basats en casos que utilitzen una visió particular de raonament analògic. Finalment, tots els problemes oberts i línies d'investigació que tenen plantejats tant la planificació com l'adquisició de coneixements es posen en comú.

Les inquietuds i la problemàtica plantejades en el capítol primer tenen un desenllaç en el capítol dos amb el desenvolupament d'un nou planificador basat en casos, BOLERO, que proporciona una nova aproximació a l'aprenentatge i generació de plans. En primer lloc se situa BOLERO dins de l'arquitectura metanivell BOLERO-SBC, on té lloc la resolució de problemes. BOLERO permet dins d'aquesta arquitectura millorar el rendiment d'un SBC i desenvolupar una planificació reactiva. En aquest capítol s'introdueix precisament com els casos proporcionats per un mestre són organitzats i emmagatzemats en BOLERO de manera que es poden reusar per resoldre problemes. També es mostra com emmagatzemant i organitzant els problemes resolts es pot aprendre a partir de l'experiència. És interessant observar que en l'aproximació de BOLERO a la planificació basada en casos existeix un biaix afavoridor al desenvolupament d'aplicacions com la diagnosi mèdica.

Els capítols 3 i 4 són capítols tècnics. En el capítol 3 es fa una descripció del sistema metanivell BOLERO-SBC. En la descripció de cadascun dels nivells que componen l'arquitectura es destaca el fet que BOLERO-SBC està constituït per dos sistemes no uniformes, que poden estar desenvolupats autònomament. L'arquitectura integra ambdós sistemes de manera que cooperen en la resolució de problemes. El comportament de

¹Fins al 1992, els membres de l'IIIA constituïen el Grup de Recerca en Intel·ligència Artificial i Lògica del Centre d'Estudis Avançats de Blanes (GRIAL, CEAB).

BOLERO-SBC esdevé, finalment, un comportament reactiu, on a més a més s'aconsegueix aprendre de l'experiència i millorar el mateix funcionament del sistema.

En el capítol 4 es detalla minuciosament el sistema BOLERO: la representació dels coneixements, l'aprenentatge per observació, la construcció de plans, l'avaluació de plans i l'aprenentatge per l'experiència pròpia.

En el capítol 5 es fa una avaluació experimental del sistema sobre una aplicació del món real. Les particularitats de l'aplicació, la diagnòsi mèdica de pneumònies, presenta dificultats tant per a l'aprenentatge com per a la planificació. Per realitzar l'avaluació experimental de BOLERO es descriu i se segueix un mètode d'avaluació basat en un conjunt de mesures. Els resultats obtinguts ens permeten formular un conjunt de conclusions, algunes no esperades, sobre els mètodes desenvolupats. L'avaluació de BOLERO obre les portes a nous problemes que requereixen un estudi i que es presenten en el capítol 7 com a línies de recerca futures.

El capítol 6 recull diferents treballs relacionats amb la recerca feta en aquesta tesi i que d'una manera o una altra han influït en el desenvolupament de BOLERO. Així, es revisen diferents sistemes basats en casos, planificadors basats en casos, planificadors reactius i sistemes d'adquisició de coneixements en general. De l'anàlisi dels treballs relacionats podem concloure que BOLERO presenta aportacions originals com són la combinació de la planificació i l'aprenentatge.

En el darrer capítol, el capítol 7, es proporciona les aportacions amb què contribueix aquesta tesi en la planificació reactiva i l'adquisició de coneixements. El treball desenvolupat ha donat peu al sorgiment de nous problemes que obren noves línies de recerca per al futur.



El sistema BOLERO va començar a ser dissenyat l'any 1989, i s'ha implementat en Common Loops Object System (CLOS) en una SUN 4/260. Alternativament, s'ha usat una SUN SPARC Station per realitzar proves experimentals. Per tal de poder experimentar amb BOLERO, s'ha generat una versió de MILORD en una SUN 4/260 (la versió original era en un VAX), a la qual s'han traduït dues de les aplicacions de MILORD [142]: PNEUMONIA [166] i RENOIR [16].

Capítol 1

Preliminars

Aquest capítol desenvolupa una visió general sobre la planificació, particularment en els sistemes basats en coneixements, i fa un recompte dels mètodes actuals per a l'adquisició de plans, i d'entre aquests, el raonament basat en casos. L'objectiu del capítol és proporcionar el marc adequat i la terminologia per comprendre la problemàtica actual de l'adquisició de plans i justificar el desenvolupament d'un nou planificador. En la primera secció s'introdueix la planificació i els problemes que té plantejats actualment. A continuació, en la secció 1.2 es presenten algunes de les aproximacions que s'han fet de la planificació en els sistemes basats en coneixements en els quals la presència de coneixements estratègics o de planificació ha esdevingut una pràctica habitual. Després, en la secció 1.3 es recullen els diferents mètodes desenvolupats per a l'adquisició de coneixements estratègics, i en la secció 1.4 s'introdueix un mètode en particular, el raonament basat en casos. Finalment, en la secció 1.5 es proporciona una visió de conjunt de totes les seccions anteriors.

1.1 Planificació, informació incompleta i reactivitat

El propòsit d'aquesta secció és introduir què és la planificació en IA, quins problemes té plantejats i com els resol. En l'apartat 1.1.1 es descriu el que ha estat fins ara la planificació en IA. En l'apartat que segueix (secció 1.1.2) es destaca la qüestió de la informació incompleta, un problema que han d'afrontar els planificadors quan treballen amb aplicacions reals. Finalment, en l'apartat 1.1.3 es presenta una tècnica d'aparició recent, la planificació reactiva, pensada especialment per superar el problema de la informació incompleta.

1.1.1 Planificació

Un planificador és responsable de generar un pla o conjunt d'accions que dirigeixen la realització d'una tasca. Tradicionalment la planificació s'ha formulat com un problema de cerca d'un camí (pla) que condueixi d'un estat inicial a un conjunt d'estats finals o objectius [114]. L'espai d'estats el constitueixen un conjunt d'operadors (accions) que permeten accedir a un estat des d'un altre estat. D'aquesta manera, algorismes de cerca heurística (com ara, la generació de subobjectius, el *backtracking* controlat, la cerca oportunista i altres estratègies similars) han estat aplicats en la construcció de plans. Les heurístiques són útils als planificadors per tal de no construir plans a cegues, és a dir, per assaig i error, però no han resultat suficients per a controlar problemes com la interacció entre

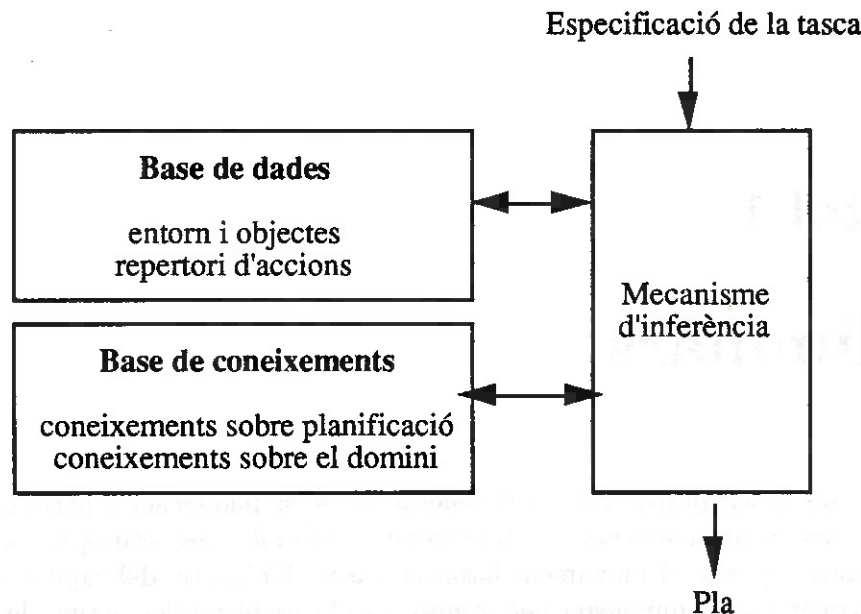


Figura 1.1: Components bàsics d'un planificador (de [152]).

objectius [36]. La incorporació de coneixements, ja siguin els propis del domini del problema a resoldre, com els del procés de planificació en si mateix, ha estat el següent pas evolutiu dels planificadors.

Els components bàsics d'un planificador són una base de dades, una base de coneixements i un motor d'inferències [152] (figura 1.1). La base de dades la constitueixen totes les dades que es coneixen sobre el món, així com el conjunt d'accions possibles. La base de coneixements conté coneixements sobre el domini del problema que es vol resoldre o sobre el mateix procés de planificació. En aquest sentit, un planificador és independent del domini si es pot utilitzar per resoldre una àmplia varietat de problemes [170]. El mecanisme d'inferència utilitza els coneixements sobre el domini o de planificació i les dades conegudes (base de dades) per construir plans que condueixin de l'estat inicial a l'estat final. Per tal de generar plans, el mecanisme d'inferència pot utilitzar qualsevol dels mètodes de cerca heurística.

A mesura que la planificació ha evolucionat han sorgit noves tècniques per tal de construir plans més precisos (que s'acostin més a la solució òptima o pla òptim). Un recull d'aquests mètodes és el següent:

- **linearització dels plans:** un planificador és lineal si construeix plans generant seqüencialment les accions (STRIPS [54]). Si els plans poden expressar-se com un ordre parcial entre accions, es tracta d'un planificador no-lineal (NOAH [136], NONLIMIT [165, 161], MOLGEN [146], SIPE [170]). La planificació lineal assumeix que els objectius conjuntius són independents i additius. Però, evidentment, existeixen interaccions entre els objectius que no són additives, i el planificador ha de fer marxa enrera (*backtrack*) en la construcció dels plans. La planificació no-lineal evita prendre compromisos prematurs sobre l'ordre de les accions i així el planificador no ha de desfer seleccions arbitràries basades en informació insuficient.
- **utilització de variables i restriccions:** les variables permeten construir plans deixant alguns objectes sense identificar mentre no existeixi alguna restricció que forci

la instanciació (mínim compromís o *least-commitment strategy*) (MOLGEN [147], SIPE [170]).

- jerarquització dels plans: un planificador és jeràrquic si els coneixements estan representats en diferents nivells, uns més abstractes (generals) que altres, de manera que un pla es pot representar en diferents nivells de detall (MOLGEN [146], NONLIMIT [164, 161], SIPE [170]).
- metaplanificació: utilització de coneixements sobre el procés de planificar pròpiament dit (vegeu la propera secció 1.2).
- replanificació: sense necessitat de començar de nou a planificar es pot modificar un pla davant de qualsevol anomalia (SIPE [170]).
- adaptació de plans: reutilització de plans contruïts amb anterioritat i adaptació a un problema nou (CASEY [88]).

Una qüestió important en la planificació és el control dels efectes de les accions, conegut com el problema del context (*frame problem*). Suposem que les accions es representen per perelles de pre-condicions i postcondicions sota la forma d'operadors o regles. Els efectes d'una acció depenen del context en el qual s'executen (per exemple, l'acció "subministrar penicil·lina al pacient", encara que sigui adient per al tractament d'una malaltia, té efectes diferents depenent de l'edat del pacient, de la gravetat, etc). Aleshores, per expressar els efectes de les accions es necessiten tants axiomes com situacions en els quals es poden aplicar les accions. És a dir, es produeix una explosió en el nombre de regles o operadors que calen per expressar les accions i, consegüentment, en el cost computacional necessari per inferir el resultat d'una acció [170, 69, 5]. Qualsevol mecanisme utilitzat per controlar les inferències en la planificació ha de ser prou flexible com per fer de la planificació un problema tractable, i a la vegada suficientment complex perquè permeti representar els diferents estats possibles en la construcció d'un pla [152].

1.1.2 El problema de la informació incompleta

Clàssicament, en planificació, el pla es construeix en un sol pas: s'assumeix que el món no canvia mentre un pla és dut a terme. En les aplicacions del món real les dades no segueixen cap ordre, no són sistemàtiques, i els coneixements sobre un domini són incerts i incomplets, raó per la qual aquest tipus de pla fracassa. Per exemple, suposem que un pacient de 25 anys amb factors de risc d'infecció per HIV es presenta en el servei d'urgències d'un hospital amb una clínica de pneumònia. Suposem també que l'hospital està proveït d'un sistema expert que determina un pla per tal de determinar el tipus de pneumònia que pateix el pacient. Aquest pla consisteix en una seqüència d'accions: realitzar una radiografia del tòrax, una hemografia i una prova de Gram-esput. En la primera prova s'obté una radiografia que indica una pneumònia intersticial. Un metge, amb aquesta informació, i tenint en compte els factors de risc del pacient per HIV, suggeriria immediatament la possibilitat que el malalt tingues la sida i decidiria realitzar una acció diferent (com per exemple una rentada bronco-alveolar, RBA) per determinar una pneumònia causada per *Pneumocystis carinii*. El sistema expert, però, ja ha decidit el pla i insistirà en la resta de proves (hemografia, Gram-esput), que deixen de tenir rellevància en la diagnòsi del pacient. En resum, el pla d'aquest sistema expert fracassa en adaptar-se al comportament desitjat.

En l'elaboració d'un pla, un planificador té una visió parcial del món, sobre la qual genera un conjunt de supòsits que fan el pla admissible. En el transcurs de l'execució del pla poden produir-se un conjunt d'esdeveniments que facin fracassar el pla, bé perquè contradiuen els

supòsits amb què s'ha desenvolupat el pla, bé perquè són nous esdeveniments que no s'han tingut en compte [139]. El grau de completesa de la informació que el planificador disposa quan construeix un pla determina el grau amb el qual el planificador distingeix diferents situacions (estats d'un problema) [75]. D'aquesta manera, amb informació més completa el risc de fracàs d'un pla és menor. Les situacions reals, però, són molt complexes i incertes [3]. Moltes de les situacions del món no poden ser representades completament, no existeixen mitjans per recopilar tota la informació necessària i, al mateix temps, la informació de què es disposa és incerta donat que altres processos a més del planificador modifiquen l'entorn.

1.1.3 Planificadors reactius

El problema de la informació incompleta ha donat lloc al desenvolupament de planificadors *reactius* o la *planificació reactiva*. Els planificadors reactius són planificadors que adequen els seus plans als canvis de l'entorn [58]. Donada una situació, el planificador reactiu només necessita d'un pla parcialment elaborat per començar a executar-lo [57]. La imbricació de l'elaboració o refinament d'un pla i la seva execució és un propietat inherent en aquests planificadors. Un canvi en una situació implica l'elaboració d'un nou pla, que és executat immediatament.

Els planificadors reactius mostren robustesa en el sentit que aquest tipus de planificadors continuen tenint un comportament plausible en situacions no anticipades [75]. Per un costat, si la informació de què es disposa del món és poc completa o feble, un planificador reactiu pot presentar un comportament gràcilment degradat, però no fracassa. Per exemple, quan el planificador no disposa d'informació suficient, pot utilitzar accions d'adquisició d'informació que en executar-les proporcionin suficient informació al planificador i, aleshores i no abans (després d'haver executat l'acció de recollida d'informació), generar un pla precís. La utilització d'accions que recullen noves informacions sobre l'entorn és una manera de mitigar la manca d'informació [7, 142] i és una tècnica emprada pels experts que els facilita la generació d'hipòtesis versemblants. Per exemple, si s'ha de diagnosticar una malaltia a un pacient, i només se sap que ha ingressat a l'hospital amb una *clínica de pneumònia*, el més natural és que el metge faci un reconeixement exploratori al malalt (que consisteix a saber si té febre, si és drogoaddicte, etc.) abans de continuar amb un pla d'accions per diagnosticar al pacient.

El problema principal de la planificació reactiva, donada la imbricació de la planificació i l'execució, és establir quan planificar i quan executar [7]. Per una banda, si el planificador gasta la major part del temps elaborant plans precisos, quan un pla s'acaba, el pla pot esdevenir inadequat perquè l'entorn ha canviat o perquè la informació obtinguda al principi de l'execució del pla fa inútil la resta de les accions. Per un altra banda, si el planificador només s'ocupa dels canvis de l'entorn, és possible que mai arribi a desenvolupar un pla complet que assoleixi els seus objectius. L'equilibri entre la precisió en la resposta i el temps de resposta defineix el cost computacional de la planificació envers els beneficis que comporta un pla més exacte. Els treballs de Dean [49] i Etzioni [53], juntament amb els de Russell i Wefald [132, 134], es dirigeixen cap a la definició de mesures d'utilitat en la planificació. A més a més, existeixen ocasions en les quals el temps de resposta a un determinat esdeveniment és molt important, i cal reaccionar immediatament. Per exemple, si hom diagnostica a un pacient i en el transcurs de les proves el pacient té una aturada cardíaca, és imprescindible generar un pla d'atenció immediata i interrompre la resta de proves que conformen el pla per diagnosticar la malaltia.

Entre els planificadors reactius, cal distingir els planificadors d'estímul-resposta,

planificadors reactius que responen ràpidament als canvis a l'entorn però no planifiquen a llarg termini. És a dir, que ni ataquen el problema del context com ho fan la resta dels planificadors, ni resolen problemes. Simplement sobreviuen en un entorn dinàmic (Pengi [3]). El problema fonamental dels planificadors estímul-resposta, més enllà de no ser útils per a la resolució de problemes complexos, és que poden induir cicles en l'execució [19, 139]. Per exemple, una regla com la de la figura 1.2, on el resultat de l'acció seleccionada (*posicionar-el-robot-per-agafar*) es reflecteix en una condició de la premissa (*c6*) pot ser aplicada més d'un cop si l'acció seleccionada finalment no es consolida. És necessari determinar en quins moments l'aplicació d'una regla és útil. Un altre problema dels mecanismes d'estímul-resposta és la resolució de conflictes entre reaccions que poden tenir lloc en un mateix moment.

REGLA2:

- Si (c1) una tassa és al camp visual
 (c2) la tassa no és a la paperera
 (c3) la bateria està al 100%
 (c4) la tassa no ha estat agafada
 (c5) la tassa és a prop
 (c6) la tassa no està ben situada

Llavors acció-seleccionada = *posicionar-el-robot-per-agafar*

Figura 1.2: Regla de tipus estímul-resposta. L'èxit de l'acció *posicionar-el-robot-per-agafar* s'avalua en la situació de la tassa (i.e. *c6*) (adaptat de [Theo]).



Per finalitzar aquesta secció, podem concloure que s'han desenvolupat un gran nombre de mètodes per tal de contruir plans millors amb els quals afrontar un conjunt de problemes com ara la interacció entre objectius conjuntius. Un problema important que té plantejat la planificació és el problema de la informació incompleta i incerta. Es tracta d'un problema important, car es presenta amb la manipulació d'aplicacions reals, aplicacions amb les quals finalment han de treballar els planificadors. Els planificadors reactius afronten directament aquest problema de la informació incompleta i incerta, i són una línia de recerca cap on molts investigadors estan dirigint els seus esforços.

1.2 Planificació en els sistemes basats en coneixements

La planificació ha esdevingut important a causa del desenvolupament d'aplicacions reals. La complexitat de les tasques que els sistemes basats en coneixements (SBC) han de dur a terme requereixen de coneixements estratègics o de planificació per il·luminar la resolució de problemes. Els plans fan que el procés de resolució de problemes sigui eficient, controlant els recursos necessaris per resoldre un problema i adaptant el procés de resolució als canvis a l'entorn [149]. D'aquesta manera, la distinció entre coneixements de domini i coneixements de control (estratègics o de planificació) ha esdevingut una pràctica habitual en els SBC. Els coneixements de domini consisteixen en descripcions dels objectes del domini d'expertesa del sistema de raonament (propietats dels objectes, relacions entre els objectes, etc). Els

coneixements de control estan conformats per estratègies que indiquen com organitzar les tasques del domini, com construir plans per assolir els objectius, etc. [158]. En aquest sentit, els coneixements de control són la implementació del que a nivell epistemològic s'anomena coneixements estratègics [22, 40]. Els coneixements estratègics es defineixen com els coneixements sobre què fer a continuació [63], i el raonament estratègic raona sobre les accions, de la mateixa manera que ho fa la planificació. Per aquesta raó, coneixements de control, coneixements estratègics i coneixements de planificació són termes utilitzats alternativament en aquest treball.

Els coneixements de planificació proporcionen flexibilitat i capacitat de control a un KBS envers diferents tipus de problemes, de la mateixa manera que la degradació del sistema envers problemes difícils resulta més gràcil (*graceful degradation* [22]). Mercès als coneixements estratègics, un SBC pot construir plans que adaptin els objectius als recursos disponibles o a les intencions de l'usuari, trobar *impasses* (destrets) i remeis per a *impasses*, etc. Per exemple, en una aplicació de medicina, per saber la malaltia d'un pacient es poden realitzar una gran quantitat de proves: anàlisi de sang, anàlisi d'orina, biòpsies, etc. Les proves representen un cost econòmic per a l'hospital, un cost en el temps necessari per realitzar-les i un cost per al pacient avaluat en el risc a què és sotmès a fi de realitzar alguna d'aquestes proves. En lloc de realitzar totes les proves, un planificador ha d'escollir les que, donades les característiques del pacient, proporcionen un diagnòstic fiable amb un cost i un termini de temps raonables.

Mentre que els coneixements de domini es representen explícitament en els SBC, els coneixements de control formen part en moltes ocasions del mateix intèrpret del sistema. Els arguments a favor de distingir un nivell explícit de coneixements de control estan àmpliament discutits en la literatura [22, 40, 39, 38, 158, 48], i es concreten en els avantatges següents:

1. Un sistema amb una representació explícita dels coneixements de control és més fàcil de desenvolupar, corregir i mantenir.
2. Una base de coneixements de domini pot ser aprofitada per diferents sistemes, cadascun amb un propòsit i coneixements de control diferents.
3. L'adquisició de coneixements de control és més fàcil.
4. Les explicacions del comportament d'un sistema són més riques, ja que a més a més de justificar *per què* s'ha trobat una solució (coneixements de domini) es poden proporcionar explicacions de *com* s'ha resolt (coneixements de control).
5. La construcció de la base de coneixements del domini pot ser merament descriptiva sense imbricar qüestions d'eficiència del sistema.

Així, la representació explícita dels coneixements estratègics en els SBC ha donat lloc al desenvolupament de les arquitectures metanivell, que tot seguit s'introdueixen.

1.2.1 Les arquitectures metanivell

Les arquitectures metanivell (*meta-level architectures* [22, 142, 158]) proporcionen una primera aproximació en la representació explícita dels coneixements estratègics. En aquestes arquitectures es distingeixen dos nivells: el nivell objecte i el metanivell¹. Cada nivell es pot veure com un sistema independent amb un llenguatge de representació, una base de coneixements i un mecanisme d'inferències (figura 1.3). Els coneixements del nivell objecte són coneixements d'un domini d'interès (coneixements de domini). Els coneixements del metanivell són coneixements sobre els coneixements del nivell objecte: coneixements de

¹Existeixen arquitectures com MILORD [142] que estenen les arquitectures metanivell a 4 nivells.

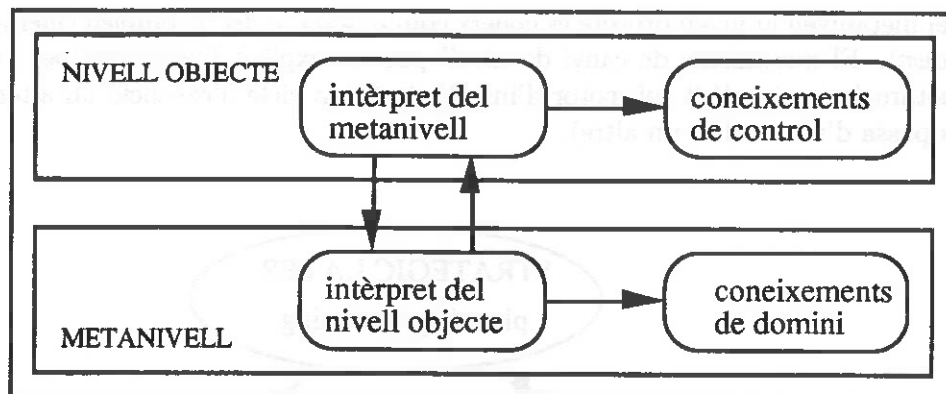


Figura 1.3: Esquema d'una arquitectura metanivell.

control, coneixements que descriguin els coneixements del nivell objecte, etc. Pels propòsits del nostre treball suposem que els coneixements del metanivell són coneixements estratègics. Ambdós nivells cooperen en la resolució de problemes. En el nivell objecte es realitzen inferències que condueixen a la resolució de problemes i el metanivell s'encarrega de controlar i dirigir les inferències del metanivell per tal d'aconseguir una resolució de problemes eficient. En aquest sentit, el metanivell pot planificar les tasques del nivell objecte i raonar sobre l'estat del nivell objecte en la resolució d'un problema. La capacitat del metanivell de raonar sobre el nivell objecte ha fet que les arquitectures metanivell siguin també anomenades reflexives (*reflective systems* [102]), car raonen sobre el seu propi estat intern. En un sistema reflexiu el metanivell és també capaç d'aturar i engegar l'execució del nivell objecte, i és capaç d'alterar l'estat intern del nivell objecte (mitjançant, per exemple, plans d'actuació).

En una arquitectura metanivell podem distingir tres elements bàsics [158]:

1. El primer és la visió parcial o model M que el metanivell té del nivell objecte. Per exemple, el metanivell pot conèixer només el conjunt d'inputs del nivell objecte com a característiques de control, és a dir, característiques que permeten avaluar l'estat del procés de resolució d'un problema en el nivell objecte en un moment donat (eg. *inference structures* [22], *control features* [63], *heuristic role annotations* [145], figura 1.4). El procés d'inferència del metanivell depèn en gran mesura d'aquest model.
2. El segon és un conjunt de regles de reflexió que regulen la connexió entre els dos nivells. Les regles ascendents (*upwards reflection rules*) arbitren la informació del nivell objecte que necessita conèixer el metanivell d'acord amb el model parcial M . Per exemple, en una regla ascendent es pot descriure com un conjunt d'informació en el nivell objecte esdevé característica de control en el metanivell. Les regles descendents (*downwards reflection rules*) normalitzen les modificacions que el metanivell realitza sobre el nivell objecte. Per exemple, en una regla descendent es pot explicitar com un pla construït en el metanivell estructura les tasques del nivell objecte. Les regles reflexives asseguren una connexió causal entre els dos nivells de l'arquitectura.
3. El tercer és el mecanisme que determina en quin nivell és actiu el procés de raonament. Així, mentre que hi ha sistemes que només recorren al metanivell en situacions d'*impass*, altres realitzen la major part de la activitat en el metanivell, i arriben fins i tot a simular el nivell objecte. El pas del procés de raonament del nivell objecte cap al metanivell es coneix com a *ascens al metanivell* (*meta-level lift up*), i el canvi del procés de raonament

del metanivell al nivell objecte es coneix com a *descens del metanivell* (*meta-level reflect down*). El mecanisme de canvi de nivell pot ser explícit (expressat, per exemple, per metaregles) o implícit (el motor d'inferència té un cicle d'execució on alternativament es passa d'un nivell a un altre).

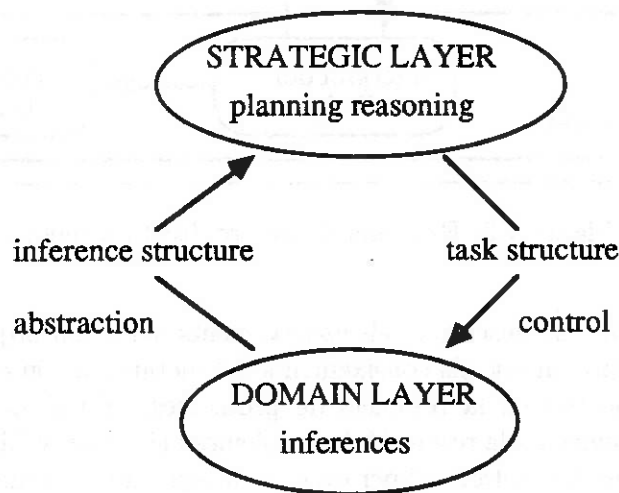


Figura 1.4: Interaccions entre el nivell estratègic i el nivell del domini (de [22])

Les arquitectures metanivell més comunes són els sistemes basats en regles, els sistemes de pissarra i els mecanismes d'agenda.

Sistemes basats en regles

En aquest apartat volem incloure tots els sistemes basats en regles que caracteritzen un conjunt especial de regles per controlar el procés d'inferència en la resolució de problemes (NEOMYCIN [38], MILORD [142], MU [41]). Aquestes regles de control o metaregles expressen coneixements estratègics específics del domini o generals. Per exemple, la metaregla que es mostra a la figura 1.5a expressa coneixements estratègics específics per a la diagnosi de pneumònia; la metaregla de la figura 1.5b exhibeix coneixements estratègics generals per a la resolució de problemes.

El motor d'inferències aplica les metaregles d'acord amb un ordre preestablert (NEOMYCIN) o quan la premissa d'una metaregla és satisfeta (MILORD). El canvi de nivell (introspecció) es dona en el primer cas quan s'acaba un acció, i en el segon cada cop que es coneix nova informació. Així, l'aplicació de metaregles per a l'activació de les premisses proporciona una planificació dinàmica on els plans no són fixos, sinó que són construïts per resoldre cada problema. El sistema MILORD, a més a més, alterna la construcció d'un pla amb la seva execució proporcionant un pla d'acord amb la informació coneguda sobre un problema en cada moment donat. Aquesta és una característica molt important, ja que el coneixement del món real és incert i incomplet i, en conseqüència, no es coneixen en el moment de la implementació del sistema expert totes les possibles situacions en què es pot trobar un planificador. MILORD, en aquest sentit, presenta un comportament reactiu.

M03025: Si el pacient té la sida
 Llavors mòduls a visitar: bacteriana-atípica pneumocystis-carinii
 TBC citomegalovirus criptococ nocardia
 aspergillus pneumococ enterobacteriàcia
 amb certesa: possible

Metarule1: Clarify.questions
 IF goal(clarify.finding \$finding1) ^
 clarifying.questions (\$finding1 \$finding2) ^
 not(value-known \$finding2)
 THEN goal(findout \$finding2)

Figura 1.5: (a) Metaregla de planificació específica del domini de pneumònies: si el pacient té sida, aleshores els objectius que s'intenten demostrar són bacteriana-atípica, pneumonistis-carinii, (de [Sierra]). (b) Metaregla de planificació de caràcter general: si l'objectiu actual és clarificar el fet finding1, i finding1 pot ser esbrinat mitjançant finding2, i finding2 és desconegut, aleshores intentar descobrir el valor de finding2 (de [Wilkins]).

Arquitectures de pissarra

Les arquitectures de pissarra (*Blackboard Architectures* [46, 116, 115]) tenen tres components principals: la pissarra, un conjunt de fonts de coneixements (*knowledge sources*) i el controlador². La pissarra és una base de dades estructurada en diferents nivells, que recull els resultats del procés de resolució d'un problema. Les fonts de coneixements (FC) contenen la informació necessària per resoldre un problema. Les FC estan representades per una pre-condició i un cos constituït per procediments, conjunts de regles o axiomes. Quan la pre-condició d'una FC es satisfà, la FC s'activa i modifica la pissarra segons els seus coneixements. D'aquesta manera, les FC responen oportunament als canvis que es produeixen en les dades de la pissarra. El motor d'inferències controla l'activació de les FC bé per una llista de preferència entre les FC, bé per l'aplicació d'estratègies de planificació. Les primeres arquitectures de pissarra implementaven les estratègies de planificació en el mateix motor d'inferències (HERSAY-II [98]), i utilitzaven l'organització jeràrquica de les FC per controlar el sistema. Posteriorment es van construir sistemes (BB1 [70]) que incloïen una FC especial amb coneixements sobre planificació.

Mecanismes d'agenda

Els mecanismes d'agenda es basen en una memòria de dades, un intèrpret i una agenda que s'utilitza per controlar el procés de resolució d'un problema. Les tasques del sistema són anotades en l'agenda i són executades per un ordre determinat. L'intèrpret pot alterar l'ordre o prioritat de les tasques d'acord amb l'estat de la memòria. En els primers mecanismes els coneixements de selecció de tasques estaven implícits en l'intèrpret. MOLGEN [146] va incorporar la *metaplanificació* en aquestes arquitectures amb la distinció d'un nivell propi de

²El motor d'inferències o controlador de les arquitectures de pissarra és anomenat en la literatura com a planificador (*scheduler*). La raó principal és que en els primers sistemes els coneixements de planificació romanien implícits en el motor d'inferències.

coneixements estratègics. En el nivell estratègic de MOLGEN es defineixen operadors que permeten focalitzar el procés de resolució d'un problema en una tasca particular, continuar tasques que havien estat suspeses per manca d'informació, prendre decisions temptatives quan el pla és encara incomplet i no es distingeix una tasca per focalitzar, i desfer les temptatives si no poden dur-se a terme.



Al llarg d'aquesta secció hem vist com la tendència actual en el desenvolupament de qualsevol sistema basat en coneixements és la de distingir els coneixements de domini i els coneixements de planificació. En particular, les architectures metanivell diferencien explícitament ambdós tipus de coneixements. Els avanços produïts en la construcció de SBC es troben bloquejats per l'adquisició de coneixements, disciplina que roman com el coll d'ampolla de la IA. En la secció següent es tracta precisament de l'adquisició de coneixements de planificació.

1.3 Adquisició de plans

L'adquisició de coneixements es defineix com l'explicitació i l'anàlisi de dades sobre l'expertesa per tal de construir sistemes experts [123, 22]. En aquest procés de transferir i transformar coneixements de les persones cap als computadors existeixen dos problemes clàssics: el problema dels coneixements implícits (*tacit knowledge*) i el problema de la modelització dels coneixements.

El problema dels coneixements implícits rau en el fet que gran part de l'expertesa és un coneixement tàcit que es basa en experiències anteriors. Aleshores, quan s'entrevista un expert, aquest és incapaç d'explicar els seus coneixements si no té cap cas concret sobre la taula que l'ajudi a explicar el seu raonament. Moltes vegades, l'expert no sap explicar com resoldre el seu problema si no és donant-hi la solució aproximada.

La finalitat de l'adquisició de coneixements és transformar els coneixements dels experts en un model computacional. Les dades que es manipulen solen ser extensives, qualitatives i no sistemàtiques, per la qual cosa la modelització no resulta en absolut trivial.

Existeixen diferents mètodes per a l'adquisició de coneixements que es poden agrupar en tres grans grups: mètodes basats en l'enginyer de coneixements, eines d'explicitació de coneixements i aprenentatge automàtic. El corrent actual és el disseny de sistemes híbrids on es combinen diferents mètodes. A continuació es presenten cadascun dels diferents mètodes i es veura que, fins ara, la major part dels mètodes han estat ideats per a l'adquisició de coneixements del domini i pocs esforços s'han realitzat per definir mètodes d'adquisició de coneixements estratègics.

1.3.1 Mètodes basats en l'enginyer de coneixements

Quan l'enginyer de coneixements és l'encarregat d'explicitar els coneixements de l'expert i implementar-lo en un computador, el mètode que més s'ha utilitzat és el que s'anomena contrucció ràpida de prototipus. Consisteix a contruir un mòdel inicial senzill, implementar-lo i testar-lo en l'ordinador, i en successives etapes anar refinant els coneixements d'aquest prototipus fins que l'expert estigui satisfet del seu funcionament. És evident que aquest mètode comporta molts problemes de manteniment del SBC. Existeixen eines i entorns que ajuden l'enginyer de coneixements a construir i a mantenir una base de coneixements,

com editors, interfícies gràfiques, etc (per exemple: KADS Power Tools [22], KEE [59], Knowledge Craft [121], etc). En aquesta línia, la metodologia KADS per a l'adquisició de coneixements [168, 22], una metodologia alternativa a la construcció ràpida de prototipus, proposa fer una anàlisi conceptual dels coneixements que lingüísticament expressa un expert a partir de la qual construir un model epistemològic. Aquest model serveix de pont entre el llenguatge de l'expert i el llenguatge d'implementació d'un sistema. El model, segons la metodologia KADS, ha de distingir quatre tipus fonamentals de coneixements: de domini, d'inferència, de tasca i estratègic. Desgraciadament, al llarg d'aquest projecte no s'ha aprofundit en l'estudi de formalismes per descriure els coneixements estratègics.

1.3.2 Eines d'explicitació de coneixements

Es tracta d'eines basades en models o en tasques específiques. Les eines d'explicitació basades en models dirigeixen l'adquisició de coneixements per models de representació que resulten útils per portar a terme una tasca específica. En aquest sentit, s'han desenvolupat algunes eines a partir de nuclis de sistemes experts, que ha donat lloc a la tècnica de nuclearització (*shellification* [22]). Consisteix a agafar un sistema expert en funcionament i extreure-li els coneixements de domini. El producte resultant s'instancia sobre un nou domini mitjançant les adequades adaptacions. Aquest és el cas d'EMYCIN [159], que prové de la nuclearització de MYCIN. Les estratègies de raonament en aquests models, però, són fixes. És a dir, per resoldre un problema sempre se segueix el mateix pla.

Entre el conjunt d'eines d'explicitació de coneixements s'ha de destacar ASK. ASK és un sistema que ajuda a l'obtenció de noves estratègies dins del procés de resolució de problemes d'un sistema expert. ASK resulta útil per adaptar coneixements estratègics de caràcter general ("*quan s'ha de cercar una hipòtesi, preguntar símptomes*", "*executar cada acció només una vegada*", "*concentrar-se en dades que seleccionin hipòtesis*", etc.) a un domini concret. ASK [63] es basa en la interacció amb l'expert i utilitza tècniques d'aprenentatge automàtic per inferir característiques de control dels coneixements del domini (veure capítol 6 per a més informació).

1.3.3 Aprenentatge automàtic

L'automatització total o parcial de l'adquisició de coneixements es pot aconseguir per tècniques d'aprenentatge automàtic [2, 18]. Els avantatges d'automatitzar l'adquisició de plans són moltes, entre els quals destaquem als següents:

- L'expertesa no és universal. Els experiments de [143, 151] ens diuen que existeix més d'una classe d'expertesa dins d'una disciplina i que no existeix una organització universalment millor dels coneixements, però sí que algunes organitzacions són millors que d'altres [143, 117]. Les causes d'aquesta multiplicitat d'organitzacions són la capacitat i habilitats de l'estudiant durant l'aprenentatge (*learning situations*), la manera en què l'entorn d'aprenentatge s'estructura³ i, finalment, els objectius de l'aprenent durant el procés d'aprenentatge. Així doncs, un individu ajusta els seus coneixements d'acord amb les tasques per les quals típicament els aplica. Per aquesta

³L'ordre dels exemples així com la seva complexitat afecta l'aprenentatge [25, 35]. En particular, una etapa d'aprenentatge organitzada, conjuntament amb una etapa de pràctica complexa, comporten una resolució de problemes eficient [25, 163].

raó, l'expertesa no és universal, sinó un tipus d'adaptació útil per a una classe de situacions.

- Les tècniques d'aprenentatge deslliuren l'expert humà d'una bona part de la tasca de transmetre els seus coneixements en la construcció de SBC. Així, l'expert pot limitar-se quasi pràcticament a la validació dels coneixements que s'aprenen.
- L'aprenentatge redueix l'esforç que ha de fer l'enginyer de coneixements per posar a punt un sistema. En efecte, sense aprenentatge automàtic l'eficiència del SBC depèn de la destresa que tingui l'enginyer de coneixements a realitzar una tasca de refinament (*tuning*) amb el SBC per assegurar la seqüència apropiada d'accions en la resolució d'un problema. Segons un estudi empíric realitzat en la Universitat Carnegie Mellon [26, 27], l'enginyer dedica d'entre un 30% a un 60% del seu esforç en el disseny de l'aplicació per realitzar aquest refinament.
- L'aprenentatge facilita el manteniment i l'aplicabilitat dels sistemes. Les condicions en què es desenvolupa una aplicació poden ser vàlides en unes circumstàncies particulars. El canvi d'alguna de les circumstàncies repercuteix en els coneixements del sistema, que necessiten ser actualitzats. Per exemple, una aplicació de diagnòstic de pneumònies pot ser vàlida per un determinat àmbit geogràfic, de manera que si es vol utilitzar el sistema en una altra país o regió alguns dels coneixements que determinen el clima han de ser corregits.

Els mètodes d'aprenentatge automàtic es poden classificar en dos grups segons el mètode fonamental d'inferència que utilitzen [79]: aprenentatge per inducció i aprenentatge per deducció. Hi ha un tercer conjunt de mètodes que s'escapen a aquesta classificació: els mètodes d'aprenentatge per analogies [76, 56]. El raonament analògic ocupa una posició principal en la recerca en IA tant en qualitat de mètode d'adquisició de coneixements com en qualitat de mètode de resolució de problemes. En la secció 1.4 farem una petita introducció al raonament analògic i a un apropament computacional en particular: el raonament basat en casos.

Mètodes inductius

Els mètodes inductius construeixen una definició legal d'un concepte a partir d'un conjunt d'exemples (positius i/o negatius) d'aquest concepte. La definició legal ha d'incloure els exemples positius com a instàncies del concepte, i no ha d'incloure els exemples negatius. Per a cada concepte existeix el que s'anomena un espai de versions [111], on el límit superior és el conjunt de definicions legals més general del concepte i el límit inferior és el conjunt de definicions legals més específiques d'entre totes les representacions legals admissibles del concepte. Una de les característiques de l'aprenentatge inductiu és la utilització de biaixos (*biases* [154, 155]). Donat que en general hi ha moltes solucions correctes a la inducció (moltes definicions legals), un sistema en genera només una o algunes d'entre aquestes; s'anomena biaix a les preferències que aplica un sistema d'aprenentatge inductiu per a l'elecció de les definicions legals generades dins les definicions legals admissibles [156]. Un problema important en aquests mètodes és que són susceptibles al soroll (i.e. possibles dades errònies) que pugui haver en els exemples. En aquest cas, el biaix és important per assegurar una bona predictivitat. En els mètodes inductius, l'aprenentatge pot fer-se en un sol pas (és a dir, s'obté la descripció dels conceptes a partir de tot els exemples), o incrementalment (la disponibilitat d'un nou exemple pot modificar la descripció del concepte).

Bàsicament existeixen dues famílies de mètodes inductius: 1) algorismes d'arbres de

decisió inductius que es basen en els elements discriminants que separen la definició d'un concepte envers d'altres, 2) l'aprenentatge basat en similituds que es basa l'abstracció de les característiques rellevants dels objectes. Els mètodes d'arbre de decisió inductius (*induction decision trees*, *IDT* [127, 128, 101, 157]) consisteixen a generar un arbre de decisió per reconèixer un conjunt d'exemples rellevants a una tasca de classificació. Els exemples estan configurats com una col·lecció d'atributs. La construcció de l'arbre es comença per l'arrel i acaba en les fulles (*top-down*), de manera que en les fulles de l'arbre es troben les classes i la resta de nodes tenen un atribut que determina els exemples que se situen a sota del node. Per construir l'arbre s'utilitza una mesura sobre la informació que els atributs proporcionen perquè un valor determini una classe. Pot existir més d'una representació possible per a un conjunt d'exemples determinat, i el millor arbre serà aquell que sigui més predictiu envers un concepte no inclòs en el conjunt d'aprenentatge, o senzillament, l'arbre més simple (la profunditat d'una branca determina el nombre de preguntes necessàries per classificar un objecte). Els algorismes d'arbres de decisió són el mètode amb més tradició com a tècnica d'aprenentatge automàtic, però no existeix cap algorisme aplicat a l'adquisició de plans.

L'aprenentatge basat en similituds analitza un conjunt d'exemples positius i negatius i obté una descripció d'un concepte. L'anàlisi es basa en característiques comunes entre les instàncies positives i en la cerca de justificacions per a les característiques que no són comunes. Cada nova instància positiva pot resultar en una descripció del concepte més general. Les instàncies negatives o contraexemples permeten validar si la descripció obtinguda és massa general, i en qualsevol cas especialitzar-la. El procés d'inducció està recolzat per coneixements de domini per tal de limitar el conjunt de possibles definicions (asserccions) que es puguin construir. La teoria del domini proporciona també relacions entre les característiques de les instàncies, la rellevància d'una característica envers al concepte, etc. [78, 105]. L'aplicació de l'aprenentatge basat en similituds ha estat aplicat a l'adquisició de coneixements de control en LEX [111, 112]. LEX és un sistema que aprèn heurístiques per millorar el procés de resolució d'integrals. Aquestes heurístiques actuen com a regles de control de cerca sobre el domini. El procés està basat en traces (seqüències de passos) del procés de resolució d'una integral, i és incremental. És a dir, en cada nova experiència el sistema és capaç de refinar les seves heurístiques.

Mètodes deductius

Els mètodes deductius es basen en la deducció de nous conceptes o cadenes deductives a partir de l'anàlisi dels exemples i mitjançant una teoria del domini. Els mètodes deductius tenen com a finalitat fer operacional un conjunt de conceptes o fer més eficient el funcionament d'un sistema. Distingim l'aprenentatge basat en explicacions i la compactació.

En l'aprenentatge basat en explicacions (EBL) es pretén operacionalitzar una teoria a partir d'instàncies (exemples) de conceptes definits en ella. És a dir, donat un exemple es genera una explicació que és una demostració que l'exemple és una instància d'un concepte determinat. El mètode basat en explicacions ha estat utilitzat per Minton en el sistema PRODIGY (PRODIGY/EBL) en l'aprenentatge de coneixements de control de cerca [106]. PRODIGY/EBL obté una explicació d'una traça del procés de resolució on es justifica la selecció de cadascuna de les accions del sistema. A partir de les explicacions s'aprenen regles de control (de selecció i preferència d'accions) utilitzant, evidentment, el mètode d'aprenentatge basat en explicacions. Les regles s'incorporen al SBC depenent d'una estimació del grau d'utilitat calculat per a cadascuna d'elles, de manera que la cerca a cegues es va

convertint en cerca guiada per l'experiència.

Els mètodes de compactació (compilació, *chunking*, macro-operadors) consisteixen a integrar en un sol pas seqüències de subobjectius desenvolupades per assolir un objectiu més general durant el procés de resolució d'un problema. SOAR [91, 130], per exemple, crea macrooperadors a partir d'un objectiu, la posició o estat inicial, i la seqüència d'operadors utilitzada en una experiència anterior. El sistema CONCLAVE [156], que distingeix dos nivells de coneixements de domini, un heurístic o superficial i un causal, quan no pot resoldre un problema amb els coneixements superficials o heurístics recorre al raonament causal; un cop obté la solució, compacta o compila els coneixements causals aplicats en la resolució del problema en regles heurístiques a utilitzar en el procés de *problem solving* en experiències futures. La compactació, doncs, millora el rendiment del procés de resolució de problemes, ja que evita repetir una descomposició d'objectius en subobjectius que ja ha estat realitzada en una experiència anterior. Essencialment, la compactació redueix la profunditat en l'espai de cerca de solucions [130].

1.3.4 Sistemes híbrids

Els sistemes híbrids d'adquisició de coneixements combinen diferents mètodes, bé per adquirir amb cada mètode un tipus de coneixement (per exemple, un mètode per adquirir coneixements de domini i un altre per adquirir coneixements estratègics), bé per fer l'adquisició de coneixements més precisa. ASK [63] utilitza l'enginyer de coneixements per definir les característiques de control que permeten avaluar l'estat actual del sistema en la resolució d'un problema. Amb aquestes característiques el sistema ASK interacciona amb l'expert per tal d'adquirir coneixements de planificació. Durant l'explicitació de coneixements pot succeir que l'expert assigni a dues situacions plans diferents, i que aquestes situacions no siguin distingibles amb les característiques de control de què el sistema disposa. Aleshores ASK invoca a un mètode d'aprenentatge basat en similituds per aprendre característiques de control noves capaces de distingir les dues situacions. ML-SMART combina EBL i SBL per aprendre conceptes en una teoria del domini incompleta o incorrecta [17]. Lavrac [96] proposa una arquitectura metanivell per desenvolupar un sistema híbrid d'aprenentatge basat en la integració de dos sistemes d'aprenentatge inductius. Cada sistema d'aprenentatge té assignat un conjunt de propietats (com per exemple, si és útil quan la informació és incompleta, o quan hi ha dades errònies, etc) que conformen el biaix de l'algorisme d'aprenentatge. Segons una anàlisi de l'exemple a partir del qual s'ha de realitzar l'adquisició de coneixements es decideix utilitzar un sistema o un altre.



Finalitzem aquesta introducció als mètodes d'adquisició de coneixements remarcant els avantatges que presenta l'automatització dels mètodes d'aprenentatge automàtic: deslliuren l'expert humà d'una bona part de la construcció de la base de coneixements, redueixen l'esforç de l'enginyer de coneixements per posar a punt un sistema, i faciliten el manteniment i l'aplicabilitat del sistema. Les tècniques d'aprenentatge automàtic presenten, però, problemes en l'escalament (*scaling-up*) dels mètodes desenvolupats per aplicacions menudes a aplicacions del món real. Els resultats que amb aplicacions de laboratori (*toy problems*) poden semblar òptims resulten pobres en grans bases de coneixements. També cal definir mesures d'avaluació de la utilitat del coneixements que s'aprenen [106, 97, 33] per evitar un creixement descontrolat

de la base de coneixements. Per exemple, en el sistema SAGE [93, 92] s'aprenen en diferents experiments un nombre de 73, 67 i 18 heurístiques, de les quals només 7, 1 i 4 respectivament resulten útils.

Donat els diferents tipus de coneixements que requereix un SBC (per exemple de domini i de planificació), s'ha de pensar en mètodes diferenciats per a l'adquisició de cadascun d'ells. En aquesta línia es desenvolupen sistemes híbrids. Malauradament, l'aprenentatge automàtic s'ha enfocat en l'aprenentatge de conceptes per a tasques de classificació i no gaire en l'aprenentatge de plans. Només els sistemes ASK [63], PRODIGY/EBL [106], SAGE [93], CHEF [68] i SMART [163] es poden considerar com a precedents importants en l'adquisició de coneixements estratègics amb tècniques d'aprenentatge automàtic (vegeu capítol 6).

1.4 Raonament analògic, raonament basat en casos i planificació

El raonament analògic és una alternativa a altres metodologies per al desenvolupament de SBC com són els sistemes de producció, els sistemes deductius i els sistemes d'inducció simbòlica. El raonament analògic es defineix com el procés d'inferir que una conclusió Q sobre una situació o objecte T és certa pel fet que T comparteix una propietat o un conjunt de propietats P amb una altre situació/objecte S que té la propietat Q . P s'anomena la similitud entre S i T ; i Q s'anomena la projecció de S en T [47]. El raonament analògic es pot esquematitzar com

$$\frac{P(S) \wedge Q(S)}{P(T)} \quad \text{analògic} \\ Q(T)$$

Cal remarcar que l'analogia no és un procés deductiu basat en la sintaxi (les conclusions no es deriven només sintàcticament de les premisses).

El procés de raonament analògic consta de les etapes següents [76, 64]:

1. Reconeixement d'un objecte anàleg S donada una descripció de l'objecte T , etapa també coneguda com a recuperació (*retrieval*).
2. Elaboració d'una correspondència (*mapping*) analògica entre el domini de S i el de T , incloent-hi un conjunt d'inferències analògiques. Aquesta etapa també es coneix per adaptació.
3. Avaluació del *mapping* i les inferències en el domini de T incloent-hi justificacions, explicacions o extensions del *mapping*.
4. Consolidació o aprenentatge del resultat de l'analogia de manera que pugui aprofitar-se per a futures situacions.

Existeixen diferents aproximacions del raonament analògic en IA, com la transformació analògica (ARIES [29, 64]), la derivació analògica [30], el raonament basat en casos [80, 65, 12], etc. No tots els mètodes desenvolupen l'etapa de consolidació o aprenentatge. Entre tots els mètodes d'aprenentatge per analogia, el raonament basat en casos fa de l'aprenentatge el centre de tot el procés analògic, raó per la qual esdevé una tècnica molt útil per a l'adquisició de coneixements. De fet, el RBC no és un mètode únic, sinó una família de mètodes cadascun dels quals amb una instanciació diferent de les etapes que conformen el raonament analògic (recuperació, adaptació, avaluació i aprenentatge [74, 148]). El raonament basat en casos significa resoldre problemes a partir de precedents, adaptant-ne solucions antigues per resoldre problemes nous, o recuperant-ne casos anteriors per il·luminar aspectes de la situació

actual [74, 148]. Una particularitat del RBC és el tipus d'analogia: mentre que el raonament analògic sol ser interdomini, és a dir es resolen problemes en un domini fent ús de problemes (i llur solució) de dominis diferents, el RBC acostuma a fer una analogia intradomini, és a dir entre casos del mateix domini [140, 1].

L'element bàsic en el RBC és el cas. El cas es pot representar per diferents característiques, com la descripció del problema (els objectius o les restriccions a satisfer), les solucions (la solució propiament dita, o acompanyada d'un conjunt d'anotacions, justificacions, etc.), les dades inicials, i fins i tot una traça que indica com s'ha dut a terme la resolució del cas. Les característiques dels casos són la clau per la construcció d'índexs que facin flexible la futura recuperació dels casos de la memòria.

En el RBC és fonamental l'organització dels casos precedents en la memòria per poder ser utilitzats en la resolució de problemes nous. Quan es construeix un mecanisme de raonament basat en casos s'ha de pensar en una organització dinàmica de la memòria, ja que la incorporació d'un nou cas en memòria pot influir en la disposició dels casos precedents. Entre les diferents organitzacions de memòria podem trobar una organització plana on els casos són emmagatzemats seqüencialment en una llista o en un fitxer, o una jeràrquica on els casos estan organitzats d'acord amb les seves característiques, de manera que cada node correspon a una característica compartida per un conjunt de casos (xarxes de discriminació, *discrimination nets*). L'organització de la memòria és fonamental per dur a terme el raonament basat en casos: recuperar, adaptar, avaluar i, finalment, aprendre de casos.

Recuperació. La clau del raonament analògic, i consegüentment la del RBC, rau a determinar quan dos casos (C^M en memòria i C el cas actual) són semblants [6, 9, 150, 133]. Entre dos casos existeixen diferents tipus de similituds: estructurals, semàntiques, organitzatives i fins i tot pragmàtiques [172]. Per determinar si un cas és semblant a un altre es defineix una funció de comparació (*matching*) on s'acaren les característiques d'un cas amb les de l'altre cas. El *matching* és exacte o parcial. Exacte significa que ambdós casos són idèntics. Parcial vol dir que existeix una semblança entre els casos determinada per un grau de similitud. En el *matching* es poden utilitzar un conjunt d'heurístiques que ajudin a determinar quines són les característiques més rellevants, que indiquin com tractar una característica que és present en un cas però no ho és en el cas amb què el volem comparar, etc. Per exemple, per a la determinació d'una característica f present a C i absent a C^M l'heurística més utilitzada és la de cercar en C^M una característica f' més abstracta que f i aleshores comparar f amb f' . Alguns sistemes utilitzen mètodes més sofisticats per resoldre el problema de la característica absent com, per exemple, un model causal que la dedueixi usant la resta de les dades del cas (CASEY [87]).

En moltes ocasions existeix no un sinó un conjunt de casos en memòria versemblants ($C_1^M, C_2^M, \dots, C_n^M$) que fan un *matching* parcial amb el cas actual C . També en aquesta circumstància la utilització d'un conjunt d'heurístiques i/o coneixements del domini permet decidir quin és el millor cas de memòria respecte a C . Per exemple, es pot escollir entre valorar les característiques que apareixen sempre conjuntament, fer un *trade-off* (mitjana) entre el nombre de característiques presents i el nombre d'absents, etc.

Adaptació L'elaboració o adaptació consisteix a acomodar la solució del cas C^M recobrat de la memòria per obtenir la solució al cas C . En l'adaptació és necessari tenir en compte les diferències entre C^M i C (per exemple si hi ha característiques de C^M absents en C , o

viceversa). Els mètodes d'adaptació són nombrosos: la reinstanciació o la substitució d'una característica per un altre (CHEF [68]); la cerca local de característiques en una xarxa de discriminació; el mètode dels *crítics* on es defineixen un conjunt especial de regles d'adaptació per tractar característiques que interaccionen (CHEF [68], CASEY [87]); el mètode anomenat *derivational replay* [163], on es recupera el procés amb el qual es possible calcular una característica absent; etc. (En [74] es pot trobar una explicació detallada de cadascun dels mètodes.)

Avaluació. L'etapa d'avaluació determina si la solució del cas elaborada en l'etapa d'adaptació és plausible [64]. Malgrat la importància de l'avaluació, molts pocs sistemes basats en casos van més enllà de consultar un oracle per determinar la validesa de la solució. Si la tasca del sistema basat en casos és la planificació, sovint l'avaluació dels plans es realitza mitjançant l'execució dels plans en el món i la comparació del resultat. Ocasionalment s'ha emprat un simulador (CHEF [68]).

Aprenentatge. La consolidació en el RBC consisteix a emmagatzemar el nou cas C resultat del raonament analògic. És en aquest procés d'emmagatzemament on sorgeix l'oportunitat de reorganitzar la memòria, i en conseqüència d'aprendre a partir de la nova experiència. Si la solució adaptada ha estat correcta (aprenentatge a partir de l'èxit), el nou cas proporcionarà nous coneixements al sistema. Si ha fracassat (aprenentatge a partir d'errors), s'hauran de reorganitzar els casos en memòria per tal que no es torni a produir un fracàs. No tots els sistemes d'aprenentatge a partir de casos consideren ambdues oportunitats en l'aprenentatge.

Tot i que el RBC és una tècnica recent, està tenint un gran ressò en la comunitat d'IA perquè presenta els avantatges següents:

1. El RBC com a tècnica d'adquisició de coneixements

- (a) deslliura l'expert humà de facilitar dades d'entrenament pròpiament dites, característica de les tècniques d'aprenentatge tradicionals, i n'hi ha prou que el sistema observi casos resolts, eventualment, per un mestre, i
- (b) facilita la interacció persona-màquina, ja que la comunicació es pot realitzar mitjançant casos concrets i no trossos parcials de coneixements, com ho són les regles (l'expert té més facilitat a utilitzar representacions basades en episodis que en regles, donat la quantitat d'esforç que envolta l'abstracció d'una regla general envers un cas particular [25]).

2. El RBC com a mecanisme de resolució de problemes permet aprendre de la pròpia experiència del sistema, de manera que un sistema inicialment no-expert esdevingui expert.

3. El RBC permet treballar en un domini de problemes de difícil estructuració i representació [14, 74, 124], característiques que normalment presenten les aplicacions reals d'intel·ligència artificial. Tot afrontant des del primer moment el desenvolupament d'una base de coneixements difícil, s'evita el problema que presenta l'escalament en els mètodes d'aprenentatge.

1.4.1 La planificació basada en casos

El raonament basat en casos aplicat a la planificació es coneix com a *planificació basada en casos* (*case-based planning*). La planificació basada en casos (PBC) es caracteritza pel fet que els plans es construeixen conforme als coneixements que té el planificador sobre plans emprats en experiències passades (casos) [68]. La construcció d'un pla no és més que la cerca d'un pla en la memòria del planificador on tots els objectius són satisfets. El planificador fa un ús extensiu de la memòria i, en conseqüència, l'organització de la memòria i la indexació juguen un paper fonamental, com en qualsevol sistema basat en casos.

La utilització del RBC per resoldre problemes de planificació compta amb precedents importants en els sistemes CHEF [68] i SMART [163]. CHEF disposa d'una memòria de plans, una memòria de modificacions, una memòria d'errors i una de reparacions a partir de les quals troba nous plans per a nous problemes. CHEF introdueix els conceptes de planificar a partir de situacions, reusar plans i aprendre plans incrementalment. El sistema SMART [163] reutilitza plans executats amb anterioritat per planificar la trajectòria d'un robot mitjançant la reexecució analògica (*analogical replay*), però només d'una manera alternativa a un procés de raonament tradicional en la construcció de plans. Aquest dos sistemes estan presentats i discutits amb més detall en el capítol 6.

1.5 Visió de conjunt

Durant aquest capítol hem recollit diferents aspectes sobre l'estat actual de la planificació que podem resumir en els punts següents:

1. Els planificadors en aplicacions reals han de manipular informació incompleta i incerta. La utilització d'una planificació reactiva permet afrontar amb èxit el problema de la informació incompleta i és, per tant, una tècnica a tenir present en el desenvolupament de qualsevol planificador.
2. Els coneixements estratègics són indispensables per dur a terme una planificació flexible, adequant els recursos disponibles, etc. En aquest sentit, les arquitectures metanivell permeten integrar els coneixements estratègics amb altres tipus de coneixements (e.g. coneixements de domini) necessaris per a la resolució de problemes.
3. Els coneixements estratègics requereixen d'una etapa d'adquisició de coneixements. L'adquisició de coneixements és encara el coll d'ampolla de la IA en el desenvolupament de sistemes bastants en coneixements. La utilització de mètodes d'aprenentatge automàtic com a mètodes d'adquisició de coneixements és necessària per agilitzar aquesta etapa del desenvolupament d'un SBC.
4. El raonament analògic es presenta com una alternativa per a la incorporació de coneixements en un sistema alhora que com a mètode de resolució de problemes. En particular, el raonament basat en casos ha demostrat ser una interfície còmoda en l'adquisició de coneixements, una tècnica de resolució de problemes que millora a partir de l'experiència i, el que és molt important, evita el problema de l'escalament que presenten altres tècniques d'aprenentatge automàtic.

Cal remarcar que la construcció de sistemes amb arquitectura metanivell està fortament consolidada i avui en dia no es dissenya un planificador sense aquest tipus d'arquitectura. També advertim que s'estan seguint, de manera independent, dues línies principals d'investigació en la planificació. Per una banda, el desenvolupament de planificadors reactius

capaços de tractar el problema de la informació incompleta i incerta. Per l'altra banda, el desenvolupament de mètodes d'aprenentatge automàtic que afrontin el problema de l'adquisició de coneixements estratègics per a la planificació. En el capítol que segueix es presenta un sistema que pretén aproximar ambdós esforços d'investigació: la planificació reactiva i l'automatització de l'adquisició de coneixements.

Capítol 2

L'aproximació de BOLERO: planificació basada en casos

En aquest capítol presentem una nova aproximació a la planificació que neix de la idea de recollir els esforços fets en els darrers anys en planificació reactiva i en l'adquisició de coneixements, i que es consolida en el desenvolupament d'un nou planificador: BOLERO. En el disseny del nou planificador s'han plantejat els problemes següents:

1. Tractar el problema de la informació incompleta i incerta
2. Millorar l'eficiència en la resolució de problemes d'un SBC que no disposi de coneixements estratègics (*speed-up learning* [153]).
3. Disminuir el cost de l'adquisició de coneixements estratègics (de planificació) al màxim mitjançant tècniques d'aprenentatge automàtic
4. Planificar en entorns reals, en particular en aplicacions mèdiques.

Cadascuna d'aquestes qüestions ha estat soltada amb les consideracions següents en el disseny de BOLERO:

1. Planificació reactiva mitjançant la construcció de plans adequats a la informació coneguda en cada moment, i alternació de la planificació i l'execució dels plans
2. Integració de BOLERO amb un sistema basat en coneixements que no disposi de coneixements estratègics en una arquitectura metanivell. Amb la integració d'ambdós sistemes s'aconsegueix millorar l'eficiència del SBC alhora que alternar la planificació i l'execució tot facilitant la planificació reactiva (figura 2.1). És a dir, ambdós sistemes romanen independents alhora que col·laboren en la resolució de problemes.
3. Dos mètodes d'aprenentatge diferents per tal d'aprendre coneixements estratègics: 1) aprenentatge per observació que permeti aprendre d'un oracle, i 2) aprenentatge per experiència a partir dels reeiximents i dels fracassos en la generació de plans.
4. Planificació basada en casos, com a arquitectura del sistema en la qual s'integrin totes les tècniques, tant de planificació com d'aprenentatge. El raonament basat en casos, i aleshores particularment, la planificació basada en casos, s'ha desvetllat com una arquitectura útil per a aplicacions reals.

Entre tots els requeriments de disseny de BOLERO, hi ha dues parts diferenciades: 1) BOLERO com a planificador basat en casos on s'esdevé la generació (recuperació i adaptació), l'avaluació i l'aprenentatge de plans, i 2) BOLERO com a integrant d'una arquitectura metanivell on realment té lloc la resolució de problemes. Així, en el supòsit de disposar d'un SBC (que a partir d'ara anomenem SBC^B) capaç de resoldre un problema correctament

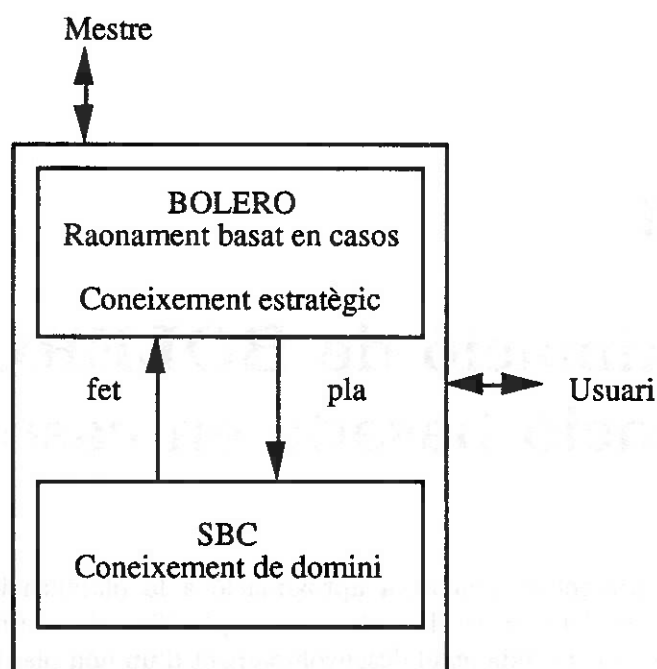


Figura 2.1: Sistema integrat resultat de la iteració entre BOLERO i un SBC.

per una cerca exhaustiva de totes les solucions possibles, BOLERO pot ser aplicat per aprendre plans amb els quals millorar l'eficiència del SBC. La manera de fer-ho és integrar ambdós sistemes en una arquitectura metanivell, que anomenem BOLERO-SBC. En aquesta arquitectura BOLERO manca de qualsevol coneixement sobre planificació. Per aquesta raó és necessari realitzar una etapa preliminar d'aprenentatge de plans per observació en la qual BOLERO aprèn plans (casos) que li proporciona un mestre. Després de l'aprenentatge per observació, BOLERO està preparat per construir plans per a resoldre problemes nous. La construcció d'un pla està imbricada amb l'execució en BOLERO-SBC on BOLERO i *SBC^B* interaccionen mitjançant una interfície. Un cop resol un problema, BOLERO té una experiència nova en la construcció de plans, és a dir, un cas nou. Una avaluació del pla del cas nou indica un reeiximent o fracàs del pla. Del resultat de l'avaluació del pla BOLERO pot aprendre per experiència.

Continuem el capítol amb una introducció a l'arquitectura metanivell BOLERO-SBC (secció 2.1) que ens permetrà mostrar com té lloc la imbricació de la planificació i l'execució dels plans. Després (secció 2.2) prosseguim explicant com és possible aprendre coneixements estratègics (i.e. plans) per l'observació de casos proporcionats per un mestre. A continuació (secció 2.3) es mostra com es poden reusar plans de casos precedents per resoldre problemes nous, que es constitueix amb les etapes de recuperació i adaptació de la planificació basada en casos. Després, en la secció 2.4, s'introdueix la problemàtica i solucions adoptades per l'avaluació de plans. I a continuació es mostra com, amb la informació obtinguda en l'avaluació de plans, és possible aprendre per l'experiència a partir tant del reeiximent dels plans com de plans imperfectes (secció 2.5). Finalitzem en la darrera secció (secció 2.6) ressenyant com tots els mètodes interaccionen per aconseguir la planificació reactiva basada en casos.

Cal remarcar que el propòsit d'aquest capítol no és proporcionar una visió de detall de cadascun dels mètodes desenvolupats (aquest és el propòsit dels capítols 3 i 4), sinó mostrar com és possible desenvolupar mètodes d'aprenentatge i generació de plans mitjançant l'ús del raonament basat en casos, i com la interacció d'un planificador basat en casos amb un altre sistema basat en coneixements pot conduir a una planificació reactiva. L'aproximació de la planificació basada en casos que es fa en aquest capítol és, doncs, una visió nova, possiblement esbiaixada pel tipus d'aplicació en el qual ha estat adreçada, però que justifica el desenvolupament d'un nou sistema. En aquest capítol es troben amb freqüència referències a seccions de capítols posteriors on s'amplien i es precisen els conceptes i mètodes que aquí s'introdueixen.

2.1 L'arquitectura metanivell BOLERO-SBC

El sistema BOLERO-SBC integrat per BOLERO i un SBC és un sistema metanivell on BOLERO juga el paper de metanivell i SBC^B juga el paper del nivell objecte (vegeu secció 1.2.1). SBC^B té els coneixements d'un domini particular per resoldre problemes i BOLERO té els coneixements estratègics específics per a aquest domini. Com en qualsevol arquitectura metanivell (vegeu secció 1.2.1) el metanivell, és a dir, BOLERO, posseeix un model parcial de SBC^B . El model es basa en la informació que SBC^B dedueix o adquireix de l'usuari. BOLERO construeix plans que modifiquen l'estat de SBC^B ; SBC^B és capaç d'interpretar i executar els plans construïts per BOLERO. Així mateix, BOLERO és capaç d'engegar i aturar l'execució de SBC^B . Ambdós sistemes estan connectats causalment i, per aquesta raó, com es veurà detalladament en el capítol 3, el sistema BOLERO-SBC és un sistema reflexiu [102, 60].

Els plans que construeix BOLERO són executats immediatament per SBC^B . BOLERO pot interrompre en qualsevol moment un pla que SBC^B estigui executant, i imposar-li un nou pla d'acció. D'aquesta manera s'aconsegueix imbricar la planificació i l'execució dels plans, element indispensable per aconseguir una planificació reactiva. En el moment de començar a resoldre un problema, BOLERO construeix un pla per defecte p a partir de la informació de què disposa. Amb aquest pla s'engega SBC^B , que comença a executar p . En el decurs de l'execució del pla, SBC^B pot conèixer nova informació de l'entorn (respostes de l'usuari, conclusions, etc.). La nova informació és aleshores coneguda per BOLERO, el qual pot decidir canviar de pla. Es realitzen tants canvis de pla fins que s'executa un pla p' amb el qual s'assoleix la solució d'un problema i BOLERO atura l'execució de SBC^B . D'aquesta manera, amb la creació immediata de plans, el sistema BOLERO-SBC reacciona als canvis de l'entorn.

Per tal de donar un exemple del comportament reactiu del sistema BOLERO-SBC are farem un seguit de suposicions¹. En primer lloc, suposem que SBC^B és un sistema basat en regles capaç de diagnosticar pneumònies. En segon lloc, suposem que un pla per SBC^B significa una seqüència d'objectius a assolir. La validació d'un objectiu es realitza, doncs, per l'encadenament enrera de les regles (*backward chaining*). Així, executar el pla $p=[bact-atip, pneumococ]$ és equivalent a assolir l'objectiu *bact-atip*, i a continuació l'objectiu *pneumococ*. Finalment, suposem que BOLERO té un model de SBC^B basat en fets, i que un fet és una parella (*atribut, valor*), com per exemple (*edat, 79*). Els fets que SBC^B dedueix mitjançant les seves inferències o adquireix de l'usuari formen part del model. Aleshores, si un usuari

¹En tota la tesi sempre que posem exemples sobre SBC^B assumirem les mateixes suposicions.

vol diagnosticar un malalt, del qual coneix unes dades, es produeix la seqüència següent de canvis en el procés de raonament en BOLERO-SBC:

- BOLERO genera el pla inicial $p_0 = [\text{recollida-dades}]$.
- SBC^B inicia l'execució amb l'objectiu *recollida-dades*. Recollida-dades és un objectiu de SBC^B que consisteix a emplenar una fitxa amb l'historial clínic del pacient.
- Durant el seu raonament, SBC^B pregunta a l'usuari l'edat del pacient, i obté el fet (edat, 79).
- La nova informació (edat, 79) és coneguda per BOLERO, però no resulta una informació rellevant i BOLERO insisteix en el pla [recollida-dades].
- El SBC continua executant *recollida-dades*.
- SBC^B continua adquirint dades de l'usuari ((clínica, *pneumònia extrahospitalària*), (estat-del-malalt, *moderadament-greu*), (sexe, *dona*), (internat-en-un-asil, *desconegut*)), però BOLERO no canvia de pla.
- Fins que en un moment donat, SBC^B aquireix de l'usuari la dada (antecedents, *diabètic*).
- En conèixer (antecedents, *diabètic*), i tenint en compte la informació que ha rebut des de l'inici de l'execució (i.e. (edat, 79), (clínica, *pneumònia extrahospitalària*), estat-del-malalt, *moderadament-greu*), (sexe, *dona*), (internat-en-un-asil, *desconegut*)), BOLERO construeix un pla nou: [bact-atip, pneumococ, legionel·la, s-piògen].
- L'interpret de SBC^B té una nova llista d'objectius, [bact-atip, pneumococ, legionel·la, s-piògen], i comença a validar bact-atip.
- ...I així successivament fins que SBC^B ha conclòs els fets (pneumococ, *segur*), legionel·la, *moderadament-possible*), que constitueixen la diagnosi del pacient, i per tant la solució al problema. BOLERO decideix aleshores aturar el sistema SBC^B i es dona per conclosa la diagnosi del pacient en BOLERO-SBC.

Com fa palès l'exemple, la interacció entre BOLERO i SBC^B permet a BOLERO testar amb una certa freqüència l'entorn (mitjançant SBC^B) i construir plans adequats als canvis que s'han produït. BOLERO és capaç de generar un pla amb informació incompleta, tal i com es mostrarà en la secció 2.3, i dins de l'arquitectura BOLERO-SBC BOLERO pot generar tants plans com canvis hi ha a l'entorn. D'aquesta manera s'aconsegueix una planificació reactiva.

Un cop instal·lat BOLERO dins de l'arquitectura metanivell BOLERO-SBC, passem a introduir en les seccions següents els mètodes desenvolupats en BOLERO com a planificador basat en casos.

2.2 Aprenentatge de plans per observació

La majoria de sistemes basats en casos assumeixen que les experiències anteriors són en memòria o bé són conseqüència de l'aprenentatge per l'experiència pròpia del sistema. En BOLERO distingim una etapa inicial o d'entrenament que hem anomenat d'aprenentatge per observació, on el sistema organitza, en un procés incremental, un conjunt de casos en memòria. En aquesta etapa BOLERO no intenta resoldre problemes, sinó que simplement observa com

Pacient b06v1:	Pacient g01v1
(a) Dades inicials	
(clínica, <i>pneumònia extrahospitalària</i>)	(clínica, <i>pneumònia extrahospitalària</i>)
(sexe, <i>home</i>)	(sexe, <i>home</i>)
(forma-d'instauració, <i>brusca</i>)	(forma-d'instauració, <i>brusca</i>)
(b) Dades recollides en l'execució	
(estat-del-malalt, <i>mod-greu</i>)	(estat-del-malalt, <i>lleu</i>)
(edat, <i>85</i>)	(edat, <i>27</i>)
(internat-en-un-asil, <i>desconegut</i>)	
(dades-RX, <i>condensació-vmaldef, confluents</i>)	(dades-RX, <i>condensació-vmaldef, p-intersticial</i>)
(<i>pneumònia-bacteriana, molt-possible</i>)	(<i>pneumònia-atípica, molt-possible</i>)
(c) Pla final	
[recollida-dades bact-atip pneumococ enterobacteriàcies legionella anaerobi tuberculosi estafilococ s-pyogenes hemofilus meningococ brahnamella]	[recollida-dades bact-atip micoplasma virus clamídia]
(d) Diagnòstics	
(<i>pneumònia-pneumocòcica, possible</i>)	(<i>pneumònia-micoplasma, mod-possible</i>)
(<i>pneumònia-hemòfilus, lleug-possible</i>)	(<i>pneumònia-vírica, força-possible</i>)
	(<i>pneumònia-clamídia, mod-possible</i>)

Figura 2.2: Dades dels casos b06v1 i g01v1.

un mestre resol problemes. Aquest problemes resolts per un mestre esdevenen exemples o casos d'entrenament per a BOLERO, a partir dels quals el sistema aprèn plans des de zero (i.e. sense coneixements previs sobre plans adients a un domini). L'aprenentatge per observació harmonitza amb la creença que una etapa d'aprenentatge organitzada conjuntament amb una etapa de pràctica complexa comporten una resolució de problemes eficient [25, 163].

L'aprenentatge per observació consisteix a organitzar en memòria un conjunt de casos que són exemples de plans. La informació representada en els casos és, doncs, la clau de l'aprenentatge de plans per observació. En el cas ha d'especificar-se tant el pla que condueix a la solució d'un problema com les les situacions (i.e. conjunt de fets) o el context en què s'ha construït i executat el pla. Per exemple, suposem que tenim dos casos "b06v1" i "g01v1" pels quals un metge (mestre) ha diagnosticat les causes de les pneumònies de dos pacients. Inicialment el metge coneix només que les dades generals d'ambdós pacients, com mostra la figura 2.2a². Amb les dades inicials el metge es proposa el pla [recollida-dades], que consta de l'acció *recollida-dades*. Executar *recollida-dades* consisteix a recollir informació sobre l'historial clínic del pacient i resultats de proves de raigs X, anàlisis de sang, etc. Un cop s'ha obtingut més informació sobre el pacient, el metge decideix continuar amb el pla [bact-atip]. L'acció *bact-atip* consisteix a determinar el tipus de pneumònia que pateix el pacient (bacteriana o atípica). En el primer pacient troba, entre d'altres, els fets (dades-RX, {*condensació-vmaldef, confluents*}) i que el tipus de pneumònia és bacteriana (*pneumonia-bacteriana, molt-possible*), mentre que en el segon pacient troba (dades-RX, {*condensació-vmaldef, p-intersticial*}) i que el tipus de pneumònia és atípica

²Els casos que es presenten són una simplificació de casos reals. La utilització de casos reals, on el nombre de característiques oscilla sobre 90, és molt difícil d'utilitzar per produir exemples.

(*pneumonia-atípica, molt-possible*) (figura 2.2b). A partir d'aquest moment el metge segueix diferents estratègies per tal de localitzar l'agent microbià causant de les malalties de cada pacient. Cada pacient es resol finalment amb dos plans diferents (figura 2.2c). Les accions de cada pla s'executen i s'obtenen les solucions a cada cas (figura 2.2d). El context o situació en què el pla del pacient "b06v1" ha tingut lloc està determinat per tota la informació coneguda del pacient, des de les dades inicials fins que s'obté la darrera solució, és a dir,

$$S_{b06v1} = \{(\text{clínica, pneumònia extrahospitalària}), \\ (\text{estat-del-malalt, moderadament-greu}), \\ (\text{edat, 85}), (\text{sexe, home}), \\ (\text{forma-d'instauració, brusca}), \\ (\text{internat-en-un-asil, desconegut}), \\ (\text{dades-RX, \{condensació-vmaldef, confluents\}}), \\ (\text{pneumònia-bacteriana, molt-possible})\}$$

Es pot veure, que el context del pla del pacient "g01v1" és diferent, i.e.

$$S_{g01v1} = \{(\text{clínica, pneumònia extrahospitalària}), (\text{estat-del-malalt, lleu}), \\ (\text{edat, 27}), (\text{sexe, home}), (\text{forma-d'instauració, brusca}), \\ (\text{dades-RX, \{condensació-vmaldef, patró-intersticial\}}), \\ (\text{pneumònia-atípica, molt-possible})\}$$

Les situacions S_{b06v1} i S_{g01v1} són diferents tant en els atributs dels fets que contenen com en els valors dels atributs. En S_{b06v1} apareix el fet (*internat-en-un-asil, desconegut*) que no està contingut a S_{g01v1} . El fet *estat-del-malalt* pren el valor *moderadament-greu* en S_{b06v1} , mentre que en S_{g01v1} té el valor *lleu*. Podem dir que els plans "b06v1" i "g01v1" i dels casos són diferents perquè estan caracteritzats per diferents situacions.

L'exemple que acabem de ressenyar ens mostra com davant d'unes dades inicials molt semblants i poc rellevants, el mestre ha desenvolupat plans finals força diferents. Si representem cada cas com el conjunt de fets inicials i els plans finals, veurem que la representació no és adequada. Les dades inicials especificades en un problema no són suficients ni completes. Per exemple, en el cas "g01v1" les dades inicials no han donat més possibilitats al metge que plantejar-se una estratègia de recollida de més informació i discernir el tipus de malaltia; però no li han permès formular un conjunt d'hipòtesis versemblants, i.e. el pla complet. Si, portant la situació a l'altra extrem, representem els casos mitjançant el context final en què s'ha realitzat el pla, la representació tampoc serà adequada. Puix que BOLERO assumirà que les dades conegudes d'un problema són força completes. Una de les nostres motivacions a BOLERO és la de poder construir plans per a entorns on la informació és incompleta. Aleshores hem de convenir en una representació dels casos que vagi més enllà d'un context i un pla.

Per tal que BOLERO sigui capaç de planificar amb informació incompleta, hem convingut que els casos d'entrenament de BOLERO han d'estar representats com a seqüències d'episodis o passos de decisió. Un episodi recull la informació que es disposa en un determinant moment sobre l'entorn o situació actual s_a (conjunt de fets), i el pla constituït per les accions realitzades fins aquell moment i l'acció amb què es continua la resolució d'un problema. Així, el cas g01v1 de la figura 2.2 es representa com es mostra a la figura 2.3b. D'aquesta manera, el darrer episodi d'un cas conté un pla final, o el pla amb què s'ha resolt finalment el problema. Qualsevol altre episodi d'un cas té associat un pla parcial. Per exemple, l'episodi E4 de la figura 2.3b conté el pla parcial [*recollida-dades, bact-atip, micoplasma,*

virus], mentre que l'episodi E_6 té associat el pla final [recollida-dades, bact-atip, micoplasma, virus, clamídia, □].

Episodi:	Dades i plans:	
	(a) b06v1	(b) g01v1
E_1 :	(clínica, pneumònia extrahospitalària) (sexe, home) (forma-d'instauració, brusca) [recollida-dades]	(clínica, pneumònia extrahospitalària) (sexe, home) (forma-d'instauració, brusca) [recollida-dades]
E_2 :	(estat-del-malalt, mod-greu) (edat, 85) (internat-en-un-asil, desconegut) (dades-rx, cond-vmaldef, confluents) [recollida-dades, bact-atip]	(estat-del-malalt, lleu) (edat, 27) (dades-rx, cond-vmaldef, p-intersticial) [recollida-dades, bact-atip]
E_3 :	(mal-de-cap, desconegut) (pneumònia-bacteriana, molt-possible) [recollida-dades, bact-atip, pneumococ]	(mal-de-cap, desconegut) (pneumònia-atípica, molt-possible) [recollida-dades, bact-atip, micoplasma]
E_4 :	(pneumònia-pneumocòcica, possible) [recollida-dades, bact-atip, pneumococ, enterobacteriàcies]	(convivència-malalts-pneumònia, no) (pneumònia-micoplasma, mod-possible) [recollida-dades, bact-atip, micoplasma, virus]
E_5 :	(icterícia, desconegut) (pneumònia-enterobacteriàcies, molt-poc-possible) [recollida-dades, bact-atip, pneumococ, enterobacteriàcies, legionella]	(infiltrats-perihilars, desconegut) (pneumònia-vírica, força-possible) [recollida-dades, bact-atip, micoplasma, virus, clamídia]
E_6 :	... [recollida-dades, bact-atip, pneumococ, enterobacteriàcies, legionella, ...]	(pneumònia-clamídia, moderadament-possible) [recollida-dades, bact-atip, micoplasma, virus, clamídia, □]

Figura 2.3: Representació de casos per episodis.

En la secció 4.1 definirem formalment què és un episodi i què és un cas per a BOLERO. Remarquem, però, que aquesta descomposició d'un cas en episodis no és gens estranya, ja que és ben sabut que per a la transferència d'eficiència entre un problema passat i un de nou és important que l'aprenent tingui els seus coneixements de resoldre problemes representats en termes relativament petits de subobjectius i mètodes. La compilació dels coneixements es realitza després de l'aprenentatge, però un principiant prefereix els exemples basats en seqüències de passos [35].

El mètode d'aprenentatge per observació de BOLERO consisteix a organitzar jeràrquicament els episodis en la memòria. Un episodi E_i està representat en un node situat en el nivell i de la jerarquia. No existeixen dos nodes amb el mateix pla. Cada node emmagatzema informació d'un pla p i una situació generalitzada de totes les situacions sotes les quals s'ha executat p (i.e. situacions dels episodis amb el mateix pla). Així, els nodes terminals tenen associats plans finals, i la resta de nodes plans parcials.

2.2.1 Etapes en l'aprenentatge per observació

L'aprenentatge de plans per observació és un procés incremental en el qual hem distingit tres etapes: la incorporació d'un pla en memòria, la generalització de dues situacions d'episodis diferents amb plans idèntics i l'aprenentatge de condicions de finalització d'un problema. L'aprenentatge incremental permet refinar els coneixements de planificació en BOLERO mitjançant la incorporació d'un cas en memòria en qualsevol moment. El procés d'incorporar un pla d'un cas C en memòria es basa a recórrer la memòria de plans, de manera descendent

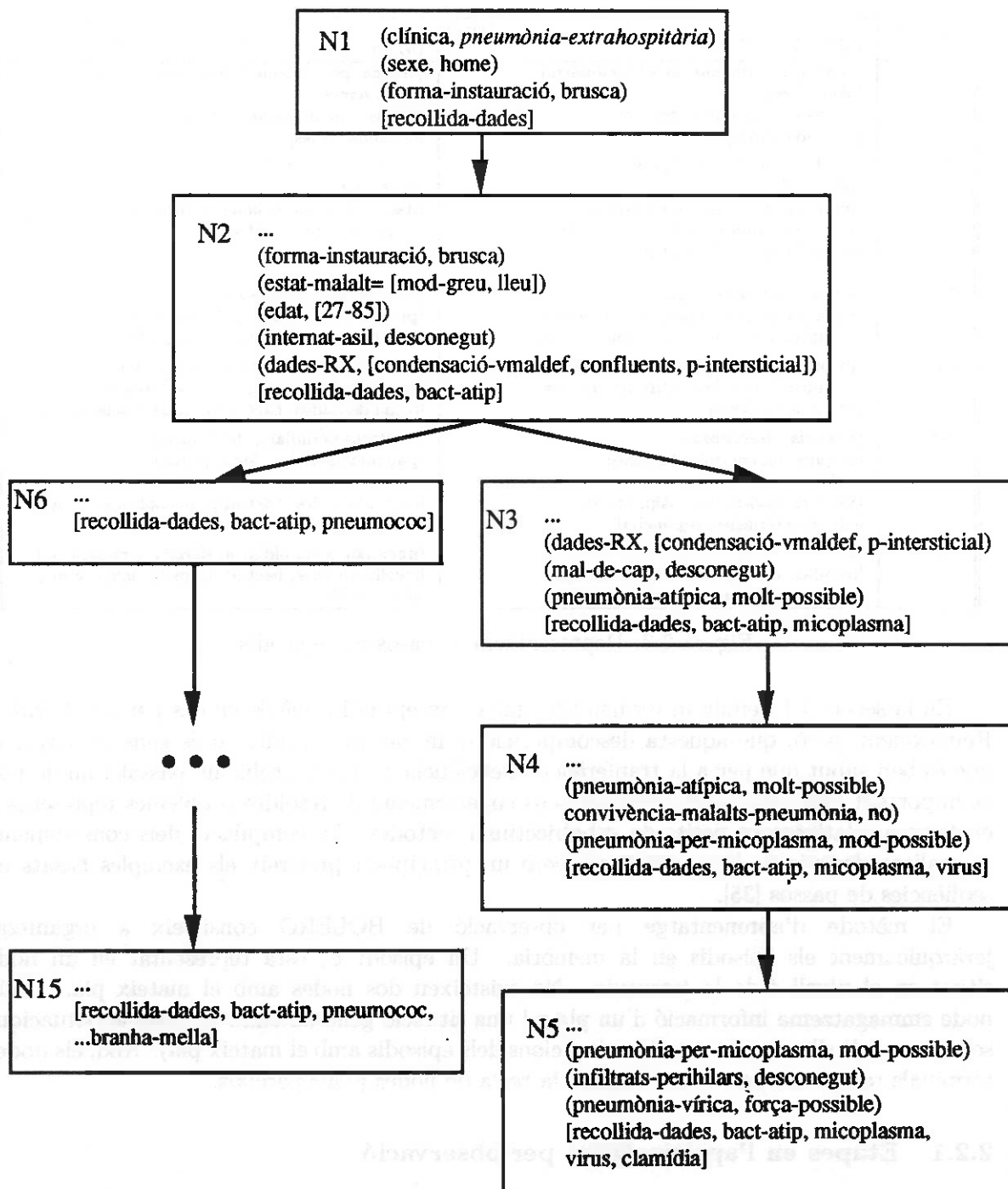


Figura 2.4: Memòria de BOLERO després d'haver après per observació els casos b06v1 i g01v1.

(*top-down*), tot comparant els plans dels episodis de C amb els plans dels nodes. La figura 2.4 mostra l'estat de la memòria després d'haver aplicat el mètode d'aprenentatge per observació en els casos "b06v1" i "g01v1". En el node N_1 hi ha representats tots els fets i el pla parcial de l'episodi E_1 ; en el node N_2 , el de l'episodi E_2 ; en el node N_3 , el de l'episodi E_3 , etc.

Les situacions generalitzades són el producte del procés de generalització entre els fets de situacions de diferents episodis amb el mateix pla. Per exemple, els episodis inicials E_1 i E_2 dels casos "b06v1" i "g01v1" tenen el mateix pla (vegeu la figura 2.3). Les situacions dels episodis E_1 són idèntiques, raó per la qual la situació generalitzada és la situació de qualsevol d'ells. El context dels episodis E_2 (b06v1) i E_2 (g01v1) són diferents, de la qual cosa en resulta la situació generalitzada en l'expressió que es mostra en el node N_2 de la figura 2.4. Les situacions generalitzades estan compostes de generalitzacions de fets (*norm* [81, 86]), i cada generalització està indexada pel fet o fets que l'han originada. Per exemple, els fets (*estat-malalt*, *mod-greu*) i (*estat-malalt*, *lleu*) dels episodis E_2 (b06v1) i E_2 (g01v1), respectivament, han donat lloc a la generalització (*estat-malalt*, [*mod-lleu*, *lleu*]). Cada generalització té associada una importància estratègica (un valor definit entre 0 i 1) que indica el pes (o rellevància) d'una generalització envers el node en què està emmagatzemada. Per exemple, en la diagnosi de pneumònies una generalització com (*dades-RX*, *condensació-vmaldef*, *patró-intersticial*) és més rellevant que la generalització (*sexe*, *home*), raó per la qual la importància del primer (0.7) és més elevada que la del segon (0.1). En el proper capítol es defineix més concretament el que és la importància estratègica d'un fet que juga un paper fonamental tant en el procés de construcció de plans com en l'aprenentatge per l'experiència pròpia.

Una particularitat que presenta BOLERO envers altres planificadors basats en casos és la inclusió d'un mètode d'aprenentatge de condicions de terminació. La creació d'aquest mètode és una conseqüència del tipus d'aplicacions desenvolupats en BOLERO: la diagnosi mèdica. En la planificació en un domini mèdic no es disposa de l'estat final abans de l'execució d'un pla. Això és, la planificació s'ha formulat tradicionalment com el procés de construir un pla que permeti arribar a un estat final donat un estat inicial. En la diagnosi mèdica l'estat inicial ve donat, però l'estat final és desconegut. El coneixement de l'estat final és equivalent al coneixement de la solució del problema, i.e., la malaltia del pacient. Si l'estat final no està disponible, aleshores és necessari que BOLERO disposi d'un mètode per reconèixer quan s'ha assolit la solució d'un problema. Altrament, un problema es resoldria per una cerca exhaustiva de totes les possibles solucions, amb la qual cosa no s'estaria millorant l'eficiència del *SBC^B*.

L'aprenentatge de condicions de finalització consisteix a adquirir aquelles condicions de domini que indiquen que s'ha arribat a la solució d'un problema (una solució correcta o prou bona). En la secció 4.3 veurem amb més detall tots aquests aspectes de l'aprenentatge per observació.

2.3 Construcció de plans amb informació incompleta

En un entorn dinàmic, la informació és incompleta i és necessari desenvolupar una tècnica de planificació que mitigui la manca d'informació. Els planificadors reactius són planificadors que en la construcció de plans tenen present el caràcter incomplet i canviant de les dades de l'entorn amb les quals basen el seu raonament. Un pla clàssic es produeix per resoldre un problema especificat en forma d'un estat o situació inicial i un objectiu o estat final.

En canvi, els plans reactius es generen d'acord amb les situacions reals d'un domini [139]. Resoldre un problema és equivalent a generar tants plans com canvis de situació hi ha a l'entorn, fins que se satisfacin els objectius. Per exemple, hem vist en la secció anterior com un metge construïa plans diferents en etapes (episodis) diferents ([recollida-dades], [bact-atip], i [micoplasma, virus, clamídia]) per resoldre el problema "g01v1". Per planificar reactivament no és indispensable conèixer anticipadament el resultat de les accions, com passa en la planificació clàssica, però és necessari alternar la construcció i l'execució dels plans.

L'alternació de la construcció i execució de plans és possible en l'arquitectura del sistema BOLERO-SBC. Com ja hem introduït en la secció 2.1, cada vegada que BOLERO construeix un nou pla, SBC^B l'executa. Cada vegada que la situació canvia (i.e. canvien les dades conegudes del pacient) pels efectes del procés de raonament en SBC^B (i.e. l'adquisició de dades per preguntes al pacient), BOLERO construeix nous plans. L'alternació de la construcció i l'execució de plans determina BOLERO com un planificador reactiu.

La construcció de plans en BOLERO es basa en les experiències de plans organitzades en memòria (recordem que com a resultat de l'aprenentatge per observació BOLERO té una memòria jeràrquica on els nodes emmagatzemen generalitzacions i plans). El mecanisme de BOLERO per a la construcció de plans, com qualsevol sistema basat en casos, consta de dos passos fonamentals: recuperació i adaptació.

Recuperació. Suposant un conjunt de dades conegudes de l'entorn o situació s_a del problema que s'esta resolent, el procés de *recuperació* consisteix a activar les situacions generalitzades emmagatzemades en memòria mitjançant la indexació de les generalitzacions a partir dels fets de s_a i a determinar el grau de comparació o *matching* entre s_a i les situacions generalitzades activades. Per comparar s_a amb qualsevol situació generalitzada v de la memòria, BOLERO té dues mesures. La primera mesura m_f serveix per comparar un fet de s_a i una generalització de v . La segona mesura m_s s'utilitza per combinar la semblança entre tots els fets de s_a i totes les generalitzacions de v , resultant en un grau de semblança total entre s_a i v (ambdues mesures estan definides en l'interval [0,1]). En el proper capítol especifiquem amb més detall cadascuna de les mesures (veure secció 4.4), però volem remarcar que la importància estratègica dels fets, com ja s'ha introduït en la secció anterior, juga un paper fonamental. Per exemple, suposem que BOLERO vol recuperar un pla de la memòria (figura 2.4) per trobar un diagnòstic per al malalt "g24v1" les dades del qual són: (clínica, pneumònia extrahospitalària), (estat-del-malalt, lleu), (edat, 30), (sexe, home), (forma-d'instauració, subaguda), (dades-RX, Broncograma-aeri, condensació-vmaldef, patró intersticial). A continuació es mostren els valors de semblança entre algunes de les generalitzacions emmagatzemades en el node N_2 de la figura 2.4 i les dades del malalt de "g24v1":

- $m_f(\text{clínica}_{N_2}, \text{clínica}_{g24v1}) = 0.1$. El resultat esperat fóra $m_f = 1.0$ car el fet clínica_{g24v1} i la generalització clínica_{N_2} tenen el mateix valor: pneumònia-extrahospitalària. Però el fet clínica té un importància estratègica molt petita en N_2 , cosa que fa disminuir el resultat de m_f .
- $m_f(\text{dadesRX}_{N_2}, \text{dadesRX}_{g24v1}) = 0.7$. En aquest cas, malgrat que els valors no coincideixen (en N_2 el valor és {condensació-vmaldef, confluents, p-intersticial} i en $g24v1$ és {broncograma-aeri, condensació-vmaldef, p-intersticial}), la importància estratègica de la generalització dadesRX_{N_2} és molt

alta, per la qual cosa el resultat de m_f és més alt del que s'espera a primer cop d'ull.

Adaptació. L'última estapa en la construcció d'un pla consisteix a adaptar el pla recuperat de la memòria a la situació actual. En el procés d'adaptació de plans hem assumit que un cop una acció ha estat realitzada, els seus efectes perduren per sempre. Assumir el comportament monòton dels efectes de les accions és una conseqüència de les aplicacions que s'han utilitzat en el desenvolupament de BOLERO. Es tracta d'aplicacions en medicina on s'espera obtenir un diagnòstic del pacient. Per tant, si s'obtenen un conjunt de dades del pacient com a conseqüència de l'execució d'una acció, és de suposar que aquestes dades no canviïn en el transcurs de la decisió del diagnòstic³. Feta aquesta suposició, l'adaptació de plans consisteix bàsicament a no repetir accions que ja han estat realitzades. Per exemple, suposem la memòria de plans que es mostra a la figura 2.5a. En un moment donat s'ha recuperat el pla del node $n5$ de memòria i s'ha aplicat per resoldre un cas nou C (figura 2.5a). Durant l'execució del pla [recollida-dades, bact-atip, pneumococ, enterobact, legionel.la], concretament un cop realitzada l'acció enterobact, es recupera el pla del node $n9$ (figura 2.5b). L'adaptació del pla de $n9$ resulta en el pla [estafilococ, tuberculosi] que tot seguit s'executa.

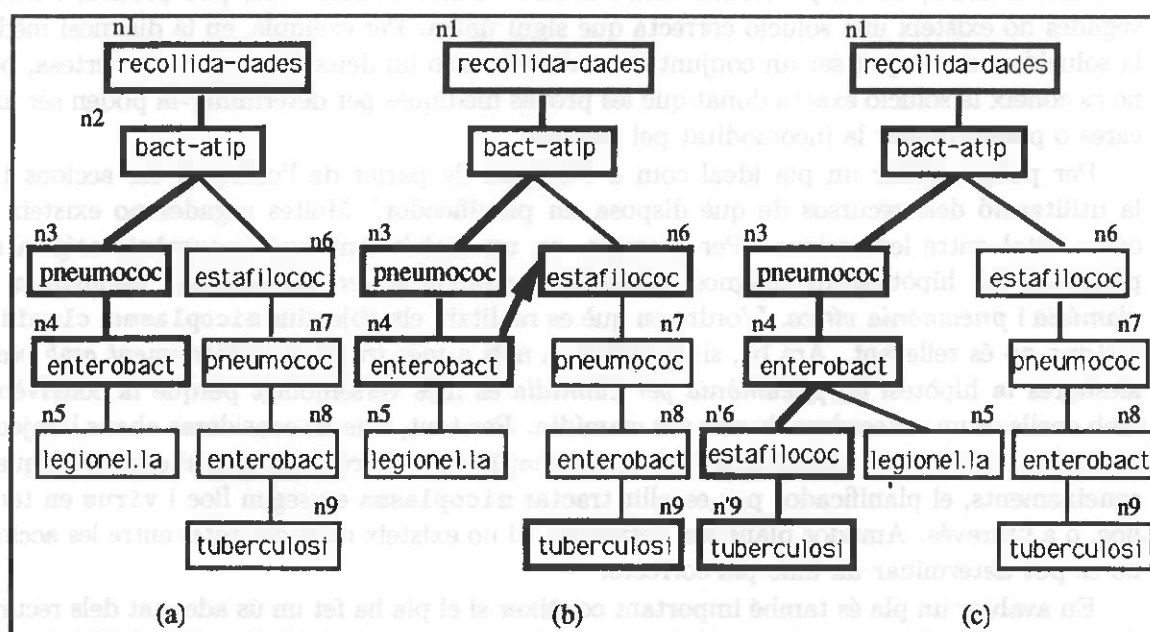


Figura 2.5: Representació esquemàtica de la resolució d'un cas. (a) El pla construït es basa en el pla del node $n5$. (b) El pla construït es basa en el pla del node $n9$. (c) Estat de la memòria després de la incorporació del cas nou. Per simplicitat, en els nodes de l'arbre només està representat la darrera acció que compon el pla parcial associat al node. Així, el node $n9$ té associat el pla [recollida-dades, bact-atip, estafilococ, pneumococ, enterobact, tuberculosi].

³ Aquesta suposició, però, no fóra vàlida en entorns on el seguiment d'un pacient es fa a llarg termini, com és el cas dels tractaments.

Després que BOLERO hagi construït n plans i SBC^B els hagi executat, BOLERO decideix terminar la resolució d'un problema fent ús dels coneixements apresos en l'aprenentatge per observació (secció 2.2). Quan un problema ha estat resolt, aleshores es pot avaluar el pla que finalment ha estat executat, i.e., la seqüència d'accions que efectivament han estat realitzades des de l'inici de la resolució en BOLERO-SBC.

2.4 Avaluació de plans

L'objectiu de l'avaluació és conèixer el reeiximent o el fracàs del pla final d'un cas resolt per BOLERO. Per tal d'avaluar plans, però, cal tenir coneixements (propis o d'un mestre) que permetin valorar una actuació com a correcta o com a errònia, i eventualment calcular el grau de bondat i correctesa. Els coneixements per realitzar l'avaluació resideixen en un patró d'or [100]. El patró d'or (PO) és la solució correcta, l'esperada o desitjada. El patró d'or és un model de funcionalitat que com a tal ha de recollir tant les solucions del domini com el pla òptim. Com es pot intuir, és molt difícil de disposar d'un PO, a causa principalment de les particularitats dels dominis pels quals es desenvolupen els SBC. Es tracta de dominis mal estructurats, on els problemes estan definits d'una manera laxa, poc precisa, i moltes vegades no existeix una solució correcta que sigui única. Per exemple, en la diagnòsi mèdica la solució a un cas pot ser un conjunt de malalties amb un determinat grau de certesa, però no es coneix la solució exacta donat que les proves mèdiques per determinar-la poden ser molt cares o poden causar la incomoditat pel pacient.

Per poder definir un pla ideal com a PO hem de parlar de l'ordre de les accions i de la utilització dels recursos de què disposa un planificador. Moltes vegades no existeix un ordre total entre les accions. Per exemple, en un malalt amb una pneumònia atípica són plausibles les hipòtesis de diagnòsi següents: *pneumònia per micoplasma*, *pneumònia per clamídia* i *pneumònia vírica*. L'ordre en què es realitzin els objectius *micoplasma*, *clamídia*, i *virus* no és rellevant. Ara bé, si el pacient a més a més tracta quotidianament amb ocells, aleshores la hipòtesi de *pneumònia per clamídia* és més versemblant perquè la convivència amb ocells és un antecedent de risc per clamídia. Per tant, s'ha de considerar abans l'objectiu *clamídia* que d'altres menys plausibles com *micoplasma* o *virus*. En aquest cas, amb aquests coneixements, el planificador pot escollir tractar *micoplasma* en segon lloc i *virus* en tercer lloc, o a l'inrevés. Ambdós plans són correctes. Si no existeix un ordre total entre les accions, no es pot determinar un únic pla correcte.

En avaluar un pla és també important conèixer si el pla ha fet un ús adequat dels recursos de què disposa. Per exemple, en el diagnòstic d'un pacient no es pot suggerir fer-li mil i una proves de laboratori per assegurar la malaltia que pateix; s'han de realitzar les proves més útils d'acord amb la informació coneguda del pacient. Per exemple, seria un absurd realitzar una anàlisi del líquid pleural si el malalt no presenta un embassament pleural en les radiografies. Per dur a terme l'avaluació de plans no és tan rellevant veure si els objectius s'han satisfet com saber si el pla construït per assolir-los és eficient. Consegüentment, la determinació d'un PO per l'avaluació no és una cosa fàcil.

Donada la dificultat que suposa determinar un PO els sistemes d'aprenentatge automàtic convenen a consultar un oracle per validar el que aprenen. L'aproximació de BOLERO ha estat la de definir un patró d'avaluació (PA) que s'acosti el més possible a la solució ideal (PO). El PA d'un cas C (que escrivim $PA(C)$) el conformen un pla final admissible, $PA_p(C)$, i un conjunt de solucions de domini, $PA_d(C)$. El pla PA_p el proporciona un únic expert que

anomenem expert mestre. Les condicions de domini PA_d estan construïdes sobre la base d'enquestes fetes a més d'un expert. En certa manera PA_d complementa PA_p . Avaluar els plans a partir d'un únic expert, i aprendre a partir de l'experiència, conduiria BOLERO a mimetitzar el mestre. Sabem que l'expertesa no és universal [143, 151], i a més a més el mestre és un expert humà, per la qual cosa és susceptible d'equivocar-se. Per tant, fóra desitjable recollir l'opinió de més d'un expert per avaluar plans. Seguint aquest raonament, es podria pensar a construir PA_p com un pla consensuat per un grup d'experts. Però resultaria força costós fer coincidir un conjunt d'experts en una solució única⁴. L'apropament de BOLERO ha estat la de construir PA_p basat en un únic expert, l'expert mestre, però alhora considerar l'opinió d'altres experts, i.e. PA_d , en l'avaluació de plans. D'aquesta manera, l'oracle de BOLERO el constitueixen l'expert mestre i un grup d'experts. Mentre que el PA_p ha estat utilitzat per mesurar la correctesa dels plans, el PA_d , juntament amb el PA_p , han estat utilitzats per avaluar la precisió dels plans (vegeu la secció 4.5). Un pla ha reeixit si és correcte i precís, altrament és un pla imperfecte. A partir de l'avaluació dels plans es consolida l'aprenentatge per experiència.

2.5 Aprenentatge a partir de plans reixits i plans imprecisos

Els *sistemes aprenents*, entre els quals hi ha BOLERO, són sistemes que contínuament aprenen de la seva pròpia actuació sense que explícitament sel's indiqui que ho han de fer [113]. L'aprenentatge per l'experiència pròpia es materialitza en BOLERO per l'aprenentatge del reeiximent d'un pla en resoldre un problema, a partir de plans imperfectes.

Un pla reeixit pot ser un pla present en un cas de la memòria o un pla nou construït per BOLERO com a conseqüència de la planificació basada en casos i de la planificació reactiva durant la resolució d'un problema. És a dir, el pla final contruït per resoldre el problema és la seqüència de les parts dels plans que BOLERO ha generat i que efectivament ha executat. Recordem, per exemple, el cas *C* de la figura 2.5a: en primer lloc s'ha executat el pla del node *n5*, i després s'ha recuperat i adaptat el pla del node *n9*. Finalment, per resoldre el cas *C* tenim el pla final: [recollida-dades, bact-atip, pneumococ, enterobact, estafilococ, tuberculosi]. Aquest nou pla no figura en memòria sinó que s'ha generat adaptant plans parcials de la memòria gràcies a la capacitat reactiva del sistema, que permet en tot moment avaluar una situació, reconsiderar el pla millor i adaptar el pla a la nova situació. BOLERO pot aprendre de l'èxit del pla amb la incorporació d'aquest pla nou resultant en una nova organització de la memòria (figura 2.5c).

Quan un pla és imperfecte, també es pot aprendre. En primer lloc l'oracle proporciona el cas correcte, el qual és incorporat en memòria, aprenent, doncs, plans nous. En segon lloc, donat que els índexs han causat la recuperació errònia d'un cas en memòria, es pot establir un mecanisme per tal d'evitar que en les mateixes circumstàncies es torni a recuperar nodes erronis. Una forma de modificar la indexació de la memòria és alterar la importància estratègica de les generalitzacions indexades i que han resultat determinants en la recuperació del pla imprecís.

D'aquesta manera, en l'aprenentatge per l'experiència hem distingit tres mètodes: 1) aprenentatge pel reeiximent, 2) aprenentatge a partir de plans imperfectes i 3) actualització de la importància estratègica de les generalitzacions. Tots els mètodes es veuran amb detall

⁴A més a més, la disponibilitat d'experts humans és força cara, cosa que obliga quasi tots els sistemes d'aprenentatge automàtic a aprendre a partir d'un sol mestre.

en el capítol 4.

2.6 Conclusions. Planificació reactiva basada en casos

Al llarg d'aquest capítol hem introduït BOLERO, un planificador basat en casos integrat en l'arquitectura metanivell BOLERO-SBC. La planificació basada en casos és utilitzada per BOLERO per construir plans per tal de millorar l'eficiència d'un SBC i aprendre plans de l'experiència pròpia. Abans de poder planificar a partir de casos, BOLERO aprèn plans des de zero. És a dir, inicialment BOLERO manca de qualsevol coneixement sobre planificació. Per aquesta raó és necessari realitzar una etapa preliminar d'aprenentatge de plans per observació en la qual BOLERO aprèn plans per l'observació de problemes resolts per un mestre (casos). L'aprenentatge per observació proporciona coneixements sobre plans indexats pels contextos (situacions generalitzades) en els quals s'han aplicat. En aquest sentit, i tenint en compte la hipòtesi de monotonía dels efectes de les accions, el problema del context (*frame problem*) que presenten els planificadors tradicionals en BOLERO queda resolt tàcitament en l'organització dels plans en la memòria, cosa que ja ha estat observada en la planificació basada en casos en general [66].

Després de l'aprenentatge per observació, BOLERO està preparat per construir plans per resoldre problemes nous. Les etapes de construcció, avaluació i aprenentatge per l'experiència se succeeixen en la resolució de problemes.

La construcció d'un pla està imbricada amb l'execució mercès a la col·laboració de BOLERO i SBC^B , que interaccionen mitjançant una interfície. En el moment de començar a resoldre un problema, BOLERO modifica l'estat d'objectius de l'interpret de SBC^B amb un pla per defecte (p). SBC^B executa el pla i és SBC^B qui realment interacciona amb l'entorn. Quan SBC^B coneix una nova dada (bé per l'aplicació del seus propis coneixements del domini, bé per un diàleg establert amb l'usuari, bé per la informació percebuda d'un sensor), la comunica al planificador. Aleshores BOLERO pot *recordar* un pla millor p' i decidir canviar el pla actual p per p' . BOLERO té així un comportament reactiu.

Un cop resolt un problema, BOLERO té una experiència nova en la contrucció de plans, és a dir, un cas nou C^n , que com a conseqüència de la planificació reactiva pot ser un exemplar de pla diferent a totes les experiències anteriors $C^{n-1}, \dots, C^0$. Una avaluació del pla final de C^n indica la correctesa i la precisió del pla. De resultes de l'avaluació del pla, BOLERO reorganitza la seva memòria mercès a la incorporació del cas en memòria i a la modificació de la importància estratègica de les característiques, aprenent doncs per la seva pròpia experiència.

Capítol 3

L'arquitectura metanivell BOLERO-SBC

El sistema BOLERO-SBC és el resultat de la integració del planificador basat en casos BOLERO i un sistema basat en coneixements SBC (que notem per SBC^B), en una arquitectura metanivell. BOLERO juga el paper del metanivell i SBC^B el paper del nivell objecte (veure figura 3.1¹). SBC^B té coneixements sobre un domini específic que li permeten resoldre problemes quan segueix un pla estratègic, mentre que BOLERO té coneixements de planificació particulars a aquest domini. L'arquitectura del sistema BOLERO-SBC és una arquitectura reflexiva [102]: el metanivell (BOLERO) és capaç de modificar l'estat del nivell objecte (SBC^B) mitjançant plans, i el metanivell s'ocupa de començar i aturar l'execució del sistema. En aquest capítol descriurem com és l'arquitectura BOLERO-SBC. En primer lloc es descriuen cadascun dels nivells de l'arquitectura (seccions 3.1 i 3.2). Després es detalla com els nivells estan connectats causalment (secció 3.3), i com el sistema presenta un comportament reactiu (secció 3.4). Finalment (secció 3.5), s'esmenten els guanys aconseguits en la integració de BOLERO i SBC^B .

3.1 El nivell objecte

Com ha estat presentat amb anterioritat, el nivell objecte és un sistema basat en coneixements que anomenem SBC^B . Per tal de concretitzar, hem suposat que el sistema basat en coneixements escollit com a nivell objecte (i.e. SBC^B) es tracta d'un sistema basat en regles. Així, SBC^B , com qualsevol sistema basat en regles, té un intèrpret i una base de coneixements. La base de coneixements conté fets, objectius i regles. SBC^B pot funcionar independentment de BOLERO-SBC. És a dir, el SBC^B disposa d'un pla per defecte que és una seqüència composta per tots els objectius definits en SBC^B . L'intèrpret de SBC^B resol un problema mitjançant l'ús de les regles que permeten inferir un a un tots els objectius especificats en el pla per defecte. Així, diem que SBC^B *valida* un objectiu o quan el seu intèrpret està intentant assolir o a partir de les regles i els fets. La validació de tots els objectius del pla per defecte condueix a la resolució d'un problema.

SBC^B es pot, doncs, representar com un parell $\langle F_{SBC}, R \rangle$. R és un conjunt de regles que permeten realitzar inferències dins del domini de SBC^B . F_{SBC} és el conjunt de fets

¹La figura 3.1 és la mateixa que la figura 2.1 del capítol 2, que hem repetit aquí per conveniència.

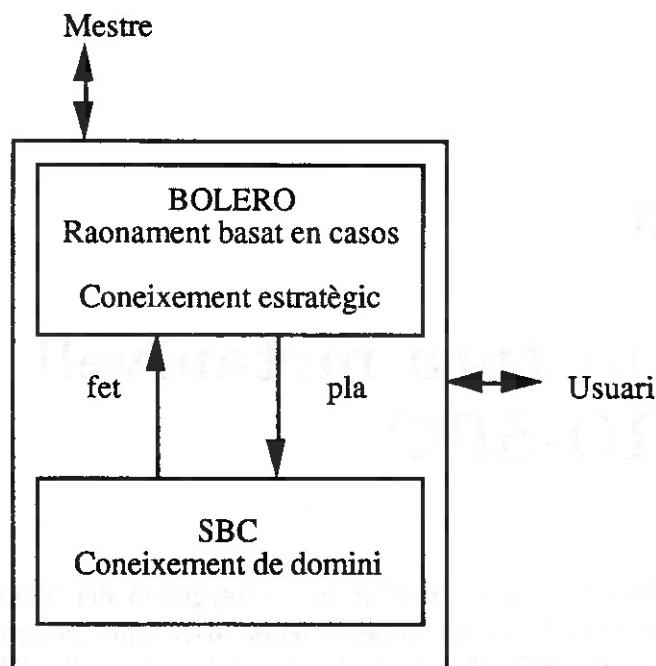


Figura 3.1: Sistema integrat resultat de la interacció entre BOLERO i un SBC.

d'un domini. Cada fet f de F_{SBC} és un parell (*atribut, valor*). Els fets de F_{SBC} estan classificats entre fets d'entrada (F_e), fets deduïbles (F_d) i objectius (*goals*, F_g). Els fets d'entrada són fets que, en la resolució de problemes, SBC^B obté mitjançant un diàleg amb l'usuari. Els fets deduïbles són fets que SBC^B obté per l'aplicació de les regles. Els objectius són fets que SBC^B obté, com els fets deduïbles, per l'aplicació de les regles, però que a més a més formen part de la solució d'un problema. Per exemple, el fet (*edat*, 79) pertany a F_e i diem que el fet *edat* té valor 79; el fet (*vell*, *cert*) és un fet deduïble el valor del qual és *cert*; i el fet (*pneumònia-per-pneumocistii-carinii*, *força-possible*) és un objectiu el valor del qual és *força-possible*. Entre els objectiu F_g cal destacar aquells que tenen un valor assignat, i.e. estan assolits, d'aquells que encara no tenen cap valor. Per exemple, l'objectiu (*pneumònia-pneumococ*, *possible*) diem que té el valor *possible*, mentre que l'objectiu (*pneumònia-pneumococ*, $\square$) diem que té el valor desconegut. R és el conjunt de regles heurístiques que SBC^B utilitza per tal de deduir els objectius de F_g a partir dels fets d'entrada F_e ². Aleshores tenim que:

$$SBC^B = \langle F_{SBC}, R \rangle$$

$$F_{SBC} = F_e \cup F_d \cup F_g$$

En el moment de resoldre un problema, SBC^B pren diferents estats entre tots els seus possibles estats Σ_{SBC} . En cada estat σ_{SBC} es distingeix el conjunt de fets d'entrada actuals E , fets deduïts D , objectius G que ja han estat assolits, i la seqüència actual p d'objectius que l'interpret de SBC^B està intentant d'assolir (pla actual). De manera que si P és el conjunt de

²Cal observar que en lloc de regles podríem representar altres tipus d'estructures, com ara operadors. Les característiques del SBC^B que es presenten són conseqüència del sistema basat en regles utilitzat en l'avaluació de BOLERO ([166]), però creiem que són característiques aplicables a altres tipus de sistemes si treballen amb una estructura de fets de tipus (*atribut, valor*).

tots els plans possibles que el SBC^B és capaç d'executar (i.e. $P = \{[f_{g_1}, \dots, f_{g_n}] | f_{g_j} \in F_g\}$), el conjunt de tots els estat possibles Σ_{SBC} de SBC^B ve donat per:

$$\Sigma_{SBC} = \mathcal{P}(F_e) \times \mathcal{P}(F_d) \times \mathcal{P}(F_g) \times P$$

Així un estat $\sigma_{SBC} \in \Sigma_{SBC}$ de la resolució de problemes en SBC^B es representa per la tupla següent:

$$\sigma_{SBC} = (E, D, G, p)$$

on $E \in \mathcal{P}(F_e)$, $D \in \mathcal{P}(F_d)$, $G \in \mathcal{P}(F_g)$ i $p \in P$.

Un canvi d'estat en el procés de raonament de SBC^B ocorre quan SBC^B coneix nous fets bé per l'aplicació de les regles, bé per qüestions preguntades a l'usuari. Si s'adquireix una dada de l'usuari, f_e , es modifica el conjunt E ; si s'aplica una regla deduint un nou fet, es modifica o bé D o bé G . Si s'aplica una regla, i es dedueix un objectiu o , llavors el pla per p de l'interpret canvia de manera que el pla nou no conté l'objectiu o (i.e. $p' = p - \{o\}$). Quan el pla de l'interpret canvia, l'estat de SBC^B canvia consegüentment³. Resumint, SBC^B canvia d'un estat a un altre per l'addició d'un nou fet i/o un canvi de pla, tal i com es mostra a continuació:

$$\sigma_{SBC} \xrightarrow{R} \sigma'_{SBC}$$

on:

$$\begin{aligned} \sigma_{SBC} &= (E, D, G, p) \\ \sigma'_{SBC} &= (E', D', G', p') \end{aligned}$$

i

$$\begin{aligned} E &\subseteq E', D \subseteq D', G \subseteq G' \\ |E'| + |D'| + |G'| &= |E| + |D| + |G| + 1 \\ p &= p' \vee p \neq p' \end{aligned}$$

Aquesta definició de canvi d'estat suposa que les situacions són monòtones i que els fets són deduïts seqüencialment. Això és, els fets poden ser afegits un a un en cada canvi d'estat i mai poden ser eliminats.

3.2 El metanivell

BOLERO constitueix el metanivell del sistema BOLERO-SBC. BOLERO com a planificador basat en casos aprèn coneixements de planificació a partir de casos, i llavors construeix nous plans per resoldre nous problemes. BOLERO és doncs un sistema en si mateix, del qual és interessant remarcar que, tal i com es descriu a la secció 4.1, la qualitat dels plans generats per BOLERO depèn de la completesa de la informació (fets) disponible per a un problema donat (un nou cas). Quan més rica és la informació que BOLERO té sobre un problema, el pla que genera és més precís.

Per tal de ser capaç de planificar els objectius del nivell objecte, BOLERO, com a metanivell en el sistema BOLERO-SBC, ha de tenir un model parcial de SBC^B (vegeu secció 1.2.1). Aquest model permet a BOLERO raonar sobre l'estat del nivell objecte en la resolució de problemes i actuar en conseqüència. El model parcial que té BOLERO de SBC^B es basa en els fets F_e i F_g de F_{SBC} . Els conjunts de fets d'entrada F_e i objectius F_g de SBC^B són representats a BOLERO pels conjunts QF_e i QF_g respectivament, i diem que els

³En la secció 3.3 veurem com un canvi d'estat causat per un canvi de pla també es pot produir quan BOLERO proporciona un nou pla d'actuació a SBC^B .

conjunts QF_e i QF_g són la codificació (*quotation*) dels conjunts F_e i F_g de SBC^B en BOLERO. Un fet $f_i \in F_e$ de SBC^B està representat en BOLERO com un fet $[f_i] \in QF_e$, on $[f_i]$ és la codificació de f_i , i.e. la representació de f_i en el llenguatge de BOLERO. Anàlogament, un fet $f_g \in F_g$ en SBC^B està representat com $[f_g] \in QF_g$ en BOLERO, on $[f_g]$ és la codificació de f_g . L'estructura dels fets en BOLERO es manté, és a dir, un fet en BOLERO també està representat per un parell (*atribut, valor*). L'excepció són els fets de SBC^B que tenen un valor numèric perquè es codifiquen en BOLERO mitjançant intervals. Així, el fet (*edat*, 69) de SBC^B es codifica en BOLERO com (*edat*, [65, 69]). Aquest procés de transformar els valors numèrics en intervals és un procés que hem anomenat *normalització* i que es veu amb més detall en el proper capítol (secció 4.1).

Aleshores BOLERO pot representar-se també per un parell (F_{BOLERO}, C) , on F_{BOLERO} és el conjunt de fets constituït per la unió de QF_e i QF_g (i.e. $F_{BOLERO} = QF_e \cup QF_g$); C és el conjunt de casos que BOLERO té en memòria.

Per tal de generar plans que SBC^B sàpiga interpretar, BOLERO també coneix les accions que es poden dur a terme en el SBC^B . Cada acció a en BOLERO s'interpreta com la validació d'un objectiu $f \in F_g$ en SBC^B . Un pla de BOLERO és una seqüència d'accions que es representa per $[p]$. Per tant, un pla de BOLERO és una seqüència d'accions $a_1, a_2, \dots, a_n$ que equival a un pla de SBC^B constituït per la seqüència d'objectius $f_{g1}, f_{g2}, \dots, f_{gn}$. Les accions, i així els plans, estan representades en el conjunt de casos C . Anomenem QP al conjunt de tots els plans possibles que BOLERO pot generar a partir de C , i que es fa correspondre a un conjunt P^B de plans que SBC^B pot interpretar (*reflect down*)⁴.

Donat un conjunt de fets $QE \in \mathcal{P}(QF_e)$, i/o un conjunt de fets $QG \in \mathcal{P}(QF_g)$, BOLERO pot generar un pla $[p] \in QP$. Un estat de BOLERO, σ_{BOLERO} , es defineix com dos conjunts de fets QE i QG , i el pla $[p]$, construït a partir dels fets. El pla generat depèn del conjunt de casos (C) que BOLERO té en la memòria (vegeu secció 4.1). Així, el conjunt de tots els estats possibles de BOLERO, Σ_{BOLERO} , ve determinat per:

$$\Sigma_{BOLERO} = \mathcal{P}(QF_e) \times \mathcal{P}(QF_g) \times QP$$

Un estat en particular σ_{BOLERO} ve definit per:

$$\sigma_{BOLERO} = (QE, QG, [p])$$

on $QE \in \mathcal{P}(QF_e)$, $QG \in \mathcal{P}(QF_g)$ i $[p] \in QP$.

BOLERO no pot canviar d'estat per si mateix si no és que el conjunt de dades $QE \cup QG$ li ve donat o es modifica. Aleshores BOLERO pot generar un pla nou i canviar d'estat. Sota l'arquitectura BOLERO-SBC, quan SBC^B és capaç d'interpretar i executar els plans que BOLERO construeix, BOLERO canvia d'estat diverses vegades durant la resolució d'un problema, com es veurà en la secció següent.

3.3 Regles de reflexió

Com s'ha introduït en el capítol 1, en un sistema reflexiu existeixen tres elements bàsics:

1. el model parcial que té el metanivell sobre el nivell objecte,

⁴S'observa que el conjunt de plans P que SBC^B és capaç d'interpretar pot ser més gran que el conjunt de plans P^B que BOLERO és capaç de generar. La generació de plans de BOLERO depèn de la seva capacitat d'aprenentatge.

2. el conjunt de regles de reflexió que garanteixen la connexió causal dels nivells, i
3. el mecanisme de canvi de nivell (*meta-level lift up* i *meta-level reflect down*).

El model parcial que BOLERO té com a metanivell del nivell objecte (SBC^B) està constituït pels conjunts de fets QF_e i QF_g , com ha estat esmentat amb anterioritat. Les regles de reflexió arbitren la connexió entre els nivells de l'arquitectura BOLERO-SBC. Les regles de reflexió ascendent indiquen els canvis d'informació que es produeixen en el nivell objecte i que BOLERO necessita conèixer per tal d'avaluar l'estat del nivell objecte en la resolució de problemes, i planificar plans d'actuació conseqüents. Les regles de reflexió ascendents (R_a) de BOLERO-SBC són les següents:

$$R_{a_e}: \frac{f_e}{[f_e]} \quad R_{a_g}: \frac{f_g}{[f_g]}$$

Les regles de reflexió descendent indiquen com les decisions preses en el metanivell tenen les seves conseqüències en el nivell objecte. En BOLERO-SBC només s'ha definit una regla de reflexió descendent, on es mostra com la generació d'un pla per BOLERO modifica el pla que l'interpret de SBC^B està validant. Així es garanteix que SBC^B segueix causalment els plans generats per BOLERO. La regla descendent (R_d) del sistema és la següent:

$$R_d: \frac{[p]}{p}$$

Les regles de reflexió de l'arquitectura BOLERO-SBC són realment regles molt senzilles. Tot canvi d'informació que es produeix en el model que BOLERO té del nivell objecte (SBC^B) és actualitzat en el metanivell, com ho regulen R_{a_e} i R_{a_g} . I les decisions preses en el metanivell són seguides pel nivell objecte mitjançant els plans, tal i com indica la regla de reflexió descendent R_d .

El darrer element a tenir en compte en una arquitectura reflexiva és el mecanisme de canvi de nivell. El canvi de nivell en BOLERO-SBC està implícit en l'arquitectura: cada cop que SBC^B obté un fet f_e o f_g , el control de BOLERO-SBC passa al metanivell (*meta-level lift up*). El metanivell pot retornar simplement el control al nivell objecte (si BOLERO no genera un pla nou) o modificar l'estat de l'interpret del nivell objecte abans de retornar el control (*meta-level reflect down*).

Meta-level lift up. Quan SBC^B rep un fet nou de l'entorn f_e o SBC^B dedueix un objectiu f_g , es produeix un *meta-level lift up* (veure secció 1.2.1), i el nou fet ($[f_e]$ o $[f_g]$) és conegut en BOLERO, tal i com estableixen les regles de reflexió ascendents. Per exemple, suposem que les regles que es mostren en la figura 3.2 són en R . Si SBC^B adquireix de l'usuari el fet (*edat*, 62), aleshores el fet $[(\text{edat}, [60, 64])]$ és conegut per BOLERO, on $[(\text{edat}, [60, 64])]$ és la representació del fet (*edat*, 62) en BOLERO. Si el mecanisme d'inferència del sistema basat en regles aplica la regla R01010b que porta a la conclusió del fet (*vell*, *cert*), no hi ha cap connexió entre SBC^B i BOLERO. El fet (*vell*, *cert*) no és ni un fet d'entrada ni un fet objectiu, sinó que es tracta d'un fet deduït. No és possible aplicar la regla de reflexió ascendent i, en conseqüència, no hi ha ocasió per a un *meta-level lift up*. Tanmateix, si l'interpret de SBC^B aplica la regla R04009 i dedueix el fet (*pneumònia-pneumococ*, *possible*), on (*pneumònia-pneumococ*, *possible*) pertany a F_g (i.e. es tracta d'un fet objectiu), aleshores

es produeix un *meta-level lift up* i el fet [(pneumònia-pneumococ, possible)] és conegut per BOLERO.

R01010b: Si edat $\in (60, 64]$

Llavors concloure vell és cert

R04009: Si vell $\wedge$ pneumònia-bacteriana és possible

Llavors concloure pneumònia-pneumocòccia és possible

Figura 3.2: Regles del sistema basat en regles PNEUMON-IA [166].

Així, un *meta-level lift up* es pot notar com segueix:

$$\Sigma_{SBC} \rightarrow \Sigma_{BOLERO}$$

Un estat de $\sigma_{SBC} = (E, D, G, p)$ en Σ_{SBC} es correspon a un estat de BOLERO $\sigma_{BOLERO} = (QE, QG, [p])$. El conjunt de fets E de l'estat σ_{SBC} correspon al conjunt de fets QE en l'estat corresponent de BOLERO, i.e. el conjunt de fets QE en BOLERO és la codificació del conjunt de fets E d'un estat de SBC^B . Anàlogament ocorre amb els conjunts G de σ_{SBC} i QG de σ_{BOLERO} . Els conjunts $QE \in \mathcal{P}(QF_e)$ i $QG \in \mathcal{P}(QF_g)$ permeten a BOLERO conèixer l'estat del nivell objecte en la resolució d'un problema.

Abans d'un *meta-level lift up* SBC^B està en un estat determinat $\sigma_{SBC_{ant}}$. Un *meta-level lift up* ocorre perquè SBC^B coneix un nou fet, i per tant es modifica E_{ant} o G_{ant} i esdevé un nou estat σ_{SBC} amb $E_{ant} \subset E$ si el nou fet f pertany a E , o $G_{ant} \subset G$ si el nou fet f pertany a G . Si la informació de SBC^B varia (i.e., o bé $E_{ant} \subset E$, o $G_{ant} \subset G$), les regles de reflexió ascendent ens asseguren que l'estat de BOLERO varia. És a dir, si abans del *meta-level lift up* BOLERO està en l'estat $\sigma_{BOLERO_{ant}}$, amb QE_{ant} i QG_{ant} , en el *meta-level lift up* BOLERO està en l'estat σ_{BOLERO} , amb QE i QG . En un *meta-level lift up* el pla de BOLERO no canvia, però sí les dades QE i QG que coneix sobre el nivell objecte.

BOLERO construeix un pla d'acord amb els fets QE i QG i, atès que després d'un *meta-level lift up* els conjunts QE i QG varien, una conseqüència immediata del *meta-level lift up* és que BOLERO pugui generar un pla nou. Aleshores, BOLERO canvia novament d'estat. Resumint, BOLERO pot canviar d'estat bé per un *meta-level lift up*, bé per la generació d'un pla nou com a conseqüència immediata del *meta-level lift up*. El canvi d'estat en BOLERO ho notem com segueix:

$$\sigma_{BOLERO} \xrightarrow{C} \sigma'_{BOLERO}$$

on

$$\sigma_{BOLERO} = (QE, QG, [p])$$

$$\sigma_{BOLERO} = (QE', QG', [p'])$$

i

$$QE \subseteq QE', QG \subseteq QG', [p] = [p'] \vee [p] \neq [p']$$

Meta-level reflect down. Quan BOLERO construeix un pla nou $[p']$, llavors es produeix un *meta-level reflect down* i, d'acord amb el que estableix la regla de reflexió descendent R_d , l'estat actual de SBC^B es modifica. En particular, l'aplicació de R_d modifica la seqüència d'objectius p que l'interpret de SBC^B està intentant demostrar, de manera que p és substituït pel nou pla p' que BOLERO imposa. D'aquesta manera el focus d'atenció del mecanisme d'inferència de SBC^B és alterat. Per exemple, si BOLERO construeix el pla [(pneumococ,

legione1.1a]], aleshores SBC^B comença a validar la seqüència d'objectius [pneumococ, legione1.1a], tot descartant el pla antic com a obsolet.

Anàlogament al *meta-level lift up*, un *meta-level reflect down* es pot notar com:

$$\Sigma_{BOLERO} \rightarrow \Sigma_{SBC}$$

on a tot estat $\sigma_{BOLERO} = (QE, QG, [p]) \in \Sigma_{BOLERO}$ li correspon un estat $\sigma_{SBC} = (E, D, G, p) \in \Sigma_{SBC}$. El pla $[p]$ generat per BOLERO és el pla p del nou estat σ_{SBC} de SBC^B . El *meta-level reflect down* causa, doncs, un canvi d'estat en la resolució de problemes en SBC^B .

3.4 El comportament reactiu de BOLERO-SBC

L'arquitectura metanivell BOLERO-SBC permet que BOLERO i SBC^B cooperin en la resolució de problemes. Durant la resolució d'un problema, la planificació i l'execució de plans s'alternen: la generació de plans d'actuació estratègics té lloc en el metanivell i l'execució dels plans en el nivell objecte. BOLERO pot interrompre en qualsevol moment el pla que està executant actualment SBC^B i manar un pla nou. L'alternació de la generació i l'execució de plans permet a BOLERO examinar l'entorn amb una certa freqüència i adequar els plans que genera als canvis que es produeixen. Així, l'execució de plans enriqueix la informació que BOLERO coneix sobre un problema i permet generar un pla més precís (adequat).

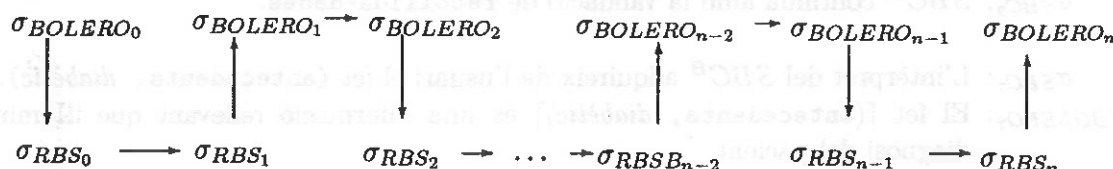


Figura 3.3: Seqüència de canvis de lloc d'acció en BOLERO-SBC.

En un moment donat, BOLERO-SBC està actiu en un dels dos nivells (en BOLERO o en SBC^B). El nivell actiu s'anomena *lloc d'acció* [158]. La seqüència de canvis de llocs d'acció que té lloc durant la resolució d'un problema es mostra a la figura 3.3. Quan un usuari vol resoldre un problema, el primer lloc d'acció és BOLERO. Notem amb σ_{BOLERO_0} l'estat inicial de BOLERO en la resolució d'un problema. En σ_{BOLERO_0} existeix un pla per defecte $[p_0]$. El *reflect down* de $[p_0]$ engega SBC^B ⁵. Representem per σ_{SBC_0} l'estat de SBC^B quan és activat per primera vegada. D'acord amb la regla de reflexió descendent R_d aplicada al pla per defecte $[p_0]$, SBC^B té un conjunt inicial d'objectius amb els quals començar el procés de raonament. El lloc d'acció és ara SBC^B , l'interpret del qual comença a validar el primer objectiu enregistrat a p en l'estat inicial σ_{SBC_0} . En el transcurs del seu procés de raonament SBC^B pot aconseguir, en un estat donat σ_{SBC_1} , un nou fet f (mitjançant l'adquisició d'un nou fet de l'usuari o per la deducció d'un fet objectiu). Aleshores, el nou fet $[f]$ és també conegut en BOLERO com ho garanteix les regles de reflexió ascendents R_{a_e} i R_{a_g} . Es produeix un *meta-level lift up* i el nou lloc d'acció és ara BOLERO en l'estat σ_{BOLERO_1} . Tots els fets

⁵ Aquest pla per defecte és coneixement de domini proporcionat a BOLERO per un mestre.

coneguts fins a l'estat σ_{BOLERO_1} (inclòs $[f]$) condueix a BOLERO en un nou estat σ_{BOLERO_2} a generar un nou pla. Aquest nou pla modifica la seqüència p d'objectius que SBC^B està intentant satisfer en σ_{SBC_1} en una nova seqüència en un nou estat σ_{SBC_2} (*meta-level reflect down*). SBC^B és en aquest moment el nou lloc d'acció. I així successivament, fins que BOLERO decideix terminar l'execució d'un problema (i.e. s'ha arribat a una solució).

Per il·lustrar amb un exemple la seqüència de canvis de llocs d'acció que s'esdevé en la resolució d'un problema, i per tant, la reacció als canvis en l'entorn, suposem que un usuari (presumiblement un metge) vol conèixer quin és el microbi que causa la pneumònia al pacient "r13v1". Les dades del pacient són: (edat, 79), (pneumònia-extrahospitalària, sí), (estat-del-pacient, moderadament-greu), (sexe, dona), (internat-en-asil, desconegut), (antecedents, diabètic). La seqüència de possible canvis d'acció és la següent⁶:

- σ_{BOLERO_0} : BOLERO genera el pla inicial $p_0 = [[\text{recollida-dades}]]$.
- σ_{SBC_0} : SBC^B comença a validar l'objectiu *recollida-dades* ($p = [\text{recollida-dades}]$).
Recollida-dades és un objectiu que consisteix a emplenar l'historial clínic del pacient, on es registren dades personals i patològiques d'interès.
- σ_{SBC_1} : En el transcurs del procés de raonament, SBC^B pregunta per l'edat del pacient a l'usuari, i s'obté el fet (edat, 70).
- σ_{BOLERO_1} : La nova informació $[(\text{edat}, [69, 100])]$ no és rellevant per a la construcció d'un pla, i.e. no activa en BOLERO més nodes dels que ja té activats.
- σ_{BOLERO_2} : BOLERO manté el mateix pla $[[\text{recollida-dades}]]$
- σ_{SBC_2} : SBC^B continua amb la validació de *recollida-dades*.
- ...
- σ_{SBC_7} : L'interpret del SBC^B adquireix de l'usuari el fet (antecedents, diabètic).
- σ_{BOLERO_7} : El fet $[(\text{antecedents}, \text{diabètic})]$ és una informació rellevant que illumina la diagnosi del pacient.
- σ_{BOLERO_8} : Conegut $[(\text{antecedents}, \text{diabètic})]$, juntament amb la resta de la informació que s'ha anat recollit des del començament de l'execució (i.e. $[(\text{edat}, [69, 100])]$, $[(\text{pneumònia-extrahospitalària}, \text{sí})]$, $[(\text{estat-del-pacient}, \text{lleu})]$, $[(\text{sexe}, \text{dona})]$, $[(\text{internat-en-asil}, \text{desconegut})]$), BOLERO construeix un nou pla: $[[\text{bact-atip}, \text{pneumococ}, \text{legionel·la}, \text{s-piògen}]]$.
- σ_{SBC_8} : L'interpret de SBC^B té una nova llista d'objectius a validar, $p = [\text{bact-atip}, \text{pneumococ}, \text{legionel·la}, \text{s-piògen}]$, i comença a perseguir *bact-atip*.

... I així successivament fins que en el n -èsim canvi de lloc d'acció, quan es coneixen els diagnòstics $[(\text{pneumococ}, \text{cert})]$ i $[(\text{legionel·la}, \text{moderadament-possible})]$, BOLERO (en l'estat σ_{BOLERO_n}) decideix aturar l'execució.

Cada vegada que es coneixen noves informacions (i.e. un fet d'entrada o un fet objectiu), es consulta el metanivell, que considera si cal generar un nou pla adaptat a les noves circumstàncies. Aleshores, si es genera un pla nou, el pla s'executa immediatament; altrament, es continua amb el pla actual. La integració de BOLERO i SBC^B permet, doncs, reaccionar als canvis en la informació, mitjançant l'alternació i l'execució de plans; per tant, el sistema BOLERO-SBC esdevé globalment un sistema reactiu.

⁶La seqüència de canvis d'estat correspon a l'exemple presentat a la secció 2.1.

3.5 El guany de la integració

L'arquitectura metanivell BOLERO-SBC mostra un comportament similar a altres arquitectures metanivell, com per exemple MILORD [142]. La particularitat de BOLERO-SBC envers d'altres arquitectures rau en el fet que els nivells de l'arquitectura són sistemes no uniformes. BOLERO és capaç d'aprendre i construir plans. SBC^B és capaç de resoldre problemes mitjançant un ús a cegues del seu coneixement (i.e. validar tots els objectius). Els dos sistemes són capaços de treballar separatament i poden ser desenvolupats i modificats autònomament. La integració dels sistemes en una arquitectura metanivell manté les característiques dels sistemes independents i permet guanyar-ne de noves.

En primer lloc ambdós sistemes cooperen en la resolució de problemes: BOLERO planifica la seqüència d'objectius que SBC^B ha de seguir, i així SBC^B evita realitzar una cerca exhaustiva de tots els objectius. El primer guany és, doncs, el millorament en l'eficiència de la resolució de problemes mitjançant la planificació estratègica dels objectius de SBC^B . En segon lloc, la construcció de plans en BOLERO s'alterna amb l'execució de plans en el SBC^B , de manera que nova informació obtinguda per l'execució d'un pla pot causar la re-consideració del pla actual, reaccionant als possibles canvis de l'entorn mitjançant l'elaboració d'un pla nou. El segon guany que resulta, doncs, de la integració és el comportament reactiu de BOLERO-SBC en la resolució de problemes. Cal remarcar que BOLERO no és un planificador reactiu per si mateix, sinó que és l'arquitectura metanivell en la qual BOLERO i SBC^B s'integren la que presenta un comportament reactiu en la resolució de problemes. En tercer lloc, BOLERO pot aprendre de la seva pròpia experiència. És a dir, quan un problema és resolt en BOLERO-SBC, s'obté un cas nou. El nou cas, com qualsevol altre cas de BOLERO, consisteix en tots els fets recollits en el transcurs de la resolució del problema més el pla final. Aleshores un nou guany de la integració és que BOLERO pot aprendre de l'experiència pròpia a partir d'aquest nou cas (vegeu secció 4.6). Finalment, i com una conseqüència de l'aprenentatge per experiència, BOLERO és capaç de construir en el futur plans que millorin l'eficiència del sistema BOLERO-SBC (*speed-up learning*).

Capítol 4

El sistema BOLERO

El propòsit d'aquest capítol és descriure els mètodes de l'aproximació a la planificació basada en casos de BOLERO: l'aprenentatge per observació, la construcció de plans (recuperació i adaptació), l'avaluació i l'aprenentatge per l'experiència (aprenentatge a partir de plans reeixits i plans imperfectes). Cadascun d'aquests mètodes s'ha implementat de manera independent constituint un mòdul dins del sistema BOLERO (vegeu la figura 4.1). L'aprenentatge per observació ha donat lloc a un component que interacciona amb el mestre i una memòria de plans. El mètode de construcció de plans (recuperació i adaptació) esdevé el mecanisme que interacciona amb *SBC^B*. El mètode d'avaluació de plans ha donat lloc a un component, l'avaluador, que interacciona amb una memòria de patrons d'avaluació i amb els plans generats. Finalment, l'aprenentatge per experiència pròpia esdevé un mòdul que interacciona amb el mestre, utilitza la base de patrons d'avaluació, fa ús del mòdul de l'aprenentatge per observació i modifica la memòria de plans.

Els mètodes de la planificació basada en casos de BOLERO determinen, doncs, la modularització del sistema. D'aquesta manera, quan es desitja que BOLERO aprengui plans, mitjançant casos eventualment proporcionats per un mestre, s'ha d'activar el mòdul de l'aprenentatge per observació. En una ocasió diferent, quan un usuari vol accedir al sistema per resoldre problemes, s'activa el mòdul de construcció de plans. Si és el mestre que està entrenant BOLERO, es pot combinar la construcció de plans, l'avaluador i fins i tot l'aprenentatge per experiència. Cada combinació dels mòduls (o variació de les característiques d'un mòdul) constitueix una *configuració* de BOLERO.

Les configuracions de BOLERO han estat utilitzades en l'avaluació experimental del sistema en el capítol 5. Però el propòsit d'aquest capítol no rau en els detalls d'implementació, sinó a descriure cadascun dels mètodes. Per aquesta raó especifiquem en primer lloc els coneixements que són representats en BOLERO per tal d'aprendre i que configuren la memòria de plans (secció 4.1). A continuació es descriu la visió global de la interacció entre els mòduls (secció 4.2). A la secció 4.3 es descriu el mètode d'aprenentatge per observació i a la secció 4.4 es detalla com té lloc la construcció de plans en BOLERO. Després, en la secció 4.5 es tracta el mètode d'avaluació de plans. I, finalment, en la secció 4.6 es descriu el mètode d'aprenentatge per l'experiència pròpia.

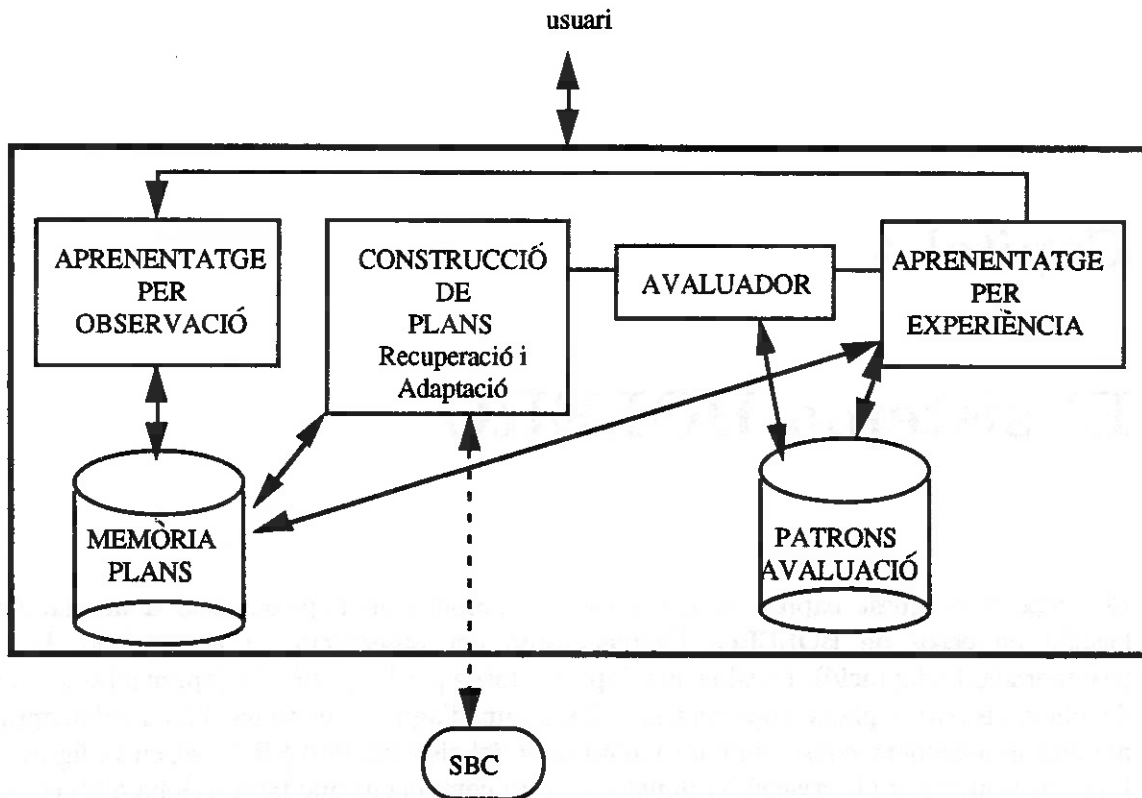


Figura 4.1: Esquema general de BOLERO.

4.1 Representació dels coneixements

Els coneixements de planificació estan representats en BOLERO mitjançant fets, situacions, accions, plans, episodis, traces i casos. Tots integren la memòria de plans a partir de la qual BOLERO pot resoldre problemes.

4.1.1 Fets

Les dades del món s'expressen per un conjunt d'esdeveniments F . Cada fet f de F és un parell $\langle id, v, \rangle$ on id és l'identificador, i v el valor. Un identificador pot ser qualsevol símbol, com ara *dades-raigs-X*. Quan parlem del fet *dades-raigs-X* ens estem referint a un fet de F l'identificador del qual és *dades-raigs-X*. El valor v ha de ser un valor plausible pel fet f en un domini donat. Els valors poden ser numèrics ($1, 0.5 \dots$), booleans (*si*, *no*), difusos (*possible*, *cert* ...), enumerats (*rovellós*, *purulent* ...), etc. El valor *desconegut* és assignable a qualsevol fet.

S'ha convingut que tots els valors siguin simbòlics. D'aquesta manera els fets que en el món prenen valors numèrics (fets numèrics), en BOLERO tenen assignat un interval. Per exemple, el fet *edat* pot prendre el valor $[0, 14]$. Els intervals d'un fet numèric no han de ser necessàriament disjunts. Així, per exemple el fet *frequència-cardíaca* pot prendre els valors $[80, 90]$ i $[0, 100]$ al mateix temps. Els intervals admissibles per a un fet numèric es defineixen per a cada aplicació. La representació d'un número x com el valor d'un fet f

s'aconsegueix mitjançant un procés de *normalització*, el qual fa correspondre a x un interval v que pertany al conjunt d'interval admissibles definits pel fet f .

Els fets amb valors difusos són comparables. És a dir, qualsevol fet difús pren valors dins d'un conjunt totalment ordenat d'etiquetes lingüístiques: (*impossible*, *poc-possible*, *lleugerament-possible*, ..., *molt-possible*, *cert*) [61]. Dos fets difusos amb el mateix identificador es poden comparar d'acord amb la posició del seu valor de certesa dins del conjunt d'etiquetes. Si v_1 és un valor difús d'un fet f_1 , v_2 el valor difús d'un fet f_2 amb el mateix identificador, i *valor* un predicat que, aplicat a un fet, retorna el seu valor, aleshores diem que

$$\text{valor}(\text{id}, v_1) \leq \text{valor}(\text{id}, v_2)$$

o bé simplement que

$$\text{valor}(\text{id}, v_1) \geq v_2$$

Per exemple, si tenim els fets (*pneumònia-pneumococ*, *possible*) i (*pneumònia-pneumococ*, *cert*)¹, podem dir que

$$\text{valor}(\text{pneumònia-pneumococ}, \text{possible}) \leq \text{valor}(\text{pneumònia-pneumococ}, \text{cert})$$

O simplement,

$$\text{valor}(\text{pneumònia-pneumococ}, \text{possible}) \leq \text{cert}$$

Si un fet pot tenir més d'un valor al mateix temps, direm que és un fet multivaluat, envers la resta de fets que són fets monovaluats, i el representarem com a $\langle \text{id}, V \rangle$, on s'ha substituït v per un conjunt de valors V . Per exemple, el fet *tos* és monovaluat ja que pot pendre el valor *si* o el valor *no* de forma exclusiva. El fet *malalties-associades* és multivaluat perquè pot prendre més d'un valor, com per exemple *aparell-respiratori* i *aparell-digestiu*.

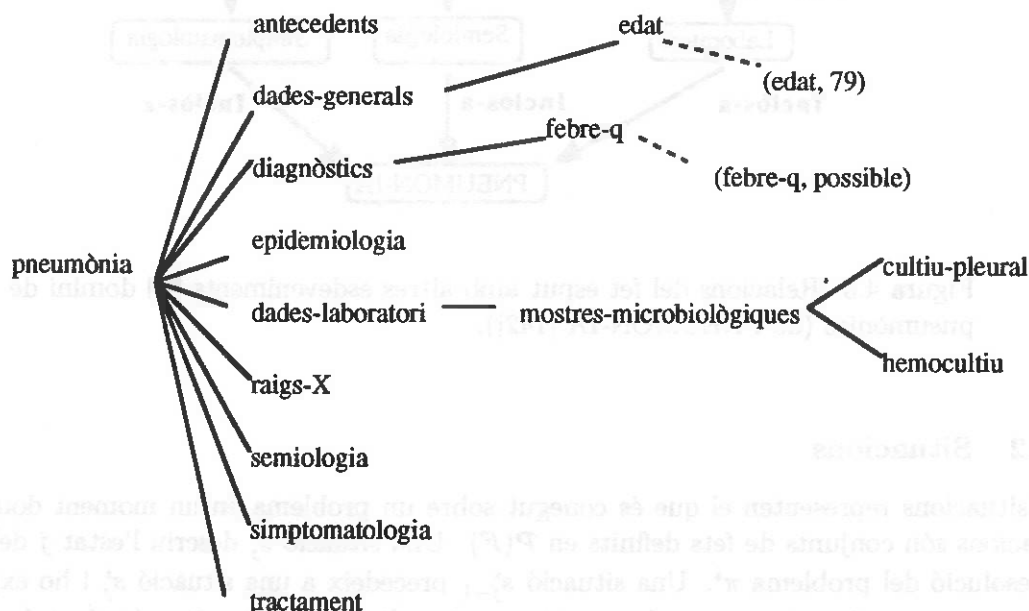


Figura 4.2: Relació d'inclusió entre les classes d'identificadors i identificadors de fets (línia llisa), i entre fets i identificadors (línia de guionets) en l'aplicació PNEUMON-IA [166].

¹Moltes vegades i per claredat simplifiquem la representació de fets com a parelles de (identificadors, valors), en lloc d'utilitzar el quartet descrit en la secció 4.1.1.

Els fets s'organitzen en una jerarquia de classes d'acord amb el seu identificador (figura 4.2). Tot fet f apunta al seu identificador amb un enllaç "és-un", i cada identificador pertany a una classe c . Per exemple, el fet (edat, 79) està enllaçat amb l'identificador **edat** que a la vegada pertany a la classe **dades-generals**, i el fet (febre-q, possible) està enllaçat al seu identificador **febre-q** que pertany a la classe **diagnòstics**. A més a més de la jerarquia de classes, els identificadors dels fets estableixen entre ells altres tipus de relació com ara, relacions d'especialització, de necessitat, etc. constituint una xarxa de coneixements. Per exemple, en la figura 4.3 es mostra una fracció de la xarxa que recull el fet *esput* amb la resta d'esdeveniments amb els quals està relacionat. La xarxa de coneixements és útil tant en l'aprenentatge per observació (veure la funció de generalització definida en la secció 4.3.2), com en l'activació de nodes en memòria, com s'il·lustra en l'apartat 4.4.1 d'aquest capítol.

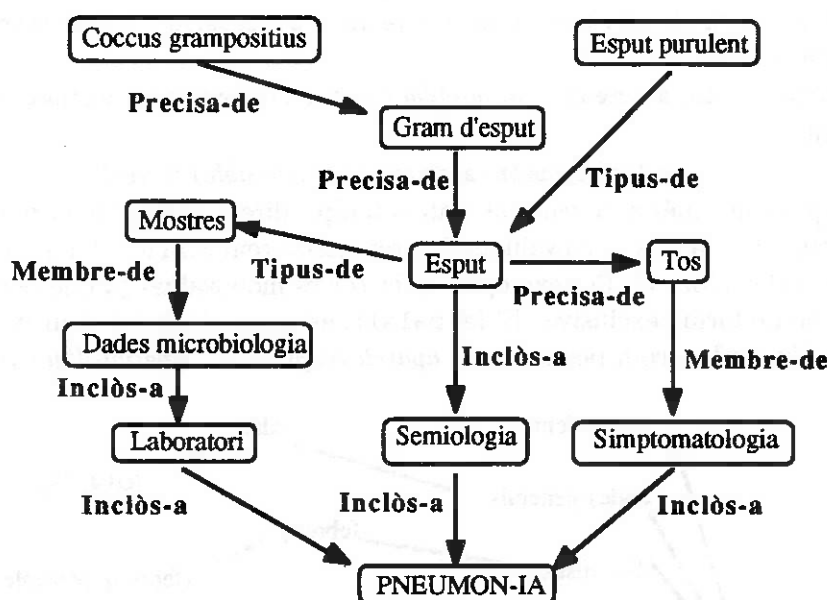


Figura 4.3: Relacions del fet esput amb altres esdeveniments del domini de pneumònies (de PNEUMON-IA [142]).

4.1.2 Situacions

Les situacions representen el que és conegut sobre un problema en un moment donat. Les situacions són conjunts de fets definits en $\mathcal{P}(F)$. Una situació s_j^i descriu l'estat j del procés de resolució del problema π^i . Una situació s_{j-1}^i precedeix a una situació s_j^i i ho expressem com a $s_{j-1}^i \prec s_j^i$. Les situacions són monòtones, és a dir, per a tota situació $s_j^i \prec s_k^i$, $s_j^i \subset s_k^i$.

4.1.3 Accions

Un problema π^i es formula mitjançant un estat o situació inicial s_0^i i una especificació d'estat final s_n^i al qual es desitja arribar (objectiu). Les accions permeten canviar d'un estat a un altre.

En BOLERO hem convingut les accions com a tasques. Una tasca focalitza el procés de resolució de problemes cap a l'assoliment d'un objectiu. És a dir, com hem vist en el capítol anterior, una acció en BOLERO significa que el SBC^B inicia el procés de deducció d'un objectiu. Per exemple, l'acció *legionel·la* en BOLERO és interpretada per SBC^B com la validació del objectiu *legionel·la*. Consegüentment, els efectes d'una acció no són coneguts *a priori* de l'execució d'un pla. Només quan SBC^B executa una acció (i.e. valida un objectiu) es poden conèixer els efectes de l'acció conjuntament amb altres esdeveniments que es puguin produir-se en l'entorn, com a conseqüència de les inferències que SBC^B realitza.

Anomenem A el conjunt de les accions a_i que coneix BOLERO, i $result(a_i) \subset \mathcal{P}(F)$ al conjunt de resultats obtinguts amb l'execució de a_i . Per exemple, l'acció *virus* pot tenir com a resultat en un cas concret

$$result(virus) = \{(virus, possible), (varicel·la, desconegut)\}$$

El resultats de totes les accions formen un subconjunt F_a dins del conjunt de fets F , de manera que $\cup_{a_i \in A} result(a_i) = F_a \subset F$. Com a conseqüència de la classe d'aplicacions que s'han tingut presents en el disseny de BOLERO (i.e. la diagnosi mèdica) hem assumit que qualsevol fet de F_a és un fet difús. És a dir, el resultat d'una acció són fets difusos, i en conseqüència, comparable amb altres fets difusos resultat d'una altra acció. La possibilitat de comparar el resultats d'accions diferents és una característica important que pot ser utilitzada per definir condicions de terminació com es detallarà a la secció 4.3.3.

Durant la resolució del problema π^i dins de BOLERO-SBC podem distingir entre dos tipus d'accions:

- *accions executades*: quan el SBC^B està intentant assolir un objectiu diem que l'acció corresponent s'està executant.
- *accions completades*: quan l'objectiu està consolidat (i.e el SBC^B dedueix l'objectiu), diem que l'acció s'ha completat.

Per a qualsevol acció completada a_j en un moment x tenim que $result(a_j) \subset s_x^{i,2}$. De la mateixa manera, una acció està executada però no completada si $result(a_j) \not\subset s_x^i$. Donat que les situacions són monòtones (si $s_x^i \prec s_{x+1}^i$, llavors $s_x^i \subset s_{x+1}^i$), els resultats de l'acció a_j es mantenen ($result(a_j) \subset s_y^i$ per a tota situació $s_x^i \prec s_y^i$).

Així, i com a conseqüència de l'alternació de la construcció i execució de plans en BOLERO-SBC, les accions executades poden ser de dos tipus:

- *Accions interrompudes*. El SBC^B , mentre valida un objectiu, pot recollir informació, amb la qual BOLERO genera un nou pla. Si el nou pla no inclou l'acció corresponent a l'objectiu que el SBC^B està deduint, la validació de l'objectiu s'interromp, i diem que l'acció s'interromp.
- *Accions continuades*. Si en una situació successiva es genera un nou pla que contingui una acció interrompuda a , es pot continuar l'acció a (i.e. el SBC^B continua la validació de l'objectiu corresponent).

Una altra forma de classificar les accions és en funció de la seva finalitat. Així, distingim els grup d'accions següents:

- *accions intermèdies*: accions necessàries perquè puguin executar-se altres accions (com per exemple *recollida-dades*).

²Si per alguna raó el SBC^B no pot aconseguir un determinat objectiu, suposem que almenys proporcionarà a BOLERO un mecanisme per fer-li-ho saber. Per exemple, si l'acció *micoplasma* és interpretada pel SBC^B com la validació de l'objectiu *micoplasma* i aquest no es pot satisfer, el SBC^B pot "deduir" el fet (*micoplasma, desconegut*) com a resultat de l'acció.

• *accions finals*: accions que poden produir la solució de problemes un cop completades. Amb aquest propòsit és possible distingir un subconjunt de F_a , que anomenem F_f (i.e. $F_f \subset F_a \subset F$), que correspon al resultat de les accions finals.

4.1.4 Plans

Un pla p_j^i és el pla construït en la situació s_j^i quan es resol el problema π^i . Un pla es defineix com una seqüència d'accions $p_j^i = [a_0^i, a_1^i, \dots, a_j^i]$, on $\mathcal{P}(A) = \Gamma$ és el conjunt de tots els possibles plans. Per exemple, [recollida-dades, bact-atip, pneumococ] és un pla i recollida-dades, bact-atip i pneumococ les seves accions. Diem que la longitud del pla p_j^i és j .

Notem la concatenació de plans amb l'operador $;$, el resultat del qual és un nou pla: $p_i^i = p_j^i; p_k^i$ on $p_j^i, p_k^i, p_i^i \in \Gamma$. En particular quan concatenem un pla p_j^i amb un altre pla p_k^i constituït per una sola acció (i.e. $p_k^i = [a_k^i]$) ho notarem per $p_j^i; a_k^i$. Direm que el pla p_j^i està inclòs dins del pla p_k^i , $p_j^i \subset p_k^i$, quan $p_k^i = p_j^i; p_l^i$. Així, un pla p_j^i inclòs en un altre pla p_k^i és més curt (i.e. $j < k$). Per exemple, el pla [recollida-dades, bact-atip] està contingut en el pla [recollida-dades, bact-atip, pneumococ, legionel·la]. Si suposem que tots els plans comencen per una acció determinada, com per exemple a , els plans poden ordenar-se jeràrquicament segons la relació d'inclusió com mostra la figura 4.4.

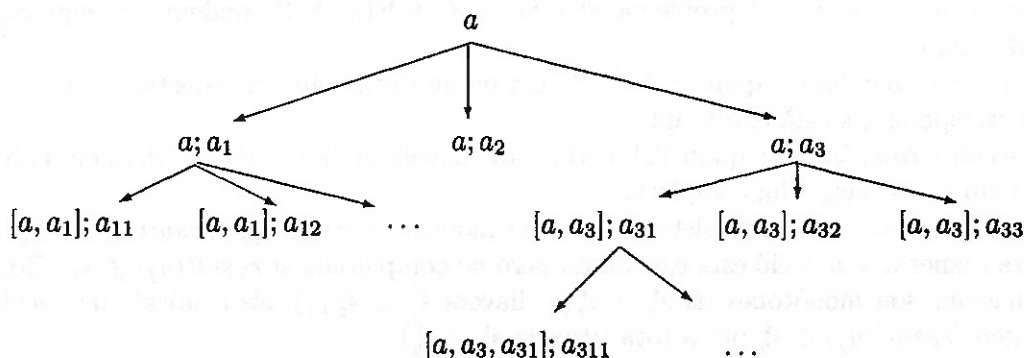


Figura 4.4: Jerarquia de plans.

En un moment donat de la resolució d'un problema podem distingir tres tipus de plans:

- pla immediat p_j^i : pla amb el qual prosseguir la resolució d'un problema des de la situació s_j^i .
- pla executat p_j^i : pla compost per totes les accions executades des de l'inici de la resolució d'un problema fins a la situació s_j^i .
- pla executat/immediat p_j^i : pla executat concatenat amb la següent acció amb la qual continuar la resolució d'un problema.

Així, si una acció a_k^i ha estat completada en s_j^i , aleshores $a_k^i \in p_j^i$ i $a_k^i \in \bar{p}_j^i$, però $a_j^i \notin p_j^i$. És a dir, no es repeteix l'execució de les accions, que per cert seria absurd, donat el caràcter no monòton de les situacions. En canvi, si una acció està executada però no completada, llavors $a_k^i \in p_j^i$ i $a_k^i \in \bar{p}_j^i$ i, eventualment pot ocórrer que $a_k^i \in p_j^i$.

Per donar un exemple de cada tipus de pla, suposem que un metge vol diagnosticar la pneumònia d'un pacient que acaba d'arribar a l'hospital. Les diferents situacions i diferents

s_0	=	{}
p_0	=	[recollida-dades]
s_1	=	$s_0 \cup \{(clínica, pneumònia-extrahospitalària)\}$
p_1	=	[recollida-dades]
s_2	=	$s_1 \cup \{(estat-malalt, greu)\}$
p_2	=	[recollida-dades]
...		
s_{10}	=	$s_9 \cup \{(antecedents, \{fumador, addicció-drogues-intravenoses\})\}$
p_{10}	=	[recollida-dades, bacteriana-atípica, estafilococ, pneumococ, enterobacteriàcies, tuberculosi, pseudomones, pneumocystis-car, criptococ, aspergil·losi]
s_{11}	=	$s_{10} \cup \{(dades-malalt, cert)\}$
p_{11}	=	[bacteriana-atípica, estafilococ, pneumococ, enterobacteriàcies, tuberculosi, pseudomones, pneumocystis-car, criptococ, aspergil·losi]
...		
s_{20}	=	$s_{19} \cup \{(dades-raigs-X, embassament)\}$
p_{20}	=	[bacteriana-atípica, estafilococ, pneumococ, enterobacteriàcies, tuberculosi, pseudomones, pneumocistii-car, critptococ, aspergil·losi]
s_{21}	=	$s_{20} \cup \{(mostra-líquid-pleural, cert)\}$
p_{21}	=	[embassament, bacteriana-atípica, estafilococ, pneumococ, enterobacteriàcies, tuberculosi, pseudomones, pneumocystis-car, criptococ, aspergil·losi]
s_{22}	=	$s_{21} \cup \{(líquid-exsudat, desconegut)\}$
p_{22}	=	[embassament, bacteriana-atípica, estafilococ, pneumococ, enterobacteriàcies, tuberculosi, pseudomones, pneumocystis-car, criptococ, aspergil·losi]
s_{23}	=	$s_{22} \cup \{(gèrmens-líquid-pleural, fals)\}$
p_{23}	=	[bacteriana-atípica, estafilococ, pneumococ, enterobacteriàcies, tuberculosi, pseudomones, pneumocystis-car, criptococ, aspergil·losi]
s_{24}	=	$s_{23} \cup \{(pneumònia-bacteriana, força-possible)\}$
p_{24}	=	[estafilococ, pneumococ, enterobacteriàcies, tuberculosi, pseudomones, pneumocystis-car, criptococ, aspergil·losi, legionella, anaerobi, hemòfil, estreptococ-piògen, meningococ, branhamella]
...		
s_{99}	=	$s_{98} \cup \{(contacte-tuberculosos, no)\}$
p_{99}	=	[tuberculosi, pseudomones, pneumocystis-car, criptococ, aspergil·losi, legionella, anaerobi, hemòfil, estreptococ-piògen, meningococ, branhamella]
s_{100}	=	$s_{99} \cup \{(tuberculosi, desconegut)\}$
p_{100}	=	[□]

Figura 4.5: Exemples de plans immediats basats en la diagnosi del pacient m15v1r

plans amb què el metge es troba durant la diagnosi estan representats en la figura 4.5. La situació inicial s_0 representa l'inici de la sessió entre el metge i el malalt, quan el metge no coneix cap dada sobre la malaltia que pateix el pacient. La situació s_{100} és la situació a què arriba el metge quan coneix el microbi que causa la pneumònia al pacient. Entre una situació s_x i situació s_{x+1} s'assumeix que s'executa la primera acció del pla immediat p_x representat explícitament en la figura 4.5. Aleshores:

1. El pla p_0 és un pla immediat alhora que un pla executat/immediat $\bar{p}_0$ en la situació s_0 ($p_0 = \bar{p}_0 = [\text{recollida-dades}]$). El pla executat $\dot{p}_0$ és [], car encara no s'ha executat cap acció.
2. El pla immediat $p_1 = [\text{recollida-dades}]$ no varia i continua essent idèntic al pla executat/immediat $\bar{p}_0$. Atès que entre s_0 i s_1 s'ha iniciat l'execució del pla p_0 , ja existeix una acció executada, *recollida-dades*, i el pla executat és $\dot{p}_0 = [\text{recollida-dades}]$.
3. En les situacions següents els plans immediats, executats, i executats/immediat no canvien.

4. En la situació s_{10} , però, el fet (*antecedents*, {*fumador*, *addicció-drogues-intravenoses*}) suggereix al metge de considerar el pla immediat:

$p_{10} = [\text{recollida-dades, bacteriana-atípica, estafilococ, pneumococ, enterobacteriàcies, tuberculosi, pseudomones, pneumocystis-carinii, criptococ, aspergil·losi}]$

que és un pla nou (i.e. $p_{10} \neq p_9 = \dots = p_0$). El pla executat continua essent el mateix: $\dot{p}_{10} = \dot{p}_1 = [\text{recollida-dades}]$. El pla executat/immediat no canvia car la primera acció de p_{10} no varia respecte a p_0 ($\bar{p}_{10} = \bar{p}_0 = [\text{recollida-dades}]$).

5. En la situació s_{11} l'acció *recollida-dades* ha estat completada. Aleshores el pla immediat canvia, car la primera acció (*recollida-dades*) ja ha estat realitzada i, per tant,

$p_{11} = [\text{bacteriana-atípica, estafilococ, pneumococ, enterobacteriàcies, tuberculosi, pseudomones, pneumocystis-carinii, criptococ, aspergil·losi}]$

és a dir, $p_{10} = \text{recollida-dades}; p_{11}$. Conseqüentment, el pla executat/immediat canvia, $\bar{p}_{11} = [\text{recollida-dades, bacteriana-atípica}]$ (i.e. $\bar{p}_{11} = \bar{p}_{10}; \text{bacteriana-atípica}$). El pla executat, però, no canvia: $\dot{p}_{11} = \dot{p}_{10} = [\text{recollida-dades}]$, car l'acció *bacteriana-atípica* encara no ha començat a executar-se.

6. En la situació s_{12} el pla executat canvia:

$\dot{p}_{12} = [\text{recollida-dades, bacteriana-atípica}]$

puix que l'acció *bacteriana-atípica* ja és una acció executada ($\dot{p}_{12} = \dot{p}_{11}; \text{bacteriana-atípica} = \bar{p}_{11}$).

Observem que si el pla immediat és $p_j^i = [a_0^i, a_1^i, \dots, a_j^i]$, aleshores el pla executat/immediat és $\bar{p}_j^i = \dot{p}_j^i; a_0^i$; i també que el pla executat/immediat en s_j^i esdevé el pla executat en s_{j+1}^i . De manera que es poden definir les següents regles que arbitren els canvis de plans:

R1 : Si el pla immediat no canvia, tampoc canvien ni el pla executat ni el pla executat/immediat.

$$p_i = p_{i-1} \Rightarrow \dot{p}_i = \dot{p}_{i-1}, \bar{p}_i = \bar{p}_{i-1}$$

R2 : Si el pla immediat canvia, el pla executat/immediat canvia. Si el nou pla immediat és $p_i = [a_{i_1}, \dots, a_{i_n}]$, i el pla immediat anterior $p_i = [a_{j_1}, \dots, a_{j_n}]$ amb $a_{i_1} \neq a_{j_1}$, aleshores el nou pla executat/immediat és:

$$\bar{p}_i = \bar{p}_{i-1}; a_1$$

Si $a_{i_1} = a_{j_1}$, el pla executat/immediat no canvia.

R3 : Si el pla immediat/executat canvia en un moment (o situació) i , el pla executat canvia en un moment posterior $i + 1$. Si el pla immediat en i és $\bar{p}_i = [a_1, \dots, a_n]$, aleshores:

$$\dot{p}_{i+1} = \dot{p}_i; a_n$$

Seguint aquestes regles, trobem que:

$p_{20} = p_{19}$	$\dot{p}_{20} = \dot{p}_{19}$	$\bar{p}_{20} = \bar{p}_{19}$
$p_{21} \neq p_{20}$	$\dot{p}_{21} = \dot{p}_{20}$	$\bar{p}_{21} = \bar{p}_{20}; \text{embassament}$
$p_{22} = p_{21}$	$\dot{p}_{22} = \dot{p}_{21}; \text{embassament}$	$\bar{p}_{22} = \bar{p}_{21}$
$p_{22} = \text{embassament}; p_{23}$	$\dot{p}_{23} = \dot{p}_{22}$	$\bar{p}_{23} = \bar{p}_{22}; \text{bact-atip}$
$p_{23} = \text{bact-atip}; p_{24}$	$\dot{p}_{24} = \dot{p}_{23}; \text{bact-atip}$	$\bar{p}_{24} = \bar{p}_{23}; \text{estafilococ}$

Observem que en la darrera situació, s_{100} , el metge finalitza la diagnosi del pacient sense completar el pla p_{99} , i tenim que:

$$\begin{aligned} p_{100} &= [\square] \\ \dot{p}_{100} &= [\text{recollida-dades, bacteriana-atípica, embassament, bacteriana-atípica,} \\ &\quad \text{estafilococ, \dots, tuberculosi}] \\ \bar{p}_{100} &= \bar{p}_{100} \end{aligned}$$

4.1.5 Episodis

Un episodi és un parell $\langle s_j^i, \bar{p}_j^i \rangle$ constituït per la situació s_j^i i el pla executat/immediat $\bar{p}_j^i$. Un episodi, doncs, associa a una situació s_j^i el pla executat fins a s_j^i i la propera acció a realitzar. Un episodi δ_j^i precedeix un altre δ_{j+1}^i , i ho notem per $\delta_j^i \prec \delta_{j+1}^i$. Si δ_k^i precedeix δ_{j+1}^i , aleshores la situació s_j^i de δ_j^i precedeix la situació s_{j+1}^i , on $|s_{j+1}^i| \geq |s_j^i| + 1$. El pla de δ_{j+1}^i conté el pla del seu predecessor directe δ_j^i , de manera que $\bar{p}_{j+1}^i = \bar{p}_j^i; a_k^j$. Donada la relació d'inclusió entre situacions i plans d'episodis, també es dona la relació d'inclusió entre episodis: $\delta_0^i \subset \delta_1^i \subset \dots \subset \delta_{n_i}^i$.

4.1.6 Traces

Una traça $T(\pi^i)$ és la seqüència d'episodis $[\delta_0^i, \delta_1^i, \dots, \delta_{n_i}^i]$ seguits en el procés de resolució del problema π^i . En una traça distingim els *plans parcials* dels *plans finals*. Un pla parcial és qualsevol pla $\bar{p}_j^i$ d'un episodi δ_j^i on $j \neq n_i$, i un pla final és el pla $\bar{p}_{n_i}^i$ del darrer episodi d'una traça. El pla final $\bar{p}_{n_i}^i$ és la solució del problema π^i o el pla que condueix de la situació inicial s_0^i a la situació final o objectiu $s_{n_i}^i$.

4.1.7 Casos

Un cas C^i és la racionalització de la seqüència d'episodis (i.e. traça) que té lloc en la resolució d'un problema.

$$C^i = R(T(\pi^i))$$

La funció racionalització R és, en certa manera, una forma de pre-adaptació del casos que es realitza per dur a terme una millor re-utilització dels casos en la resolució de problemes nous. Si Δ és el conjunt de totes les traces, definim la *racionalització* d'una traça com la funció $R: \Delta \rightarrow \Delta$, on es produeix una reordenació de les accions. La raó és la següent:

$\delta_0:$	$s_0 = \{ \}$
	$\bar{p}_0 = [\text{recollida-dades}]$
$\delta_1:$	$s_1 = \{ (\text{clínica, pneumònia-extrahospitalària}), (\text{estat-malalt, moderat}), (\text{forma-instauració, brusca}) \}$
	$\bar{p}_1 = [\text{recollida-dades, bact-atip}]$
$\delta_2:$	$s_2 = s_1 \cup \{ (\text{antecedents, immunodepressió}), (\text{tipus-immunodepressió, càncer-avançat, tractament-citostàtics}), (\text{dades-RX, embassament-pleural}), (\text{dispnea, no}), (\text{mostra-líquid-pleural, si}) \}$
	$\bar{p}_2 = [\text{recollida-dades, bact-atip, embassament}]$
$\delta_3:$	$s_3 = s_2 \cup \{ (\text{exsudat-pleural, desconegut}), (\text{efectuada-tinció-pleural, si}), (\text{gèrmens-tinc-pl, si}), (\text{empiema, cert}), (\text{germen-ident-pl, desconegut}), (\text{tinció-gram-pl, cocs-grampositius-parallelles}), \dots \}$
	$\bar{p}_3 = [\text{recollida-dades, bact-atip, embassament, bact-atip}]$
$\delta_4:$	$s_4 = s_3 \cup \{ (\text{pneumònia-bacteriana, cert}), (\text{mal-de-cap, desconegut}), (\text{pneumònia-atípica, desconegut}) \}$
	$\bar{p}_4 = [\text{recollida-dades, bact-atip, embassament, bact-atip, pneumococ}]$
	$\dots$
$\delta_9:$	$s_9 = s_8 \cup \{ (\text{pneumònia-pseudomones, desconegut}) \}$
	$\bar{p}_9 = [\text{recollida-dades, bact-atip, embassament, bact-atip, pneumococ, enterobacteriàcies, legionella, tuberculosi, pseudomones, } \square]$
(a)	
$\delta_0:$	$s_0 = \{ \}$
	$\bar{p}_0 = [\text{recollida-dades}]$
$\delta_1:$	$s_1 = \{ (\text{clínica, pneumònia-extrahospitalària}), (\text{estat-malalt, moderat}), (\text{forma-instauració, brusca}), (\text{antecedents, immunodepressió}), (\text{tipus-immunodepressió, càncer-avançat, tractament-citostàtics}), (\text{dades-RX, embassament-pleural}), (\text{dispnea, no}), (\text{mostra-líquid-pleural, si}) \}$
	$\bar{p}_1 = [\text{recollida-dades, embassament}]$
$\delta_2:$	$s_2 = s_1 \cup \{ (\text{exsudat-pleural, desconegut}), (\text{efectuada-tinció-pleural, si}), (\text{gèrmens-tinc-pl, si}), (\text{empiema, cert}), (\text{germen-ident-pl, desconegut}), (\text{tinció-gram-pl, cocs-grampositius-parallelles}), \dots \}$
	$\bar{p}_2 = [\text{recollida-dades, embassament, bact-atip}]$
$\delta_3:$	$s_3 = s_2 \cup \{ (\text{pneumònia-bacteriana, cert}), (\text{mal-de-cap, desconegut}), (\text{pneumònia-atípica, desconegut}) \}$
	$\bar{p}_3 = [\text{recollida-dades, embassament, bact-atip, pneumococ}]$
	$\dots$
$\delta_8:$	$s_8 = s_7 \cup \{ (\text{pneumònia-pseudomones, desconegut}) \}$
	$\bar{p}_8 = [\text{recollida-dades, embassament, bact-atip, pneumococ, enterobacteriàcies, legionella, tuberculosi, pseudomones, } \square]$
(b)	

Figura 4.6: (a) Traça del procés de diagnòstic del pacient g22v1. (b) Cas g22v1 (racionalització de la traça del procés de diagnòstic del pacient g22v1). Observem com a la traça (a) l'acció bact-atip ha estat executada en l'episodi δ_1 , interrompuda en l'episodi δ_2 i continuada en l'episodi δ_3 . En el pla final $\bar{p}_9$ de la traça, l'acció bact-atip figura dues vegades. Mentre que, després d'aplicar el procés de racionalització, el pla $\bar{p}_8$ del cas (b), l'acció bact-atip només apareix una vegada.

una acció a_j^i pot haver estat començada, suspesa i, després de l'aplicació d'altres accions, continuada (veure secció 4.1.3). Per exemple, en la figura 4.6a mostra la traça del procés de diagnosi d'una pneumònia en el pacient g22v1. En l'episodi δ_2 s'estava executant l'acció **bact-atip** quan el fet **mostra-líquid-pleural** ha causat un canvi de pla. En l'episodi δ_3 l'acció **bact-atip** ha estat suspesa, i una nova acció **embassament** ha estat començada. Quan l'acció **embassament** ha finalitzat amb l'obtenció dels resultats **empiema**, **germen-ident-pl**, ..., s'ha reprès l'execució de **bact-atip**.

La interrupció i continuació d'accions fa possible que en el pla final p_n^i d'una traça pugui aparèixer a_j^i en més d'una ocasió. La funció R racionalitza l'ordre de les accions per tal d'evitar cicles com començar-suspènre-continuar en l'execució de les accions. Si no es realitzés un procés de racionalització de les traces abans d'emmagatzemar els casos en memòria, en la resolució de problemes s'hauria de tenir present els cicles de començar-suspènre-continuar en l'adaptació de casos recuperats de memòria, per tal de no repetir-los. La racionalització representa, doncs, un avanç del procés d'adaptació que, en no fer-ho en el moment de l'emmagatzematge, hauria de fer-se en reusar el cas en una situació futura.

La racionalització és aplicable en una traça $T(\pi^i)$ quan un episodi δ_j^i que precedeix un altre episodi δ_k^i , i satisfà que

$$\begin{aligned} \delta_j^i &= \langle s_j^i, \bar{p}_j \rangle & \bar{p}_j &= \bar{p}_{j-1}^i; a_x^i \\ \delta_{j+1}^i &= \langle s_{j+1}^i, \bar{p}_{j+1}^i \rangle & \bar{p}_{j+1}^i &= \bar{p}_{j-1}^i; a_x^i; a_{x+1}^i \\ \delta_k^i &= \langle s_k^i, \bar{p}_k \rangle & \bar{p}_k &= \bar{p}_{j-1}^i; a_x^i; a_{x+1}^i; \bar{p}_l^i; a_x^i \end{aligned}$$

Aleshores la racionalització consisteix a suprimir l'episodi δ_j^i i transformar els episodis δ_{j+1}^i i δ_k^i com a continuació s'indica (l'algorisme es mostra en la taula 4.1):

$$\begin{aligned} \delta_{j+1}^i &= \langle s_{j+1}^i, \bar{p}_{j+1}'^i \rangle & \bar{p}_{j+1}'^i &= \bar{p}_{j-1}^i; a_{x+1}^i \\ \delta_k^i &= \langle s_k^i, \bar{p}_k' \rangle & \bar{p}_k' &= \bar{p}_{j-1}^i; a_{x+1}^i; \bar{p}_l^i; a_x^i \end{aligned}$$

D'aquesta manera s'indica que l'acció a_x^i és convenient executar-la només després de la realització del pla $\bar{p}_{j-1}^i; a_{x+1}^i; \bar{p}_l^i$. La figura 4.6b mostra el resultat d'aplicar la funció de racionalització a la traça de la solució de g22v1, esdevenint el cas "g22v1".

racionalització(T)

; $T = [\delta_0, \delta_1, \dots, \delta_n]$

1. $i = 0$

2. Mentre $i < n$

3. $a = \text{darrera-acció}(\bar{p}_i)$

4. Si existeix $\bar{p}_k \in T$, $k > i$ tal que $a \in \bar{p}_k$

5. Llavors $T = T - \delta_i$

6. $\bar{p}_i = \bar{p}_i - a$

7. $i = i + 1$

Taula 4.1: Algorisme de racionalització d'una traça.

El pla de l'episodi j d'un cas C^i té, doncs, j accions diferents. Dos plans p_x^i i p_y^j de dos casos diferents C^i i C^j poden ser iguals (i.e. $p_x^i = p_y^j$) i les situacions dels seus episodis corresponents diferents (i.e. $s_x^i \neq s_y^j$), encara que és d'esperar que $s_x^i \cap s_y^j \neq \emptyset$. Per exemple, els casos b02v1 i b05v1 tenen el mateix pla final [recollida-dades, bacteriana-atípica, pneumococ, legionel·la, enterobacteriàcies, anaerobi, hemòfil, estafilococ, s-piògen, meningococ, branhamel·la, tuberculosi]. Ara bé, les situacions finals divergeixen en els fets següents: mentre que en la situació final del cas

b02v1 apareixen els fets (*antecedents*, *fumador*), (*dades-RX*, *{v-maldef, confluent}*), i (*pneumònia-bacteriana*, *força-possible*), en el cas b05v1 apareixen els fets (*antecedents*, *desconegut*), (*dades-RX*, *{patró-intersticial}*), i (*pneumònia-bacteriana*, *moderadament-possible*). També és lògic suposar que si dos situacions de casos diferents són iguals, siguin també iguals el pla de l'episodi a què corresponen. Altrament no tenim manera de caracteritzar els plans i l'aprenentatge no seria viable. Ens mancarien coneixements de domini o més informació en els casos per tal de distingir les situacions en què han tingut lloc cada pla.

És interessant distingir la solució de domini d'un cas de la solució d'un cas des del punt de vista de la planificació. Per a un planificador la solució d'un cas és el pla final, i.e. $\bar{p}_n^i$. La solució del domini S_{C^i} d'un cas C^i és el resultat de totes les accions del pla final $\bar{p}_n^i$ que pertanyen a F_f (resultats d'accions finals):

$$S_{C^i} = \left(\bigcup_{a_k \in \bar{p}_n^i} \text{result}(a_k) \right) \cap F_f$$

Per exemple, les accions del pla final del cas g22v1 representat a la figura 4.6b donen els resultats següents:

recollida-dades	→ { (<i>dades-malalt</i> , <i>cert</i>) }
embassament	→ { (<i>empiema</i> , <i>cert</i>), (<i>germen-ident-pl</i> , <i>desconegut</i>) }
bact-atip	→ { (<i>pneumònia-bacteriana</i> , <i>cert</i>), (<i>pneumònia-atípica</i> , <i>desconegut</i>) }
pneumococ	→ { (<i>pneumònia-pneumococ</i> , <i>possible</i>) }
enterobacteriàcies	→ { (<i>pneumònia-enterobacteriàcies</i> , <i>moderadament-possible</i>) }
legionel·la	→ { (<i>pneumònia-legionel·la</i> , <i>força-possible</i>) }
tuberculosi	→ { (<i>pneumònia-tuberculosi</i> , <i>poc-possible</i>) }
pseudomones	→ { (<i>pneumònia-pseudomones</i> , <i>desconegut</i>) }

Sabent que les accions pneumococ, enterobacteriàcies, legionel·la, tuberculosi, i pseudomones són accions finals, la solució de domini de g22v1 és:

$$S_{r22v1} = \{ (\text{pneumònia-pneumococ}, \text{possible}), \\ (\text{pneumònia-enterobacteriàcies}, \text{moderadament-possible}), \\ (\text{pneumònia-legionel·la}, \text{força-possible}), \\ (\text{pneumònia-tuberculosi}, \text{poc-possible}), \\ (\text{pneumònia-pseudomones}, \text{desconegut}) \}$$

Les solucions del domini dels casos són útils per aprendre, com veurem en la secció 4.3.3, les condicions de terminació.

4.1.8 Memòria de plans

La memòria de plans és una memòria que conté tots els plans dels casos amb què BOLERO aprèn. La memòria de plans s'organitza en base a la seqüència d'episodis que compon un cas. Si suposem que tot episodi inicial δ_0^i té el mateix pla (i.e. $\delta_0^i = \langle s_0^i, p \rangle$), aleshores, donada la relació d'inclusió entre els episodis de les traces, es pot pensar en una organització jeràrquica dels casos. Nodes no terminals tenen associats plans parcials i informació sobre situacions corresponents a episodis δ_j de casos, on $j \neq n$, mentre que nodes terminals tenen plans finals i informació sobre els darrers episodis dels casos δ_n .

Un node N^i té associat un triplet $\langle v^i, \bar{p}^i, \tau^i \rangle$ compost per un pla parcial $\bar{p}^i$, una *situació generalitzada* v^i i la tipicalitat τ^i del node. La situació generalitzada v^i és el resultat d'un procés de generalització de les situacions s de tots els episodis δ_j^k que presenten el pla $\bar{p}^i$ (vegeu

propera secció 4.3.2). Les situacions generalitzades estan compostes de generalitzacions, i.e. $v^i = \{g_j^i\}$. Cada generalització g_j^i és una tupla $\langle id_j^i, v_j^i, w_j^i, \sigma_j^i \rangle$ on id_j^i és un identificador, v_j^i un valor o conjunt de valors, w_j^i la importància estratègica de la generalització i σ_j^i la freqüència d'aparició de la generalització en situacions generalitzades de nodes terminals descendents del node N^i .

La importància estratègica d'una generalització és un valor numèric definit en l'interval $[0,1]$. Com hem introduït en el capítol anterior i veurem en les seccions següents, els fets serveixen per indexar les generalitzacions i, consegüentment, recordar el pla del node en el qual estan emmagatzemades. Aleshores, la importància estratègica representa el pes (ponderació) amb què s'indexa el pla a partir de la generalització. Una generalització amb importància estratègica 0 és una generalització completament irrellevant en la construcció del pla del node en que està emmagatzemada; mentre que una generalització amb importància estratègica 1 representa una informació determinadament rellevant. La importància estratègica de les generalitzacions és, d'alguna manera, la substitució de la freqüència en altres mecanismes d'aprenentatge, com COWEB [55] i LAMDA [122]. El mecanisme d'indexació d'aquest sistemes confien molt en la freqüència de les generalitzacions, però en el domini de medicina en el qual hem aplicat BOLERO hem descobert que una generalització amb una freqüència σ molt petita pot ser força rellevant³. Per exemple, en la diagnòsi mèdica, els antecedents d'un pacient (*fumador*, *drogoaddicte*, etc) és un fet molt freqüent (es coneix el 86,07% de les vegades), mentre que un hemocultiu es realitza només un 3,79% de les vegades. Ara bé, en el moment de realitzar una diagnòsi, el metge disposa més confiança en les dades que resulten de l'hemocultiu que en els antecedents del pacient. Les dades més específiques tendeixen a ser les més rellevants, malgrat que no siguin disponibles amb tanta freqüència. A més a més, en una aplicació del món real (com per exemple en la diagnòsi mèdica), el domini és molt ampli, no existeix una sistemàtica en l'aparició de les dades i és necessari disposar d'un nombre molt gran de casos per poder establir una confiança en la freqüència de les generalitzacions amb un acceptable marge d'error.

La importància estratègica de les generalitzacions és coneixement de domini proporcionat per l'expert-mestre. Per dur a terme l'assignació de la importància estratègica de les generalitzacions s'ha utilitzat la relació establerta entre els identificadors de les generalitzacions en la xarxa de coneixements. Cada generalització té un identificador, i cada identificador pertany a una classe. Aleshores, es pot fer ús de l'agrupament dels identificadors en classes per facilitar l'assignació de la importància estratègica. Per exemple, totes les generalitzacions amb identificadors que pertanyen a la classe *dades-generals* tenen una importància estratègica inicial 0.1, mentre que les generalitzacions que pertanyen a la classe *mostra-microbiològiques* tenen una importància estratègica inicial 0.9. La importància estratègica assignada per defecte a una generalització pot ser corregida en l'aprenentatge per l'experiència pròpia, com es veurà en la secció 4.6.3.

La tipicitat τ^i d'un node N^i és un nombre natural que indica el nombre de vegades que el pla $\tilde{p}^i$ ha aparegut en els casos. El pla del node N^i té una acció més que el pla del node pare N^{i-1} i una acció menys que el pla del seus fills N^j , de manera que

$$\tilde{p}^j = \tilde{p}^i; a_k^j$$

on lògicament a_k^j és diferent per a cada successor de N^i . La inclusió dels plans entre els nodes respon a la inclusió que hi ha entre els plans de dos episodis veïns δ_j^i i δ_{j+1}^i . De la mateixa

³Descobriment que també s'ha donat en el desenvolupament d'EUREKA [52].

manera, s'estableix una relació d'inclusió entre les situacions generalitzades entre un node i les situacions generalitzades dels seus successors. La situació v^i d'un node N^i és més general que la situació v^j d'un successor N^j de N^i .

Un *node terminal* N_t^i , a més a més de contenir la informació pròpia d'un node qualsevol, té associat un conjunt de solucions S^i , $N_t^i = \langle v^i, \bar{p}^i, \tau^i, S^i \rangle$. Al conjunt de solucions S^i pertanyen les solucions de domini de tots els casos que tenen $\bar{p}^i$ com a pla final: $S^i = \bigcup_j S_{Cj}^i$. El pla d'un node terminal correspon al pla final d'un o més casos.

◇

Cal remarcar que la implementació de la jerarquia pot fer-se mitjançant la utilització de mecanismes d'herència. D'aquesta manera, un node N^i està representat per un triplet $\langle \theta^i, a^i, \tau^i \rangle$. θ^i conté les generalitzacions de v^i i no conté cap generalització ja representada en les situacions generalitzades v^j dels nodes antecessors de N^i . D'aquesta manera, v^i es pot calcular com a $\bigsqcup_{j=0}^i \theta^j$, on

$$\begin{aligned} \bigsqcup : \forall f \\ \text{if } \exists g' \in v^i \\ \text{then } v^i = (v^i \setminus g') \cup \{g\} \\ \text{else } v^i = v^i \cup \{g\} \end{aligned}$$

L'herència es pot implementar també en els plans dels nodes. A cada node s'associa només l'acció que el diferencia del seu antecessor directe⁴. Aleshores $\bar{p}^i$ és el resultat de la concatenació de les accions dels antecessors de N^i des de l'arrel de l'arbre. És a dir, $\bar{p}^i = [a^0, \dots, a_i]$, on a^0 és l'acció del node arrel N^0 , i a^i l'acció del node N^i . Malgrat que la implementació de BOLERO s'ha realitzat utilitzant mecanismes d'herència, per qüestions de simplicitat, en aquest capítol s'expliquen els mètodes basats en una representació extensiva de la memòria (i.e. sense entrar en detalls d'implementació dels mecanismes d'herència que compliquen sovint els algorismes).

4.2 Visió global

Existeixen diferents modes de treball en BOLERO: "aprenentatge-per-observació", "resoldre problemes", "avaluar", "aprenentatge-per-experiència", i és l'usuari qui escull el mode en el qual vol treballar (veure taula 4.2). És evident que quan l'usuari selecciona l'opció "aprenentatge-per-observació", primer s'obté el cas que es vol incorporar i, a continuació, s'activa el mètode d'incorporació d'un cas en memòria (taula 4.4). El mateix passa amb l'opció "avaluar": primer s'obté el cas que es vol avaluar, després el patró d'or i finalment s'activa el mètode d'avaluació de plans. L'opció "aprenentatge-per-experiència" encadena tres mètodes: en primer lloc BOLERO resol el problema, després l'avalua i finalment utilitza el mètode d'aprenentatge per experiència.

L'opció "resoldre problema" activa el mètode de resoldre problemes que volem remarcar, que és diferent que activar el mètode de construcció d'un pla (vegeu algorisme en la taula 4.3). Resoldre un problema π^i en BOLERO-SBC significa, com s'ha vist en el capítol 3, construir n plans d'acord amb les diferents situacions (o estats del problema). La situació actual s_j^i s'aconsegueix mitjançant l'aplicació d'un pla p_{j-1}^i en una situació anterior s_{j-1}^i . L'execució de

⁴Aquesta simplificació dels plans dels nodes per l'acció que els diferencia del seu pare ha estat utilitzada moltes vegades, per qüestions de simplicitat, en les figures del capítol.

```

bolero()
1.  opt = preguntar-opció()
2.  Si opt = "aprenentatge-per-observació"
3.    Llavors incorporar(obtenir-cas())
4.  Altrament Si opt = "resoldre-problemes"
5.    Llavors resoldre-problema()
6.  Altrament Si opt = "avaluar"
7.    Llavors C = obtenir-cas()
8.    avaluar(C,obtenir-patro(C))
9.    Altrament aprenentatge-per-experiència()
aprenentatge-per-experiència()
; C: cas proporcionat per l'usuari
; CB: cas obtingut del procés de resolució de BOLERO
1.  CB = resoldre-problema()
2.  Si avaluar(CB,obtenir-patro(CB)) = "pla-reeixit"
3.    Llavors incorporar(CB)
4.  Altrament C = obtenir-cas()
5.    apr-importància-estratègica(C,CB)
6.    incorporar(C)

```

Taula 4.2: Algorisme global de BOLERO. D'acord amb l'opció que l'usuari escull en *preguntar-opció*, s'executa un dels mètodes: l'aprenentatge per observació (incorporació), la resolució de problemes (el pas elemental del qual és la construcció de plans), l'avaluació i l'aprenentatge per experiència (opció per defecte).

p_{j-1}^i condueix a conèixer un nou fet, per exemple f , que determina la nova (actual) situació $s_j^i = s_{j-1}^i \cup \{f\}$. D'aquesta manera, quan es resol un problema, la seqüència d'episodis $[\delta_0^i, \delta_1^i, \dots, \delta_{n_i}^i]$ representa els canvis de situació que s'han trobat durant el procés de resolució de problemes i els diferents plans $\bar{p}_0^i, \bar{p}_1^i, \dots, \bar{p}_{n_i}^i$ amb què el sistema ha reaccionat en cada situació. Un cas no és res més que un problema resolt, on els diferents estats del procés de resolució i els plans generats estan representats en seqüències d'episodis, de manera que podem dir que els casos són, en certa mesura, exemples (i memòries) de comportament reactiu. En el darrer episodi d'un cas hi ha el pla que condueix a assolir la solució de domini. La resolució de problemes és, doncs, una tasca que té lloc en l'arquitectura BOLERO-SBC. El mètode de construcció de plans que es presenta en aquest capítol és, per tant, el pas elemental de la resolució de problemes en BOLERO-SBC.

4.3 Aprenentatge per observació

BOLERO aprèn plans per observació organitzant en la seva memòria un conjunt de n casos $C^0, C^1, \dots, C^n$, que representen els problemes $\pi^0, \pi^1, \dots, \pi^n$ eventualment resolts per un mestre. La memòria de casos resultant de l'aprenentatge ha de ser útil per resoldre els mateixos problemes $\pi^0, \pi^1, \dots, \pi^n$, o problemes nous $\pi^j, j > n + 1$. El conjunt dels n casos $C^0, \dots, C^n$, s'anomena conjunt d'entrenament.

L'aprenentatge a BOLERO és incremental: inicialment la memòria és buida i els casos són incorporats un a un. La incorporació d'un nou cas en memòria resulta en una nova

```

resoldre-problema() → C:cas
    ; p: pla immediat amb el qual continuar la resolució de problemes
    ;  $\bar{p}$ : pla immediat/executat
    ; f: fet actual
    ; S: situació actual
    ; E: episodi actual
    ; C: cas actual
1.  p = pla-per-defecte
2.   $\bar{p} = p$ 
3.  S = {}, E = {}, C = {}
4.  Mentre final-rp = fals
5.      f = call-SBC(p)
6.      Si f = "final-acció"
7.          Llavors E = ⟨S,  $\bar{p}$ ⟩
8.          C = C ∪ E
9.          S = {}, E = {}
10.     final-rp = test-final-resolució-problemes
11.     Altrament S = S ∪ {f}
12.     p = construir-pla(f, S)
13.      $\bar{p} = \bar{p}$  ; primera-acció(p)
14. retornar(racionalització(C))

construir-pla(f:fet, A:situació) → p:pla
    ; N: conjunt de nodes activats i seleccionats
1.  N = N ∪ recuperació(f, S)
2.  Si N = ∅
3.  Llavors final-rp = cert
4.  Altrament adaptar(N)

```

Taula 4.3: Algorisme de resolució de problemes.

organització d'aquesta. La incorporació d'un cas en memòria involucra un procés de generalització i un procés d'aprenentatge de condicions de terminació. Continuem aquesta secció explicant cadascun dels components de l'aprenentatge per observació: la incorporació d'un cas (secció 4.3.1), la generalització de fets (secció 4.3.2) i l'aprenentatge de condicions de terminació (secció 4.3.3).

4.3.1 Incorporació d'un cas

Un cas C^i s'afegeix a la memòria per la comparació del pla $\bar{p}_j^i$ del seu episodi δ_j^i amb qualsevol node en el nivell j de la jerarquia. Si existeix un node N^k en el nivell j que tingui un pla $\bar{p}^k = \bar{p}_j^i$, llavors la generalització v^k de N^k s'actualitza com segueix

$$v^k = g_d(v^k, s^i)$$

on g_d és una funció de generalització esbiaxada pels coneixements sobre un domini (vegeu el proper apartat). El procés d'incorporació del cas C^i continua amb el següent episodi δ_{j+1}^i i els nodes del nivell $j+1$ successors de N^k . Aquest procés es realitza per a cada episodi del cas. Si no troba cap node amb un pla parcial igual a p_j^i , llavors es comença una nova branca

de la jerarquia des del nivell $i-1$ i la resta dels episodis del cas, $\delta_j^i, \dots, \delta_n^i$, esdevenen nodes d'aquesta nova branca. La figura 4.4 mostra l'algorisme del procés d'incorporació d'un cas en memòria.

```

incorporar(C:cas)
; cas:  $C = [\delta_0, \dots, \delta_n]$ ,  $\delta_i = \langle s_i, p_i \rangle$ 
; node:  $N = \langle v, \tilde{p}, \tau \rangle$ 
; node-terminal:  $N_t = \langle v, \tilde{p}, \tau, S \rangle$ 
1.  $i = 0, N = \text{arrel}$ 
2. Mentre  $i < n$ 
3.   Si  $\exists N' \in \text{successors}(N) | \tilde{p}' = p_i$ 
4.     Llavors  $\tau' = \tau' + 1$ 
5.        $v' = g_d(v', s_i)$ 
6.        $N = N'$ 
7.        $i = i + 1$ 
8.   Altrament  $N = \text{construir-nodes}([\delta_i, \dots, \delta_n], N)$ 
9.      $i = n$ 
10.  Si  $\exists N'_i \in \text{successors}(N)$ 
11.    Llavors aprenentatge-condicions-terminacio( $C, N'_i$ )
12.       $\tau' = \tau' + 1$ 
13.       $v' = g_d(v', s_i)$ 
14.  Altrament construir-node-terminal( $C, N$ )

construir-nodes( $\Sigma$ :episodis,  $N$ :node)
;  $\Sigma = [\delta_i, \dots, \delta_n]$ 
1.  $k = i$ 
2. Mentre  $k < n$ 
3.    $N' = \text{nou-node}$ 
4.    $\text{successors}(N) = \text{successors}(N) \cup \{N'\}$ 
5.    $v' = s_k$ 
6.    $\tilde{p}' = p_k$ 
7.    $k = k + 1$ 

```

Taula 4.4: Algorisme d'incorporació d'un cas en memòria.

4.3.2 Generalització de fets

Quan un pla ha estat realitzat en dues situacions diferents (i.e. existeixen dos episodis conformats pel mateix pla i diferents situacions), el procés de generalització proporciona una nova situació (generalitzada) que conté les dues situacions originals. Donada una situació generalitzada v^i d'un node i una situació s_j^k d'un episodi δ_j^k d'un cas, el mètode de generalització consisteix en els passos següents:

1. Realitzar una generalització de valors entre els fets de s_j^k i les generalitzacions g_j^i de v^i amb el mateix identificador (i.e. fets en $v^i \cap s_j^k$, on la operació d'intersecció es realitza respecte als identificadors de fets i generalitzacions).
2. Entre la resta de fets de s_j^k que no tenen cap generalització en v^i amb un identificador igual (i.e. fets en $s_j^k \setminus v^i$):

- (a) Si és possible una abstracció entre identificadors de fets i generalitzacions, aleshores cal fer-la
- (b) Altrament, si el fet no pertany a v^i , aleshores cal afegir el fet a v^i com una generalització (i.e. incorporant-li la importància estratègica i la freqüència adequada).

En aquest procés es distingeixen, doncs, la generalització de valors i l'abstracció d'identificadors [22]. La generalització de valors s'efectua entre els valors d'un fet i d'una generalització amb el mateix identificador. La generalització de dos valors v_1 i v_2 és la unió d'abdos valors $\{v_1\} \cup \{v_2\}$. Per a fets multivaluats s'hauria d'escriure $V_1 \cup V_2$. Per exemple, el resultat de la generalització entre els fets (dades-RX, {condensació-vmaldef, confluents}) i (dades-RX, {condensació-vmaldef, patró-intersticial}) és el nou fet (dades-RX, {condensació-vmaldef, confluents, patró-intersticial}). L'abstracció d'identificadors es basa en les relacions definides entre els identificadors en la xarxa de coneixements del domini. Donat un conjunt d'identificadors $id_1, id_2, \dots, id_n$, l'abstracció produeix un únic identificador id . El nou identificador id té una relació amb cadascun dels identificadors id_i en la xarxa de coneixements. Per exemple, suposem que disposem dels coneixements del domini representats en la figura 4.7 i els següents fets i/o generalitzacions: (assoc-càrdio-circulatòries, pericarditis), (assoc-neurològiques, encefalitis), i (assoc-cutànies, cellulitis), que pertanyen a diferents situacions de diferents episodis amb plans idèntics. El procés d'abstracció infereix la generalització (dades clíniques associades, {assoc-càrdio-circulatòries, assoc-neurològiques, assoc-cutànies}) que s'incorpora a la situació generalitzada corresponent.

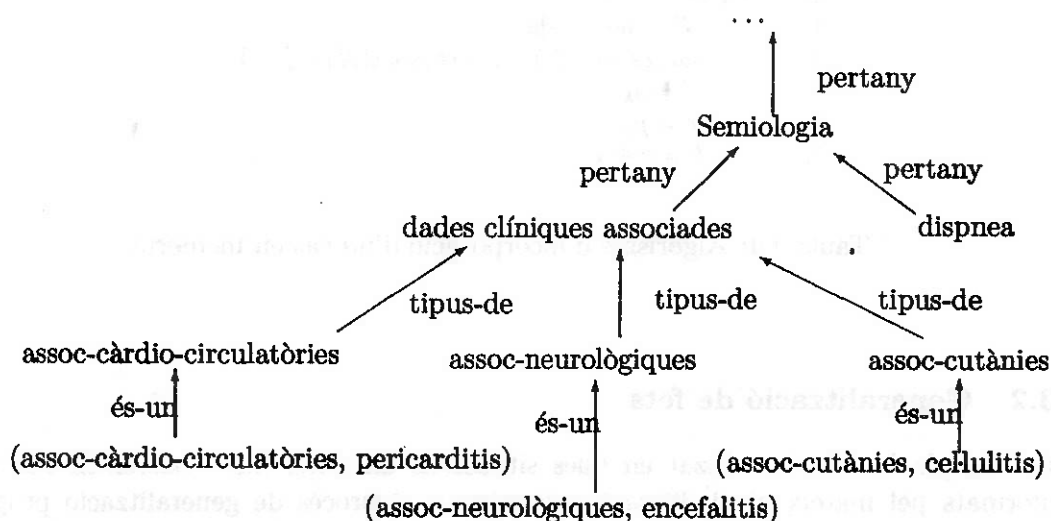


Figura 4.7: Visió parcial dels coneixements del domini on es mostren les relacions entre diferents fets. Els enllaços "és-un" representen lligams entre fets i identificadors, mentre que la resta de lligam ("pertany", "tipus-de") representen lligams entre identificadors.

En el procés de generalització s'actualitza la freqüència σ_j^i per a cada generalització de v^i . La freqüència σ_j^i d'una generalització g_j^i de la situació generalitzada v^i del node N^i es

modifica d'acord amb la següent equació [50]:

$$\sigma_j^i = \sigma_j^i + \frac{\lambda - \sigma_j^i}{\tau^i + 1}$$

on: τ^i és la tipicitat del node N^i ; $\lambda = 1$ si hi ha una generalització $g \in v^i$ amb un identificador igual a un fet f en la situació s_j^i de l'episodi; $\lambda = 0$ altrament.

El mètode de generalització és un mètode d'aprenentatge disjuntiu. És a dir, els plans es qualifiquen amb totes les seves característiques: per una banda, si una característica f_i en s_k^j no és comú a v^l (i.e. no pertany a $v^l \cap s_k^j$), aleshores s'afegeix f_i a v^l ; per una altra banda, si una generalització g_m^l pertany a v^l i no existeix cap fet f_i en s_j^l amb el mateix identificador, g_m^l es manté a v^l . La descripció disjuntiva dels plans ajuda a acarar el problema de l'ajustament excessiu (*overfitting* [15]) que presenten molts algorismes d'aprenentatge. El problema de l'ajustament excessiu relaciona el nombre de casos d'entrenament i la predictivitat dels conceptes apresos (la probabilitat que futurs casos del mateix concepte encaixin en la descripció del concepte après). Els algorismes d'aprenentatge conjuntius (i.e. algorismes que només tenen en compte les característiques comunes [120]) tendeixen a ser menys predictius quan el nombre de casos d'entrenament augmenta.

En algorismes disjuntius on les característiques que finalment s'emmagatzemen depenen de la seva freqüència d'aparició, també és possible construir generalitzacions ajustades excessivament, causades pel soroll en els exemples o pel processament d'un gran nombre d'exemples, amb dades molt diferents [52]. En aquests algorismes el problema de l'ajustament excessiu es divideix en dues parts [126]: les sobregeneralitzacions (*overgeneralizations*) i les sotsgeneralitzacions (*undergeneralizations*). Si una situació generalitzada v^i representa n casos, una generalització de v^i està sobregeneralitzada si està present en pocs dels n casos. Suposant una organització jeràrquica de la memòria anàloga a la de BOLERO, una generalització g de la situació generalitzada v^i en N^i està sotsgeneralitzada si existeix una generalització g' en una situació generalitzada v^j d'un node anterior N^j ($N^i \in \text{successor}(N^j)$) amb el mateix identificador, i ambdues generalitzacions estan compartides per una àmplia majoria dels casos que representen v^i i v^j . Detectar generalitzacions inadequades i corregir-les és molt important per a l'eficiència d'un sistema basat en casos. Quan una generalització és reconeguda com a sobregeneralitzada o sotsgeneralitzada, s'elimina de la situació generalitzada a la qual pertany. D'aquesta manera s'intenta pal·liar el problema de l'ajustament excessiu. BOLERO no fa cap tractament de la detecció de sobregeneralitzacions ni sotsgeneralitzacions perquè no elimina cap generalització dependent de la seva freqüència. La freqüència dels fets, però, sí que s'utilitza en la recuperació dels casos en memòria. Com es veurà en la secció 4.4.1, l'absència de fets que tenen una freqüència elevada en les generalitzacions penalitza més que l'absència de fets que tenen una escassa freqüència d'aparició.

4.3.3 Aprenentatge de condicions de terminació

BOLERO com a metanivell ha de saber aturar la resolució d'un problema en l'arquitectura BOLERO-SBC quan s'ha obtingut la solució. Reconèixer la solució d'un problema normalment és possible perquè les especificacions de la situació final que conté la solució d'un problema vénen donades. Però, com ha estat introduït en la secció 2.2, en el tipus d'aplicacions en què BOLERO ha estat experimentat (diagnosi mèdica) les especificacions de l'estat final no

es proporcionen i, consegüentment, BOLERO, juntament amb l'aprenentatge de plans, també ha d'aprendre les condicions sota les quals un problema es considera resolt. Així, es defineixen en BOLERO les condicions de terminació com el conjunt de fets S^j emmagatzemats en els nodes terminals N_t^j més un conjunt de condicions S_c definides per a un domini en concret. Tant S^j com S_c determinen la solució d'un problema, com es mostra en la secció 4.4.3.

Les condicions de terminació S^i són les solucions de domini que apareixen en els casos, i.e. S_C^i (veure secció 4.1.7). És important recordar que casos amb el mateix pla final poden tenir una solució de domini diferent. Com hem vist en la secció 4.1.7, la solució de domini d'un cas es determina pel resultat de les accions del pla final. I el resultat de les accions depenen dels esdeveniments conegut en el cas. Per exemple, els casos de la taula 4.5 tenen el mateix pla final [recollida-dades, bact-atip, pneumococ, legionel·la, enterobacteriàcies, s-piògen], però les solucions de domini són diferents. El fet (esput, rovellós) ha estat determinant per concloure (pneum-pneumococ, cert) en el cas r13v1; mentre que el fet (esput hemoptoic) ha estat crucial per deduir (pneum-enterobacteriàcies, molt-possible) en el cas m03v1; i, finalment, els fets antibiòtics-previs, malalties-associades, i hiponatrèmia han estat decisius per diagnosticar (pneum-legionel·la, molt-possible), en el cas m07v1.

	r13v1	m03v2	m07v1
antecedents	<i>diabètic mal-resp-crònica</i>	<i>fumador alcohòlic mal-resp-crònica</i>	<i>fumador alcohòlic mal-resp-crònica</i>
tos	<i>sí</i>	<i>sí</i>	<i>sí</i>
expectoració	<i>sí</i>	<i>sí</i>	<i>sí</i>
esput	<i>rovellós</i>	<i>purulent hemoptoic</i>	<i>mucós</i>
antibiòtics-previs	<i>no</i>	<i>no</i>	<i>sí</i>
dades-raigsX	<i>condensació-vmaldef confluents</i>	<i>condensació-vmaldef confluents</i>	<i>broncograma-aeri</i>
malalties-associades		<i>aparell-respiratòri aparell-digestiu</i>	<i>aparell-disgestiu</i>
hiponatrèmia	<i>desconegut</i>	<i>desconegut</i>	<i>força-possible</i>
dades-malalts	<i>sí</i>	<i>sí</i>	<i>sí</i>
pneum-bacteriana	<i>molt-possible</i>	<i>molt-possible</i>	<i>força-possible</i>
pneum-pneumocòccia	<i>cert</i>	<i>possible</i>	<i>possible</i>
pneum-legionella	<i>poc-possible</i>	<i>mod-possible</i>	<i>molt-possible</i>
pneum-enterobacteriàcies	<i>poc-possible</i>	<i>molt-possible</i>	<i>possible</i>
pneum-s-piògen	<i>desconegut</i>	<i>poc-possible</i>	<i>desconegut</i>

Taula 4.5: Fets de tres casos que s'han resolt amb el mateix pla final: [recollida-dades, bact-atip, pneumococ, legionel·la, enterobacteriàcies, s-piògen]. Els darrers fets són els resultats de les accions.

L'aprenentatge de les condicions de terminació S^j consisteix en l'emmagatzematge de les solucions de domini dels casos en els nodes terminals, sense efectuar cap procés addicional.

Com s'ha mencionat amb anterioritat, les condicions de terminacions definides en els nodes terminals són utilitzades per determinar la solució d'un cas. Tanmateix, S^i són condicions dèbils en el sentit que és molt estrany que dos casos tinguin la mateixa solució, i en conseqüència és molt optimista pensar que en el futur trobarem casos amb solucions idèntiques. Per aquesta raó, també s'ha proporcionat a BOLERO un conjunt de condicions de terminació S_c .

Les condicions de terminació S_c són coneixements de domini que no s'aprenen, sinó que les estableix l'expert mestre. Les condicions de terminació definides a S_c són condicions fortes en el sentit que si alguna d'elles es compleix, la solució d'un problema es pot considerar assolida. Aquestes condicions de terminació van ser definides en l'entorn de desenvolupament de sistemes experts MILORD [142] per tal d'establir la terminació dels sistemes experts que s'hi han implementat. Les condicions de terminació tenen la sintaxi següent:

$$\text{condició} ::= \text{conegut}(f) \wedge (\text{valor}(f) \text{ OperRel } v)$$

on $\text{conegut}(f)$ és un predicat que és cert quan el fet f és un esdeveniment conegut; $\text{valor}(f)$ és un predicat que retorna el valor de f , OperRel és un operador relacional com $=$, $\geq$, etc., i v és un valor donat. El format de les condicions assumeix que només poden ser aplicades a fets definits en F_a i, doncs, fets que tenen valors comparables utilitzant OperRel (vegeu 4.1.1). La representació de les condicions de S_c poden estendre's a altres tipus de condicions, on apareguin més termes, més tipus de predicats o fins i tot es puguin fer càlculs. Les condicions de terminació en S_c són condicions fortes, de manera que si qualsevol d'elles se satisfà, es pot dir que s'ha trobat la solució del problema i, per tant, que es pot aturar el procés de resolució. La secció 4.4.3 explica com s'efectua la terminació de la resolució de problemes.

4.4 Construcció de plans

En aquesta secció ens focalitzem en com BOLERO construeix un únic pla en una situació determinada. El mètode, com qualsevol altre sistema basat en casos, consta de dos passos fonamentals: recuperació i adaptació. Aquest dos passos s'iteren cada vegada que es coneix una nova situació s^i de l'entorn, com a conseqüència de la imbricació de la planificació i l'execució de plans en el sistema multinivell BOLERO-SBC (veure secció 2.1), fins que BOLERO decideix aturar el procés de resolució d'un problema (vegeu l'algorisme de resolució de problemes en la taula 4.3).

A continuació descrivim els mètodes de recuperació de plans (secció 4.4.1) i adaptació (secció 4.4.2), i finalitzem aquesta secció explicant el mètode desenvolupat per establir la terminació del procés de resolució (secció 4.4.3).

4.4.1 Recuperació de plans

Suposem que estem resolent el problema π^i , i que es coneixen un conjunt de fets o situació actual s_a^i . El mètode de recuperació consisteix a recordar les situacions generalitzades emmagatzemades en els nodes mitjançant un mecanisme de *propagació restringida d'activació* (*constrained spreading activation* [72, 45]), i utilitzant els fets com a índexs (*reminders*). Totes les situacions generalitzades recuperades es comparen amb s_a^i i només se'n selecciona una d'entre totes. L'algorisme de recuperació es mostra en la taula 4.6.

La propagació d'activació consisteix a activar els nodes que tenen qualsevol fet f contingut en la situació s_a^i i qualsevol altre fet relacionat amb f mitjançant la xarxa de coneixements

- recuperació(f :fet, S :situació) $\rightarrow \mathcal{N}$: conjunt de nodes
 ; $\mathcal{N}$: conjunt de nodes activats en la resolució de problemes
1. $\mathcal{N} = \mathcal{N} \cup \text{propagació-activació}(f)$
 2. Per a cada $N^i \in \mathcal{N}$
 3. calcular $m_s(S, v^i)$
 4. Seleccionar($\mathcal{N}$)

Taula 4.6: Algorisme de recuperació de nodes.

del domini. Si un fet $f_1 \in s_a^i$ té una relació de pertinença, inclusió, etc. en els coneixements del domini amb un altre fet $f_2 \notin s_a^i$, aleshores el fet f_2 s'activa. La propagació restringida es refereix al fet que no totes les relacions establertes en la xarxa de coneixements del domini poden usar-se en l'activació. Per exemple, si un fet f_1 activa un fet f_2 perquè f_1 és un tipus de f_2 , no es pot activar un tercer fet f_3 perquè estableixi una relació de pertinença amb f_2 com per exemple f_2 s'especialitza en f_3 . Un fet f activat (i.e. pertanyi o no a s_a^i) s'usa per activar els nodes que indexa.

Les situacions v^i dels nodes activats en el procés de propagació d'activació es comparen amb la situació s_a^i per determinar el grau de semblança entre v^i i la situació actual s_a^i . Aquest procés és conegut com el *matching* de dues situacions. El pla $\tilde{p}^k$ del node N^k associat a la situació v^k que té el grau de semblança més alt amb s_a^i és seleccionat. El *matching* es realitza de forma incremental per l'esdeveniment obtingut més recentment, f (cal recordar que l'execució i la planificació estan imbricades, de forma que s'adquireix informació de l'entorn incrementalment).

La funció de comparació o *matching* es basa en dues funcions de similitud, m_f i m_s , i el resultat és un valor dins de l'interval $[0,1]$. La primera funció m_f és una mesura de similitud entre fets i generalitzacions, mentre que la funció m_s determina la similitud entre situacions. Donada una generalització g_k^l que pertany a una situació generalitzada de memòria v^l , i un fet f_a de la situació s_a^i amb el mateix identificador (altrament $m_f(g_k^l, f_a) = 0$),

$$m_f(g_k^l, f_a) = \begin{cases} 1 & \text{si } (\text{monovaluat}(f_a) \wedge v_a = v_k^l) \vee (\text{multivaluat}(f_a) \wedge V_a \cap V_k^l \neq \emptyset) \\ 1 - d(v_a, v_k^l) & \text{si } \text{difús}(f_a) \\ 0 & \text{altrament} \end{cases}$$

on $d(v_a, v_k^l)$ és una funció que s'avalua en l'interval $[0,1]$ i mesura la distància entre els valors difusos v_a i v_k^l ⁵. La funció m_s combina l'avaluació de m_f de cadascuna de les generalitzacions g_k^l de v^l de la manera següent:

$$m_s(v^l, s_a^i) = \frac{\sum_{g_k^l \in v^l} W(g_k^l, f_a) m_f(g_k^l, f_a)}{n+k} + \frac{\sum_{g_k^l \in v^l} W(g_k^l, f_a) m_f(g_k^l, f_a)}{2 |s_a^i|}$$

on n és el nombre de generalitzacions de v^l que tenen una freqüència d'aparició igual a 1.0; k és el nombre de generalitzacions de v^l que també són a s_a^i (i.e. estan actives) i la freqüència de les quals és més petita que 1; $|s_a^i|$ és la cardinalitat de s_a^i i W la importància estratègica

⁵Cal recordar que els valors de fets numèrics pertanyen a un conjunt d'etiquetes lingüístiques totalment ordenades, i que aleshores és possible assignar un número a cada etiqueta.

calculada per a la generalització g_k^l en la situació s_a^i . El valor de W es determina com segueix:

$$W(g_k^l, f_a) = \begin{cases} w_k^l & \text{si } \text{monovaluat}(f_a) \\ C_n(\{w_i | v_i \in V_k^l \cap V_a\}) & \text{si } \text{multivaluat}(f_a) \\ 0 & \text{altrament} \end{cases}$$

on C_n és una funció d'agregació disjuntiva de n evidències [122, 51, 21], basada en una t-conorma. En l'apèndix B és descriu la funció t-conorma, però per poder continuar detallem que entre les formes possibles que pot prendre C_n , hem experimentat amb la següent: $C_n(x_1, \dots, x_n) = \max(x_1, \dots, x_n)$.

La funció m_s calcula una mitjana entre el grau d'inclusió (primer terme del numerador) i el grau d'exclusió (segon terme) dels fets de la situació actual en una situació generalitzada v^l . Cal destacar que la importància estratègica dels fets és determinant en el resultat de l'avaluació de m_s , ja que determina el pes amb què cada generalització intervé en la determinació de la similitud entre situacions.

Quan v^l i s_a^i són iguals, aleshores, encara que $m_f(g_k^l, f_a) \leq 1$ per a qualque g_k^l de v^l , el *matching* final val 1, i.e. $m_s(v^l, s_a^i) = 1$. És a dir les funcions reconeixen les situacions idèntiques.

En el procés de *matching*, doncs, es calcula la funció m_f per a les generalitzacions de la situació generalitzada v^l dels nodes activats, i a continuació es calcula m_s respecte a s_a^i per a cada node. Cada node té associat finalment un grau de versemblança amb la situació actual s_a^i , i d'entre tots s'ha d'escollir un amb el qual construir un pla. En primer lloc s'eliminen aquells nodes les accions del pla del qual ja estan completament executades (veure 4.1.3). Si la resta de nodes activats $\mathcal{N}$ són tots versemblants, BOLERO disposa d'un conjunt de criteris heurístics per escollir-ne només un. Aquestes heurístiques són:

1. Escollir ancestres comuns: si tots els nodes de $\mathcal{N}$ tenen un ancestre comú N , escollir N .
2. Escollir pla comú: si existeix un pla comú a tots els nodes (i.e. $\cap_{N^i \in \mathcal{N}} p^i \neq \emptyset$), aleshores seleccionar tots els nodes.
3. Preferir plans pendents més curts: d'entre tots els nodes $\mathcal{N}$, escollir aquell que tingui el nombre més petit d'accions en el pla que encara no s'han executat
4. Preferir nodes terminals: escollir entre els nodes activats $\mathcal{N}$ aquells que siguin nodes terminals N_t .
5. Preferir plans pendents de descendents més curts: d'entre tots els nodes, escollir aquell que tingui un descendent que sigui node terminal amb el pla més curt.
6. Preferir nodes típics: escollir el node que presenta la tipicalitat τ més elevada.
7. Pla actual: si existeix un node N^i a $\mathcal{N}$ el pla del qual és el que s'està executant, aleshores escollir N^i .
8. Pla més general: escollir el node que està situat més a prop de l'arrel de la memòria de plans (nivell inferior).

El resultat del procés de recuperació és, d'aquesta manera, un únic pla $\tilde{p}^k$ corresponent al node N^k activat i seleccionat⁶.

⁶En la segona heurística pot produir més d'un node seleccionat, però que corresponen a un únic pla $p^k = \cap_{N^i \in \mathcal{N}} p^i$. Per qüestions de simplicitat ens referirem a un únic node seleccionat N^k , però el que es digui és aplicable als $\mathcal{N}$ casos en el cas que s'hagin seleccionat per la segona heurística.

4.4.2 Adaptació de plans

L'última estapa en la construcció d'un pla consisteix a adaptar el pla recuperat de la memòria a la situació actual. L'adaptació del pla $\tilde{p}^k$ del node N^k consisteix a no repetir accions completades. És a dir, si una acció a_j^i ha estat realitzada en un moment x , aleshores, i donat el caràcter monòton de les situacions, per a qualsevol situació s_y^i posterior a s_x^i , $result(a_j^i) \subset s_y^i$. En un moment donat x del procés de resolució d'un problema π^i existeix un conjunt d'accions completades (veure 4.1.3) que anomenem $\hat{A} \in A$. Les accions completades són aquelles que $a_j^i \in \tilde{p}_x^i$ i $result(a_j^i) \subset s_x^i$. El pla $\tilde{p}^k$ es modifica de manera que el resultat de l'adaptació és un nou pla $p_a^i = [a_0^i, \dots, a_n^i]$, on tota acció a_0^i pertany a $\tilde{p}^k$ però no pertany a $\hat{A}$. El pla p_a^i es comença a executar a partir de l'acció a_0^i .

4.4.3 Terminació de l'execució

Existeixen quatre possibilitats amb les quals posar terme a la resolució d'un problema: 1) exhauriment dels nodes actius, 2) satisfacció d'una condició d'acabament de S_c , 3) distància entre la solució actual i una solució S^i d'un node terminal sota un llindar i 4) detecció de dissimilitud entre situacions.

Les dues primeres possibilitats són força evidents. L'exhauriment dels nodes actius succeeix quan el procés de recuperació no produeix cap node amb el qual continuar la resolució del problema (bé perquè no s'han activat nodes, bé perquè les accions dels plans dels nodes activats estan completades). La segona possibilitat de posar terme a la resolució d'un problema esdevé quan qualsevol de les condicions especificades en les solucions de domini proporcionades per l'oracle S_c se satisfà en la situació actual s_a^i .

Les dues darreres possibilitats amb les quals es posa fi a la resolució d'un problema són el resultat d'un procés més estudiat que es va iniciar en observar que les dues primeres condicions de terminació de la resolució de problemes no resultava eficient. En primer lloc, l'exhauriment de nodes actius significa que per resoldre un problema s'ha recorregut tots els casos precedents, fent una cerca exhaustiva de totes les solucions, i BOLERO mostraria doncs, un comportament "poc" intel·ligent, i.e. no milloraria el comportament de SBC^B . En segon lloc, les condicions de terminació de S_c són condicions fortes però que no sempre succeeixen. És a dir, es poden executar totes les accions possibles d'una aplicació i encara no haver-se satisfet cap condició de S_c . La raó s'ha de cercar en el tipus d'informació que compon els casos: moltes vegades la informació és tan incompleta que resulta impossible obtenir fets com a conseqüència de l'aplicació de les accions (i.e. fets de F_f) amb un cert grau d'evidència. Per exemple, suposem que el cas "r01v1" ha estat resolt per l'expert mestre amb el pla final:

$p_{r01v1} = [\text{recollida-dades, bacteriana-atípica, pneumococ,} \\ \text{enterobacteriàcies, anaerobi, hemòfil,} \\ \text{micoplasma, virus, tuberculosi}]$

que ha conduït a conèixer els diagnòstics següents:

(dades-malalt, cert) (pneumònia-bacteriana, força-possible)
 (pneumònia-atípica, lleugerament-possible)
 (pneumònia-pneumocòccia, possible)
 (pneumònia-per-enterobacteriàcies, moderadament-possible)
 (pneumònia-per-gèrmens-anaerobis, moderadament-possible)
 (pneumònia-per-hemòfil, lleugerament-possible)
 (pneumònia-per-micoplasma, molt-poc-possible)
 (pneumònia-vírica, molt-poc-possible)
 (pneumònia-tuberculosa, gens-possible)

Suposem també que les condicions de domini S_c adquirides de l'expert mestre són les que es mostren en la figura 4.8, i que els valors difusos possibles són els següents (ordenats): gens-possible, molt-poc-possible, lleugerament-possible, moderadament-possible, possible, força-possible, molt-possible, cert. Aleshores, BOLERO no aturarà la resolució del problema un cop executat el mateix pla que el mestre, car cap fet diagnòstic del cas "r01v1" satisfà cap condició de les definides en S_c :

valor(pneumònia-pneumocòccica) = possible < molt-possible
 valor(pneumònia-per-enterobacteriàcies) = moderadament-possible < força-possible
 valor(pneumònia-per-gèrmens-anaerobis) = moderadament-possible < força-possible
 valor(pneumònia-per-hemòfil) = lleugerament-possible < força-possible
 valor(pneumònia-per-micoplasma) = molt-poc-possible < força-possible
 valor(pneumònia-vírica) = molt-poc-possible < força-possible
 valor(pneumònia-tuberculosa) = gens-possible < força-possible

La necessitat de cercar alternatives perquè BOLERO sigui capaç d'aturar la resolució de problemes ha dut a la definició de dos nous mecanismes que determinin la terminació d'un problema.

La primera proposta ha estat la de poder reconèixer quan s'ha assolit una solució de domini a partir de les solucions de casos precedents emmagatzemades en els nodes terminals. Per dur a terme aquesta proposta s'ha determinat la semblança entre la solució de domini actual i una solució d'un node terminal N_i^t basada en la distància euclidiana. El node terminal N_i^t ha de ser un node el pla $\tilde{p}^i$ del qual hagi estat recuperat de memòria, i que totes les accions de $\tilde{p}^i$ hagin estat completades. Consegüentment, tots els resultats de les accions de $\tilde{p}^i$ pertanyen a s_a^i , i podem determinar la solució del domini $S_{s_a^i}$ per a la situació actual. Aleshores es defineix la distància euclidiana reduïda entre la solució $S_{s_a^i}$ i una solució $S_{Cj} \in S^i$ com:

$$d_e(S_{s_a^i}, S_{Cj}) = \sqrt{\frac{1}{n} \sum_{m=1}^n (v_{am} - v_{jm})^2}$$

on: n és el nombre de resultats possibles en el domini (i.e. $n = |F_f|$); v_{am} és un valor numèric associat al valor del fet difús f_m que apareix a la solució actual $S_{s_a^i}$; i v_{jm} és un valor numèric associat al valor del fet difús f_m que apareix a la solució S_{Cj} . La distància euclidiana reduïda entre solucions ja ha estat utilitzada amb anterioritat en l'avaluació de sistemes experts ([166, 16]). Per calcular la distància s'assumeix que els fets dels resultats de les accions són difusos, i.e. $result(a_j) \in F_f$ i així es pot assignar un valor numèric als valors dels fets que resulten de l'execució de les accions⁷. Utilitzant la distància euclidiana definida

⁷Per exemple es pot assignar 1 al valor gens-possible, 2 per molt-poc-possible, ..., i 8 per cert. En particular al valor desconegut s'assigna el pitjor número del rànquing: 0.

C1:	conegut(pneum-pneumococ) $\wedge$ valor(pneum-pneumococ) $\geq$ molt-possible
C2:	conegut(pneum-per-enterobacteriàcies) $\wedge$ valor(pneum-per-enterobactèries) $\geq$ força-possible
C3:	conegut(pneum-per-clamídia) $\wedge$ valor(pneum-per-clamídia) $\geq$ força-possible
C4:	conegut(febre-q) $\wedge$ valor(febre-q) $\geq$ possible
C5:	conegut(pneum-per-micoplasma) $\wedge$ valor(pneum-per-micoplasma) $\geq$ força-possible
C6:	conegut(pneum-vírica) $\wedge$ valor(pneum-vírica) $\geq$ força-possible
C7:	conegut(pneum-per-citomegalovirus) $\wedge$ valor(pneum-per-citomegalovirus) $\geq$ possible
C8:	conegut(pneum-per-p-carinii) $\wedge$ valor(pneum-per-p-carinii) $\geq$ força-possible
C9:	conegut(pneum-per-aspergillus) $\wedge$ valor(pneum-per-aspergillus) $\geq$ possible
C10:	conegut(pneum-per-nocàrdia) $\wedge$ valor(pneum-per-nocàrdia) $\geq$ mod-possible
C11:	conegut(pneum-per-criptococ) $\wedge$ valor(pneum-per-criptococ) $\geq$ possible
C12:	conegut(pneum-tuberculosa) $\wedge$ valor(pneum-tuberculosa) $\geq$ força-possible
C13:	conegut(pneum-per-pseudomones) $\wedge$ valor(pneum-per-pseudomones) $\geq$ força-possible
C14:	conegut(pneum-meningocòccia) $\wedge$ valor(pneum-meningocòccia) $\geq$ mod-possible
C15:	conegut(pneum-per-branhamella) $\wedge$ valor(pneum-per-branhamella) $\geq$ mod-possible
C16:	conegut(pneum-per-gèrmens-anaerobis) $\wedge$ valor(pneum-per-gèrmens-anaerobis) $\geq$ força-possible
C17:	conegut(pneum-per-hemòfil) $\wedge$ valor(pneum-per-hemòfil) $\geq$ força-possible
C18:	conegut(pneum-varicel·lar) $\wedge$ valor(pneum-varicel·lar) $\geq$ força-possible
C19:	conegut(pneum-vírica-complicada) $\wedge$ valor(pneum-vírica-complicada) $\geq$ força-possible
C20:	conegut(pneum-per-legionella) $\wedge$ valor(pneum-per-legionella) $\geq$ força-possible
C21:	conegut(pneum-estafilocòccia) $\wedge$ valor(pneum-estafilocòccia) $\geq$ força-possible
C22:	conegut(pneum-per-s-piògen) $\wedge$ valor(pneum-per-s-piògen) $\geq$ possible

Figura 4.8: Condicions de terminació adquirides de l'expert mestre per a la diagnosi de pneumònies [166]. Per qüestions d'espai s'ha abreujat "pneumònia" per "pneum", i "moderadament-possible" per "mod-possible".

com a dalt, podem establir un llinar ξ , i aleshores si

$$d_e(S_{s,i}, S_{C,i}) \leq \xi$$

diem que la solució actual $S_{s,i}$ s'assembla suficientment a una solució d'una experiència anterior, i es pot donar per resolt el problema.

Un altre mecanisme desenvolupat en BOLERO per aturar la resolució de problemes es basa en el nombre de casos precedents en memòria (i.e. plans finals o nodes terminals). La quantitat de nodes activats depèn del nombre de nodes en memòria. És a dir, si només hi ha 2 casos en memòria, i volem resoldre un tercer, és molt probable que es recuperin els dos casos precedents per resoldre el tercer. D'aquesta manera s'atura la resolució de problemes per l'exhauriment de nodes actius. Ara bé, el grau d'activació dels nodes recuperats és molt baix, donat que és molt poc plausible que el tercer cas sigui similar a cap dels precedents. Si s'aprenen, però, 100 casos, el nombre de nodes augmenta, i hi ha més possibilitats de trobar casos semblants en el procés de resolució d'un problema donat. Utilitzar tots els nodes activats per resoldre un problema pot resultar costós, com ja ha estat mencionat amb anterioritat, i és una qüestió que cal evitar. Utilitzar un únic node per resoldre un problema no resulta suficient tenint en compte que, donada la incompletesa i incertesa de la informació, en l'inici de la resolució de problemes es fa ús de nodes amb plans parcials, i que per tant no condueixen a l'assoliment d'una solució de domini. Per aquesta raó s'ha definit un mecanisme per aturar la resolució de problemes després que s'hagin executat els plans d'un nombre determinat de nodes, mecanisme que es basa en la detecció de dissimilitud entre situacions.

Per explicar què és el que entenem per situacions dissimilars, imaginem la memòria de plans com una classificació (*clustering*) de generalitzacions. Cada node terminal N_i^t conté una classe v^i . Quan més classes hi ha en memòria (i.e. nodes terminals), més fina és la classificació

i, conseguintment, més petita la distància entre classes. Aleshores, la incorporació en memòria de més casos amb plans finals diferents (i.e. més nodes terminals) determina que la classificació sigui de granularitat més fina i, en conseqüència, la distància entre les classes és més reduïda. La distància entre classes està implementada en BOLERO mitjançant el paràmetre *mdf*. Aquest paràmetre varia d'acord amb el nombre de classes com segueix (figura 4.9):

$$mdf = 1 - e^{-\frac{\lambda}{n}}$$

on n és el nombre de nodes terminals en memòria i λ un paràmetre estimat experimentalment (vegeu el capítol 5 per una avaluació d'aquest paràmetre). Quan més gran sigui λ , més petita serà la distància *mdf*.

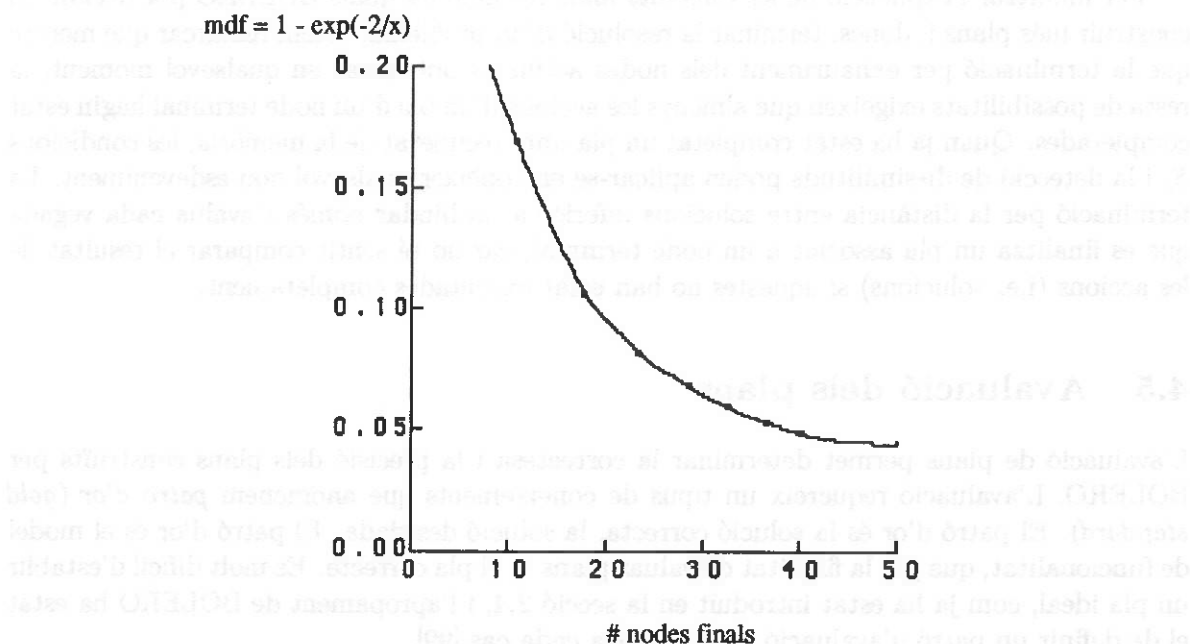


Figura 4.9: Variació del paràmetre *mdf* en funció del nombre de nodes terminals per $\lambda = 2$.

La situació actual s_i^a pot pertànyer a més d'una classe, segons l'avaluació de la funció de *matching* entre s_i^a i les situacions generalitzades v . Quan existeixen dues situacions generalitzades v^1 i v^2 tals que el *matching* de cadascuna amb la situació actual s_i^a és el mateix, finalment, tal i com s'ha descrit en el mètode de recuperació, només s'escull un node associat a una situació. Suposem que se selecciona N_t^1 , i s'executa el pla p^1 (un cop adaptat). Suposem també que un cop completades totes les accions de p^1 (i.e. el pla ha estat executat) en s_j^a , no hi ha la possibilitat de posar terme a la resolució del problema. BOLERO pot continuar amb la resolució del problema seleccionant el pla de N_t^2 (si encara continua v^2 essent la situació generalitzada més versemblant en s_j^a). En les noves circumstàncies, pot existir una tercera situació generalitzada v^3 tal que $m_s(v^2, s_j^a) = m_s(v^3, s_j^a)$. D'aquesta

manera, i després de l'execució del pla p^2 (adaptat), BOLERO podria executar el pla del node N_t^3 . I així successivament fins que BOLERO executa tots els plans de memòria.

El que realment ocorre és que $m_s(v^1, s_i^a) = m_s(v^2, s_i^a)$, però $m_s(v^1, s_j^a) \geq m_s(v^2, s_j^a)$. És a dir, malgrat que en una determinada situació s_i^a , v^1 i v^2 foren similars, a mesura que es coneix més informació, ambdues situacions generalitzades esdevenen dissimilars respecte a la situació actual. Per detectar les dissimilituds entre situacions s'utilitza la distància mdf . Si N és un node el pla del qual està completament executat, i N' el darrer node recuperat de memòria (i amb el qual BOLERO construirà un nou pla per continuar la resolució del problema), el procés de resolució s'atura en s_a si la dissimilitud és superior al llindar mdf , és a dir si:

$$m_s(s_a, v) - m_s(s_a, v') \geq mdf$$

Per finalitzar l'explicació de les diferents maneres amb les quals BOLERO pot decidir no construir més plans i, doncs, terminar la resolució d'un problema, volem remarcar que mentre que la terminació per exhauriment dels nodes actius es pot donar en qualsevol moment, la resta de possibilitats exigeixen que almenys les accions d'un pla d'un node terminal hagin estat completades. Quan ja ha estat completat un pla final recuperat de la memòria, les condicions S_c i la detecció de dissimilituds poden aplicar-se en conèixer qualsevol nou esdeveniment. La terminació per la distància entre solucions inferior a un llindar només s'avalua cada vegada que es finalitza un pla associat a un node terminal, car no té sentit comparar el resultat de les accions (i.e. solucions) si aquestes no han estat executades completament.

4.5 Avaluació dels plans

L'avaluació de plans permet determinar la correctesa i la precisió dels plans construïts per BOLERO. L'avaluació requereix un tipus de coneixements que anomenem *patró d'or* (*gold standard*). El patró d'or és la solució correcta, la solució desitjada. El patró d'or és el model de funcionalitat, que per la finalitat d'avaluar plans és el pla correcte. És molt difícil d'establir un pla ideal, com ja ha estat introduït en la secció 2.4, i l'apropament de BOLERO ha estat el de definir un patró d'avaluació $PA(C^i)$ per a cada cas [99].

El patró d'avaluació d'un cas C^i es defineix com un parell $\langle PA_p(C^i), PA_d(C^i) \rangle$ on $PA_p(C^i)$ és un pla admissible i $PA_d(C^i)$ un conjunt de solucions del domini. L'avaluació de plans s'aplica sobre els plans finals dels casos que ha resolt BOLERO (un cop BOLERO atura la resolució del problema π^i , s'obté un nou cas C^i amb un pla final $\bar{p}_n^i$). El pla admissible $PA_p(C^i)$ s'utilitza per determinar la correctesa del pla $\bar{p}_n^i$ i en la determinació de la precisió dels plans intervenen tots els components del patró d'avaluació $PA(C^i)$. A continuació es descriu el mètode per determinar la correctesa d'un pla (secció 4.5.1) i després es detalla el mètode per determinar la precisió dels plans (secció 4.5.2).

4.5.1 Correctesa dels plans

Per parlar de la correctesa dels plans hem definit el *grau d'assoliment* d'un pla. El grau d'assoliment d'un pla $\bar{p}_n^i$ d'un cas C^i és la raó entre el nombre d'accions de $\bar{p}_n^i$ incloses en el pla admissible $PA_p(C^i)$ respecte al nombre total d'accions de $PA_p(C^i)$. En aquest sentit, un pla és totalment correcte si el grau d'assoliment del pla és del 100%. Per exemple, suposem que BOLERO resol el problema *g24v1* mitjançant el pla final:

[recollida-dades, bacteriana-atípica, clamídia,
micoplasma, febre-q]

i que el $PA_p(g24v21)$ és:

[recollida-dades, bacteriana-atípica, virus,
micoplasma, clamídia]

En aquestes circumstàncies, el pla de BOLERO no és correcte ja que l'acció virus no ha estat inclosa en el pla, i el grau d'assoliment del pla és del 80% (4/5).

Definint la correctesa com el grau d'assoliment del 100% garanteix que BOLERO realitza totes les accions que realitza el mestre i, per tant, suposant que el mestre disposa dels mateixos coneixements de domini de SBC^B , BOLERO aconseguirà la mateixa solució de domini que el mestre. El fet d'obligar a contenir totes les accions emmagatzemades en $PA_p(C^i)$ fa, però, que BOLERO tingui un biaix (*bias*) conservador. Aquest biaix conservador és una conseqüència del tipus d'aplicació desenvolupat en BOLERO, que s'adequa a l'actitud dels metges en la diagnosi mèdica: els metges prefereixen realitzar proves addicionals abans que oblidar-se de considerar alguna malaltia perillosa. I aquest biaix s'accentua en un sistema automàtic d'ajuda al diagnòstic. Dit altrament, fer diagnòstics de més és un error força més acceptable que fer diagnòstics de menys, cosa que és un error greu.

Volem fer constar que inicialment es va pensar amb una mesura de la correctesa més estricta. Consistia a dir que un pla $\bar{p}_{n^i}$ era correcte si incloïa el patró PA_p ,

$$PA_p \subseteq p_{n^i}$$

Però aquesta mesura resulta injusta quan hi ha plans de diferents longituds. És a dir, amb una mesura com aquesta és més probable que plans curts siguin correctes, i que plans llargs siguin incorrectes. Per aquesta raó s'ha definit la correctesa dels plans basada en el grau d'assoliment.

4.5.2 Precisió dels plans

La precisió dels plans es determina mitjançant dues mesures: el grau d'enfocament i el grau d'ordenació.

El grau d'enfocament. El grau d'enfocament es defineix fàcilment com la mesura contrària al grau de des-enfocament:

$$\text{grau d'enfocament} = 100 - \text{grau de des-enfocament}$$

El grau de des-enfocament és el percentage d'accions completades que es podrien haver estalviat (i.e. que no eren necessàries) respecte al nombre total d'accions del pla p_{n^i} del pla generat per BOLERO. La detecció d'accions sobreres (no necessàries) en un pla final $\bar{p}_{n^i}$ es basa en el pla admissible $PA_p(C^i)$ i les solucions de $PA_d(C^i)$. Entre totes les accions d'un pla final $\bar{p}_{n^i}$ que no figuren a $PA_p(C^i)$, es consideren accions sobreres les accions el resultat de les quals no està reflectit en $PA_d(C^i)$. Això és, una acció a_j^i en $\bar{p}_{n^i}$ és sobrerera si

$$a_k^i \notin PA_p(C^i) \wedge \text{result}(a_k^i) \notin PA_d(C^i)$$

Per exemple, suposem de nou el cas g24v1 amb el pla final i patró esmentats abans. Mirant només al pla $PA_p(g24v1)$, trobem que l'acció febre-q podria haver-se evitat: el grau de des-enfocament seria del 16.67% (1/6) i el d'enfocament del 83.33%. Tanmateix, l'execució de febre-q ha permès d'aconseguir el diagnòstic (febre-q, *possible*), resultat que pertany

a $PA_d(g24v1)$. Per aquesta raó febre-q no es pot jutjar com a acció sobrera. El grau de des-enfocament pel cas $g24v1$ és del 0% i, consegüentment, el grau d'enfocament és del 100%. D'aquesta manera contrastem l'opinió d'altres experts (continguda a $PA_d(g24v1)$) amb la de l'expert mestre (proporcionada a $PA_p(g24v1)$), i evitem que BOLERO tingui un *feedback* excessivament restringit.

El grau d'ordenació. El grau d'ordenació indica la prioritat amb la qual s'han executat les diferents accions. Les accions el resultat de les quals són fets difusos amb un grau d'evidència alt haurien de ser executades abans d'accions el resultat de les quals són fets difusos amb menys grau d'evidència⁸. Per exemple, suposem que les accions *pneumococ* i *s-piògen* donen els resultats següents:

$\text{result}(\text{pneumococ}) = \{ (\text{pneumococ}, \text{cert}) \}$

$\text{result}(\text{s-piògen}) = \{ (\text{pneumònia-per-s-piògen}, \text{molt-poc-possible}) \}$

Aleshores l'ordre de les accions [*pneumococ*, *s-piògen*] és correcte, mentre que l'ordre [*s-piògen*, *pneumococ*] és incorrecte.

La determinació del grau d'ordenació de les accions no és molt rellevant, atès que és molt difícil indicar un ordre adequat quan la informació que es coneix sobre un problema (sobretot inicialment) és incompleta. Però ha estat necessari introduir algun criteri d'ordenació per tal de flexibilitzar l'ordre de les accions indicat pel mestre en el patró d'avaluació. Prendre l'ordre adequat de les accions com l'ordre indicat en el patró PA_p resultaria massa estricte.

Per definir el grau d'ordre d'un pla s'ha comparat el pla generat per BOLERO $p_{n_i}^i$ i el pla $PA_p(C^i)$, obtenint un conjunt d'accions O que difereixen en la seva posició. Aleshores el grau d'ordenació es pot descompondre en el *grau d'enriquiment* més el *grau de desencert*, i.e.

$\text{grau d'ordenació} = (\text{grau d'enriquiment}, \text{grau de desencert})$

El grau d'enriquiment és la raó entre el nombre d'accions d' O que han estat ordenades correctament en $p_{n_i}^i$ respecte al nombre total d'accions de $p_{n_i}^i$. El grau de desencert és la raó entre el nombre d'accions d' O que han estat ordenades incorrectament en $p_{n_i}^i$, respecte al nombre total d'accions de $p_{n_i}^i$. El grau d'enriquiment, doncs, ens proporciona informacions sobre el millor comportament de BOLERO respecte als plans construïts pel mestre.

Per exemple, en el cas $g24v1$ BOLERO ha planificat *clamídia* abans que *micoplasma*, al contrari del que assenyala el patró $PA_p(g24v1)$. El resultat de l'acció *clamídia* és (*clamídia*, *possible*), mentre que el resultat de *micoplasma* és (*micoplasma*, *molt-possible*). Aleshores, donat que la evidència dels fets resultants de l'acció *micoplasma* és més gran, és millor executar primer *micoplasma* abans que *clamídia*, com ho fa el mestre a $PA_p(g24v1)$. L'ordre de *clamídia* en el pla final $p_{n_i}^i$ de BOLERO ha estat inadequat, i diem que el grau de desencert és del 16.66% (1/6), i el grau d'enriquiment, del 0%. En cas que *clamídia* hagués obtingut el resultat (*pneumònia-clamídia*, *cert*), aleshores diríem que BOLERO ha millorat l'ordre respecte al mestre, car accions amb més evidència (i.e. més prioritàries) les ha considerat abans. El grau desencert en aquest cas seria del 0%, i el d'enriquiment, del 16.66%.

Un pla és precís quan el grau d'enfocament és del 100% i el grau de desencert és del 0%. Si, a més a més, el grau d'enriquiment és més gran del 0%, aleshores diem que el pla de BOLERO és més precís que el pla de l'expert mestre.

⁸Cal observar que el criteri d'ordenació de les accions està esbiaixat, com també passa amb el grau d'assoliment, pel tipus d'aplicació desenvolupat en BOLERO, i.e. la diagnòsi mèdica: es més important considerar diagnòstics amb més grau d'evidència abans que d'altres amb menys.

4.6 Aprenentatge per l'experiència pròpia

BOLERO és un sistema aprenent i, com a tal, aprèn de l'experiència pròpia tant pel reeiximent dels plans que construeix com de plans imperfectes. Un pla reïx quan el grau d'assoliment del pla és del 100%, el grau d'enfocament també és del 100% i el grau de desencert és del 0% (i.e. és correcte i precís). Quan el grau d'assoliment és més petit del 100%, el grau d'enfocament és més petit del 100%, o el grau de desencert és més gran del 0% (i.e. el pla és incorrecte o imprecís), aleshores diem que el pla és imperfecte. En l'aprenentatge per l'experiència s'han distingit tres mètodes:

1. aprenentatge a partir del reeiximent d'un pla (secció 4.6.1),
2. aprenentatge a partir de plans imperfectes (secció 4.6.2), i
3. aprenentatge de la importància estratègica dels índexs (secció 4.6.3).

El primer mètode és aplicable quan un pla reïx; els dos darrers mètodes són possibles quan un pla és imperfecte. A continuació es descriuen aquest tres mètodes.

4.6.1 Aprenentatge a partir del reeiximent del pla

L'aprenentatge pel reeiximent ocorre quan el pla construït per BOLERO és correcte i precís. Aleshores, i com a conseqüència de la planificació reactiva, el pla pot ser un pla nou no contingut en la memòria de BOLERO (vegeu secció 2.5). L'aprenentatge pel reeiximent dels plans consisteix a incorporar aquest pla nou en BOLERO mitjançant el mètode d'incorporació descrit en la secció 4.3.

4.6.2 Aprenentatge a partir d'un pla imperfecte

L'aprenentatge pel fracàs té lloc quan un pla és incorrecte o imprecís. Si un pla és imperfecte, llavors es demana a l'expert mestre que proporcioni el cas correcte C^c . Cal observar que el patró d'avaluació conté el pla que coincideix amb el pla final del cas C^c que proporciona el mestre (i.e. $PA_p(C^c) = \bar{p}_{n^c}$). Però $PA_p(C^c)$ no conté informació detallada del procés de resolució d'un problema, i.e. la seqüència d'episodis. Per això és necessari demanar el cas C^c al mestre. Aquest cas C^c proporcionat per l'expert mestre s'incorpora en memòria mitjançant el mètode d'aprenentatge per observació explicat en la secció 4.3.

4.6.3 Aprenentatge de la importància estratègica de les generalitzacions

L'aprenentatge de la importància estratègica de les generalitzacions també s'esdevé quan un pla construït per BOLERO és imperfecte. Un pla imperfecte indica que els índexs estan mal definits, ja que s'ha recuperat erròniament el pla imperfecte de memòria i no s'ha recuperat el pla correcte. Aleshores s'imposa una modificació del mecanisme d'indexació per tal d'evitar que futurs plans fracassin.

El mecanisme de recuperació de plans es basa en les funcions m_f i m_s definides a la secció 4.4. La primera funció (m_f) indica la similitud entre fets i generalitzacions. La segona funció (m_s) depèn estretament de la importància estratègica w de cadascuna de les generalitzacions que s'estan comparant. La importància estratègica, o bé la proporciona l'expert mestre, o bé pot ser inicialitzada amb un mateix valor per a totes les generalitzacions. Mitjançant un mètode d'aprenentatge es pot modificar la importància estratègica de les generalitzacions, i d'aquesta manera refinar el mecanisme de recuperació de plans (i.e. adequar la indexació).

El primer pas del mètode de l'aprenentatge de la importància estratègica de les generalitzacions és, com en l'aprenentatge a partir de plans imperfectes, demanar a l'expert mestre el cas correcte C^c (veure l'algorisme de la taula 4.2). En el mètode d'aprenentatge de la importància de les generalitzacions, com en qualsevol altre mètode per experiència de BOLERO, s'ha de tenir present que el pla avaluat $\bar{p}_{n^i}^i$ (que és imperfecte) pot ser un pla contingut o no en memòria. És a dir, la planificació reactiva fa que el pla final $\bar{p}_{n^i}^i$ és un pla existent en memòria o bé és el resultat de l'adaptació de n plans de n nodes $\mathcal{N} = \{N^1, N^2, \dots, N^n\}$ recuperats de memòria. Així, l'algorisme d'aprenentatge de la importància estratègica es basa a distingir aquells nodes de $\mathcal{N}$ activats erròniament en la resolució de problemes ($\mathcal{N}_e$) i els nodes que no han estat activats i haurien d'haver-ho estat (N^{PA}), i alterar la importància estratègica de les seves generalitzacions conseqüentment. L'algorisme es mostra a la taula 4.7.

Definim el node model N^{PA} com el node terminal que té el pla més similar al pla $\bar{p}_{n^c}^c$ del cas C^c seguint el següent procés:

1. Si existeix un node N_i^i en memòria tal que el pla $\bar{p}^i$ coincideix amb el pla del cas C^c (i.e. $checkp_i = \bar{p}_{n^c}^c$), aleshores $N^{PA} = N_i^i$
2. Altrament, si existeix un node N^i tal que el seu pla es pot descompondre en dues parts s i s' , de manera que

$$\bar{p}^i = s; s' \text{ i } \bar{p}_{n^c}^c = s; s''$$

llavors N^i té el pla més similar a C^c i s'escull com a node model N^{PA} . En aquest cas es defineix el conjunt de nodes complementaris $\overline{\mathcal{N}^{PA}}$. Els nodes de $\overline{\mathcal{N}^{PA}}$ són nodes amb accions que no té el pla de N^{PA} però sí el pla $\bar{p}_{n^c}^c$. Per tant, podem suposar que executant el pla de N^{PA} i a continuació els plans de $\overline{\mathcal{N}^{PA}}$ s'aconsegueix la resolució correcta del problema (i.e. $\bar{p}_{n^c}^c$). D'aquesta manera, entre tots els nodes que tenen accions de s'' s'esculleix aquells que tenen menys accions no-necessàries respecte al pla $\bar{p}_{n^c}^c$ per constituir $\overline{\mathcal{N}^{PA}}$.

Per exemple, suposem que el pla final del cas correcte és:

$$\bar{p}_{n^c}^c = [\text{recollida-dades, bact-atip, virus, micoplasma clamídia}] ,$$

I que en memòria hi ha els nodes N_1 , N_2 , i N_3 amb els següents plans:

$$\begin{aligned} N_1 &= [\text{recollida-dades, bact-atip, virus, clamídia}] \\ N_2 &= [\text{recollida-dades, bact-atip, micoplasma, virus}] \\ N_3 &= [\text{recollida-dades, bact-atip, pneumococ, enterobacteriàcies, micoplasma}] \end{aligned}$$

D'acord amb el procediment explicat abans, obtenim que $N^{PA} = N_1$ i $\overline{\mathcal{N}^{PA}} = \{N_2\}$. N_3 podria formar part de $\overline{\mathcal{N}^{PA}}$, car té l'acció micoplasma, que no té N_1 i que sí que és a $\bar{p}_{n^c}^c$. Però N_3 presenta les accions pneumococ i enterobacteriàcies no contingudes a $\bar{p}_{n^c}^c$ i, per tant, si escollim N_3 dins de $\overline{\mathcal{N}^{PA}}$ afavorim l'execució de les accions pneumococ i enterobacteriàcies que no resulten necessàries per al cas C^c .

Un cop determinats els nodes que s'haurien d'haver activat, N^{PA} i $\overline{\mathcal{N}^{PA}}$, hem de distingir els nodes $\mathcal{N}_e$ que s'han activat erròniament del conjunt total de nodes activats $\mathcal{N}$. Diem que un node N de $\mathcal{N}$ ha estat recuperat de memòria erròniament si:

1. No pertanyen a $\{N^{PA}\} \cup \overline{\mathcal{N}^{PA}}$.
2. Algun dels seus ancestres és el node model N^{PA} .
3. Algun dels seus descendents és N^{PA} .

Si algun node compleix alguna de les condicions esmentades, s'inclou dins del conjunt $\mathcal{N}_e$ de nodes recuperats erròniament.

apr-importància-estratègica(C^c :cas, C^i :cas)

- ; C^c : cas proporcionat pel mestre
- ; C^i : cas obtingut de BOLERO
- ; $\mathcal{N}$: conjunt de nodes activats i seleccionats en la resolució de problemes
- ; N^{PA} : node de memòria amb el pla més semblant al pla final de C^c
- ; $\overline{N^{PA}}$:conjunt de nodes complementari a N^{PA} (nodes amb plans que tenen
- ; accions del pla final de C^c i que no estan a $\overline{N^{PA}}$

1. Si existeix N_t^i tal que $\tilde{p}^i = p_{n^c}^c$
2. Llavors $N^{PA} = N_t^i$
3. Altrament Si existeix N^i tal que $p_{n^c}^c = s; s' \wedge \tilde{p}^i = s; s''$
4. Llavors $N^{PA} = N^i$
5. $\overline{N^{PA}} = \text{model-complementari}(s', p_{n^c}^c, \mathcal{N} \setminus \{N^{PA}\})$
6. Per a cada $N^j \in \mathcal{N}$
7. Si $N^j \neq N^{PA} \wedge N^j \notin \overline{N^{PA}} \wedge$
8. $\text{ancestres}(N^j) \cap \{N^{PA}\} = \emptyset \wedge \text{ancestres}(N^{PA}) \cap \{N^j\} = \emptyset$
9. Llavors Per a cada $f_k \in v^j \cap s_{n^i}^i$ tal que $\text{identificador}(f_i) = \text{identificador}(f_k)$
10. Si existeix $f_l \in v^{PA}$ tal que $\text{identificador}(f_l) = \text{identificador}(f_k)$
11. Llavors Si $\text{multivaluat}(f_k)$
12. Llavors Per a cada $v_m \in V_k \setminus V_l$
13. $w_m = \text{actualització-}w(-1, w_m)$
14. Altrament $w_k = \text{actualització-}w(0, w_k)$
15. Altrament $w_k = \text{actualització-}w(-1, w_k)$
16. $Z = Z \cup \{f_k\}$
17. Per a cada $f_j \in v^{PA}$
18. Si existeix $f_k \in Z$ tal que $\text{identificador}(f_j) = \text{identificador}(f_k)$
19. Llavors Si $\text{multivaluat}(f_j)$
20. Llavors Per a cada $v_m \in V_j \setminus V_k$
21. $w_m = \text{actualització-}w(+1, w_m)$
22. Altrament $w_j = \text{actualització-}w(0, w_j)$
23. Altrament $w_j = \text{actualització-}w(+1, w_j)$

model-complementari(p_1 :pla, p^c :pla desitjat, $\mathcal{N}$)→conjunt de nodes

1. Per a cada $N^i \in \mathcal{N}$
2. Si $\tilde{p}^i \cap p_1 = \emptyset$
3. Llavors $\mathcal{N} = \mathcal{N} \setminus \{N^i\}$
4. *escollir-node-amb-pla-més-similar*($\mathcal{N}$, p^c).

Taula 4.7: Algorisme d'aprenentatge de la importància estratègica dels fets.

Aleshores, en la resolució d'un problema es poden distingir el següent conjunt de generalitzacions:

Tipus-I. Generalitzacions que pertanyen a una situació generalitzada v^i de qualsevol node de $\mathcal{N}_e$ i que no pertanyen a N^{PA} :

$$\left(\bigcup_{i=1}^n v^i\right) \setminus v^{PA}$$

Les generalitzacions de tipus I són generalitzacions que han contribuït a la recuperació errònia de nodes de memòria.

Tipus-II. Generalitzacions que no pertanyen a cap situació generalitzada v^i de qualsevol node de $\mathcal{N}$ i que pertanyen a N^{PA} .

$$v^{PA} \setminus \left(\bigcup_{i=1}^n v^i\right)$$

Les generalitzacions de tipus II són generalitzacions que només pertanyen a v^{PA} i que no han tingut prou pes en la mesura m_s (mitjançant la importància estratègica) com per activar el node N^{PA} amb un grau de *matching* suficient.

Tipus-III. Generalitzacions que pertanyen a una situació generalitzada v^i de qualsevol node de $\mathcal{N}$ i que pertanyen a N^{PA} .

$$\left(\bigcup_{i=1}^n v^i\right) \cap v^{PA}$$

Les generalitzacions de tipus III són generalitzacions comunes a N^{PA} i als nodes recuperats erròniament, raó per la qual no poden proporcionar-nos informació per corregir la importància estratègica.

Definits aquests tres conjunt de generalitzacions, l'aprenentatge de la importància estratègica consisteix en els passos següents:

1. Disminuir la importància estratègica dels fets de tipus I.
2. Augmentar la importància estratègica dels fets de tipus II.
3. Deixar invariant la importància estratègica dels fets de tipus III.

La disminució i augment de la importància estratègica de les generalitzacions es realitza mitjançant la següent funció de combinació d'evidències:

$$actualització-w(g_i) = \begin{cases} T(1-n, w_i) & \text{si } g_i \text{ és de tipus I} \\ C(w_i, T(n, w_i)) & \text{si } g_i \text{ és de tipus II} \end{cases}$$

on n és un paràmetre del sistema ($n \in [0, 1]$); $T(x, y)$ és una t-norma; i $C(x, y)$ és una t-conorma dual a $T(x, y)$, i.e. $C(x, y) = 1 - T(1-x, 1-y)$ (apèndix B). En el desenvolupament de BOLERO s'ha pres les següents formes per a les c-normes i t-normes:

$$C(x, y) = x + y - xy$$

$$T(x, y) = xy$$

Es pot comprovar fàcilment que la funció *actualització-w(g_i)*, amb els valors de t-norma i t-conorma proveïts, és una funció contínua en l'interval $[0, 1]$. Com a treball futur de recerca roman pendent experimentar amb altres possibles t-normes i t-conormes.

El paràmetre n indica la proporció que es modifica la importància estratègica. En l'experimentació de BOLERO hem optat per valors petits de n ($n = 0.2$), ja que suposem que canvis petits produïts amb regularitat condueixen a resultats millors que els resultats obtinguts per canvis bruscos.

Amb l'aprenentatge de la importància estratègica dels fets es finalitza la descripció de tots els components del sistema BOLERO. En el proper capítol es proporciona l'avaluació experimental del sistema.

Capítol 5

Avaluació experimental de BOLERO

En aquest capítol es proporcionen els resultats obtinguts en BOLERO en una aplicació del món real. En la primera secció s'introdueix l'aplicació en la qual s'ha posat a prova BOLERO. En la secció 5.2 es defineix el mètode d'avaluació emprat. En la tercera secció es detallen els resultats aconseguits i en la secció 5.4 es realitza una avaluació experimental sobre diferents components del sistema. En la secció 5.5 es proporcionen els arguments que ens permeten hipotetitzar que BOLERO és un planificador independent del domini i, finalment, en la secció 5.6 es relaten les conclusions.

5.1 L'aplicació

En aquest treball hem aplicat BOLERO per a l'aprenentatge i la construcció de plans per a la diagnosi de l'agent microbià que causa la pneumònia en un pacient. Les pneumònies són una malaltia freqüent que requereixen un tractament precoç [166]. Un tractament precoç significa que el metge necessita diagnosticar el microbi que causa la infecció prenent avantatge de la informació que coneix sobre el pacient, i abans de conèixer els resultats d'algunes proves de laboratori. Encara que poden realitzar-se exploracions complementàries, el metge normalment no coneix l'etiologia de la malaltia. La informació incompleta i incerta són característiques de la diagnosi de pneumònies que perseveren al llarg de tot el procés terapèutic. La informació incompleta i incerta apareix per diferents motius [166]: la qualitat de la informació (la interpretació d'una radiografia de tòrax pot variar d'acord amb la projecció, la qualitat de la placa, etc.), la complexitat de la situació en què s'interrelacionen diversos factors (tractar una meningitis bacteriana en un malalt suposadament al·lèrgic als betalactàmics), les dades que manquen (un nen petit o un adult incapaç de comunicar-se no estan en condicions de subministrar informació), les dades incertes pròpies de l'exercici de la medicina (l'avaluació del dolor, per exemple, és una informació moltes vegades subjectiva). La planificació reactiva és especialment important en aquesta aplicació on el problema de la informació incompleta està aguditzat per la necessitat de prendre decisions i emetre judicis que afecten la vida dels malalts. S'han de prendre decisions, i l'experiència del metge en casos anteriors juga un paper fonamental en la diagnosi del pacient.

El punt de partida han estat 79 casos proporcionats per un mestre (un metge expert en la diagnosi de pneumònies, vegeu apèndix C) i corresponen a casos reals presos de quatre

hospitals. La selecció dels casos va ser realitzada per l'expert, que la va fer de forma aleatòria estratificada a partir de grups de diagnòstics de l'Organització Mundial de la Salut [166]

Per a la representació del domini de pneumònies es disposa de 460 fets. Cada cas té una mitjana d'uns 88.23 fets, el 22,87% dels quals són fets amb valor *desconegut*, i.e. valors que no es coneixen al llarg de la diagnosi del pacient. Les característiques dels casos i, en general, les del domini ens han dut a realitzar una anàlisi de les condicions d'aprenentatge o contingència el resultat de la qual es presenta a la secció 5.1.1. També s'ha estudiat la presència de fets irrelevants i coneixement irrelevant que determinen l'aprenentatge (secció 5.1.3).

Per executar els plans construïts per BOLERO, hem disposat de PNEUMON-IA* [166], un sistema basat en regles capaç d'interpretar i executar els plans (per a més detalls, vegeu apèndix C). Per a l'avaluació dels plans generats per BOLERO s'han construït patrons d'avaluació basats en l'opinió de cinc experts, uns amb millors coneixements que altres. A cada expert se li ha proporcionat l'historial clínic dels 79 pacients i n'ha donat la seva diagnosi. Atès que en el domini de pneumònies no hi ha una solució única a la diagnosi d'un pacient, les opinions dels cinc experts i la de l'expert metge no són idèntiques la major part de les vegades. Això fa que es plantegi el problema de les classes alternatives, que s'explica en l'apartat 5.1.2 d'aquesta secció i que té conseqüències en l'aprenentatge per l'experiència pròpia (veure secció 5.4.2).

5.1.1 Contingència

Estudis psicològics en l'aprenentatge parlen de la contingència o les condicions en què pot tenir lloc l'aprenentatge [138]. Donats una classe c i un índex (*estímul*) de la classe i , la contingència es defineix en termes bayessians com a $p(c|i) > p(c|\neg i)$ [138]. És a dir, la probabilitat que es reconegui una classe c amb la presència d'un índex i ha de ser superior a la probabilitat que es reconegui la classe c amb l'absència d' i . El nivell de sorolls que apareixen en les dades dels exemples i fins i tot en la informació de classes és important en la contingència. Per un costat, si el soroll en els índexs és del 50% (és a dir, hi ha un 50% de possibilitats que l'índex i sigui en realitat $\neg i$), l'aprenentatge és impossible. Per altra banda, el soroll en la informació de la classe (és a dir, suposar que l'oracle proporcioni classificacions incorrectes com dir que un exemple positiu és negatiu) fa més difícil l'aprenentatge [129].

Per tal de realitzar una anàlisi de contingència sobre els casos de PNEUMON-IA* s'ha estudiat quina és la relació entre la presència d'un fet i l'esdeveniment d'un pla (i.e. classe). De plans possibles, n'hi ha molts, per la qual cosa hem simplificat l'anàlisi estudiant només quina és la relació entre la presència d'una acció a en un pla final i un fet f . Suposem, per exemple, que tenim la probabilitat $p(a|f)$ que l'acció a aparegui en un pla donada l'evidència f , i $p(a|\neg f)$ la probabilitat que l'acció a no s'executi davant l'absència de f . Aleshores, la contingència es pot analitzar en quatre aspectes [138]:

1. lliure de sorolls (*noise-free*): la presència d'un fet és condició necessària i suficient per a l'execució d'una acció:

$$p(a|f) = 1 > p(a|\neg f) = 0$$

2. reforçament parcial (*partial reinforcement*): la presència d'un fet és una condició necessària però no suficient per a l'execució d'una acció

$$1 > p(a|f) > p(a|\neg f) = 0$$

3. advertiment parcial (*partial warning*): la presència d'un fet és condició suficient per a l'execució de l'acció, però si no és present l'acció pot també executar-se:

$$p(a|f) = 1 > p(a|\neg f) > 0$$

4. indicatiu inapropiat per a l'aprenentatge: l'execució d'una acció és possible tant en la presència com en l'absència d'un fet, i es desconeix en quina proporció:

$$1 > p(a|f)? \quad p(a|\neg f) > 0$$

Les quatre aparences de la contingència són possibles en els casos disponibles per a l'aprenentatge en la diagnosi de pneumònies. A continuació es mostra un exemple de cadascuna de les quatre possibilitats:

1. lliure de sorolls: sempre que apareix el fet **mostra-líquid-pleural** s'executa l'acció **embassament**; si el fet **mostra-líquid-pleural** no apareix, aleshores l'acció no s'executa.

$$p(\text{embassament} | (\text{mostra-líquid-pleural}, sf)) = 1 > p(\text{embassament} | \neg(\text{mostra-líquid-pleural}, sf)) = 0$$

2. reforçament parcial: quan hi ha un **embassament-pleural** en les radiografies, pot executar-se l'acció **embassament**; si no hi ha un **embassament-pleural**, no s'executarà mai l'acció **embassament**.

$$1 > p(\text{embassament} | (\text{raigs-X}, \text{embassament-pleural})) > p(\text{embassament} | \neg(\text{raigs-X}, \text{embassament-pleural})) = 0$$

3. advertiment parcial: quan un pacient té com a antecedents patològics una **proteïnosi-alveolar** segur que l'acció **nocàrdia** s'executa, car la **proteïnosi alveolar** és un antecedent de risc per a una pneumònia per **nocàrdia**. Però la pneumònia per **nocàrdia** pot estar causada per altres factors com una **sarcoïdosi**. Per tant, en l'absència del fet **proteïnosi-alveolar** també es pot executar **nocàrdia**.

$$p(\text{nocàrdia} | (\text{antecedents}, \text{proteïnosi-alveolar})) = 1 > p(\text{nocàrdia} | \neg(\text{antecedents}, \text{proteïnosi-alveolar})) > 0$$

4. indicatiu inapropiat: el fet que el pacient pateixi una pneumònia atípica a vegades determina l'execució de l'acció **carinii** (i.e. determinar si la pneumònia està causada per l'agent microbià *Pneumocystis carinii*), però no sempre. La relació entre l'execució de **carinii** i el fet **pneumònia-atípica** no està determinada.

$$1 > p(\text{carinii} | (\text{pneumònia-atípica}, \text{possible}))? \quad p(\text{carinii} | \neg(\text{pneumònia-atípica}, \text{possible})) > 0$$

La situació lliure de sorolls és realment poc freqüent. No es pot determinar quins fets determinen l'execució de les accions. Les accions i, per tant, els plans es classifiquen més aviat per combinacions de fets, cosa que ens fa pensar en el problema del context (vegeu secció 1.1.1). Però els casos de què disposem i sobre els quals s'ha fet l'estudi de contingència són aquells amb què un expert humà es troba en la realitat, i per tant casos amb els quals BOLERO ha de realitzar l'aprenentatge. Altrament, l'expert no hauria de proporcionar casos, sinó exemples ben explicitats. Proporcionar exemples i no casos amb què diàriament treballa l'expert ens allunyaria d'un dels objectius de BOLERO: proveir d'un mecanisme d'adquisició de coneixements amb el qual l'expert humà proporcionï els coneixements amb el menys esforç possible. Els resultats de la secció 5.3 demostren experimentalment que la planificació basada en casos resol tàcitament en l'organització dels casos en memòria el problema del context [66].

5.1.2 El problema de les classes alternatives

Els experts presenten diferents opinions, entre ells i respecte del mestre, i, consegüentment els casos poden ser classificats de forma diferents segons l'opinió de cada expert. Un expert e_1 pot construir el pla p_1 per un cas, mentre que per un altre expert e_2 pot suggerir el pla p_2 pel

(a) $\mathcal{C}$	(b) $\mathcal{C}'$		
c recollida-dades bact-atip pneumococ legionella enterobacteriàcies anaerobi hemòfil estafilococ s-piògen meningococ branhamella tuberculosi	c'_1 recollida-dades bact-atip pneumococ legionella enterobacteriàcies anaerobi hemòfil estafilococ s-piògen meningococ branhamella tuberculosi	c'_2 recollida-dades bact-atip pneumococ legionella enterobacteriàcies anaerobi hemòfil estafilococ s-piògen meningococ branhamella tuberculosi micoplasma virus clamídia febre-q	c'_3 recollida-dades bact-atip pneumococ micoplasma virus clamídia legionella enterobacteriàcies s-piògen hemòfil branhamella febre-q tuberculosi anaerobi estafilococ meningococ
b02v1 b14v1 b04v1 b05v1	b02v1 b14v1	b04v1	b05v1

Figura 5.1: Diferents classificacions de 4 casos. (a) Classificació $\mathcal{C}$ basada exclusivament en el mestre. (b) Classificació $\mathcal{C}'$ basada en l'opinió de l'expert mestre i 5 experts més. En la classificació $\mathcal{C}$ els quatre casos són a la mateixa classe, mentre que en $\mathcal{C}'$ els casos es classifiquen entre tres classes diferents (per tal com tenen tres plans diferents).

mateix cas. Així, tenint en compte l'opinió de l'expert mestre, els casos es poden classificar, d'acord amb els plans finals, en una classificació $\mathcal{C}$, i tenint en compte, a més a més, l'opinió de cinc experts, podem classificar els casos en una classificació diferent $\mathcal{C}'$ (figura 5.1). És interessant observar que, atès que en el mètode d'aprenentatge per observació els plans són proporcionats per l'expert mestre, i en l'aprenentatge per l'experiència pròpia els plans poden ser diferents als de l'expert mestre a causa de l'avaluació multiexpert, les classificacions dels casos que s'aprenen amb cadascun dels dos mètodes són diferents.

És de suposar que una classificació $\mathcal{C}$ dels casos obtinguda a partir de l'expert mestre tingui menys nombre de classes que una classificació $\mathcal{C}'$ basada en 5 experts, de manera que l'opinió de més experts contribueix a un refinament en la classificació dels casos. En aquest sentit, una classe c en $\mathcal{C}$ que agrupa casos d'acord amb el seu pla final p pot contenir $|c|$ elements, mentre que una classe c' en $\mathcal{C}'$, que agrupa casos amb el mateix pla p , pot tenir un nombre menor d'elements $|c'| \leq |c|$. Això es causat pel fet que la classe c en $\mathcal{C}$ es refina en diferents classes $c_1, c_2, \dots, c_n$ en $\mathcal{C}'$, on eventualment existeix una classe c_i tal que $c_i = c'$. Aleshores diem que $c_1, c_2, \dots, c_n$ són classes alternatives.

El problema de les classes alternatives rau en el fet que les dades proporcionades en els casos les ha recollides el mestre i són vàlides per classificar els casos d'acord amb $\mathcal{C}$. Ara bé, quan es fa una nova partició dels casos sense variar la informació podem trobar-nos que no podem classificar adequadament un cas. Per exemple, suposem que els casos "b05v1",

"b02v1", "b14v1" i "b04v1" tenen el mateix pla final

[recollida-dades, bact-atip, pneumococ, legionel·la,
enterobacteriàcies, anaerobi, hemòfil, estafilococ,
s-piògen, meningococ, branhamel·la, tuberculosi]

Malgrat que el pla final és el mateix en tots els casos, les dades dels casos (i.e. fets) difereixen. Així, les situacions finals dels casos "b15v1" i "b02v1" es diferencien en els següents quatre fets:

$$\begin{aligned} S^{b05v1} \setminus S^{b02v1} &= \{(\text{antecedents-interès, desconegut}), \\ &\quad (\text{antecedents, desconegut}), \\ &\quad (\text{dades-RX, inter}), \\ &\quad (\text{pneumònia-bacteriana, mod-possible})\} \\ S^{b02v1} \setminus S^{b05v1} &= \{(\text{antecedents-interès, si}), \\ &\quad (\text{antecedents, fumador}), \\ &\quad (\text{dades-RX, \{vmaldef, confluent\}}), \\ &\quad (\text{pneumònia-bacteriana, força-possible})\} \end{aligned}$$

En la classificació C de la figura 5.1a els casos "b05v1", "b02v1", "b14v1", i "b04v1" pertanyen a una mateixa classe c d'acord amb l'opinió de l'expert mestre. En la figura 5.1b els mateixos casos s'han classificat seguint l'opinió de l'expert mestre i 5 experts més. Així, els casos pertanyen a tres classes diferents: "b02v1" i "b14v1" pertanyen a la classe c'_1 , "b04v1" pertany a la classe c'_2 , i "b05v1" pertany a la classe c'_3 . Suposem ara que el cas "r12v1" s'hagués de classificar com el cas "b14v1", és a dir, a c en C i/o amb c'_1 en C' ; altrament el cas "r12v1" es classificarà erròniament. Però tenim que el cas "r12v1" es més semblant al cas "b04v1" que al cas "b14v1". Conseqüentment, el cas "r12v1" s'assigna correctament a la classe c en C , però erròniament a la classe c'_2 de C' (figura 5.1b). Existeixen dues classificacions possibles dels casos, C i C' , però la informació dels casos no varia. Donat que els casos estan proveïts per l'expert mestre, qui configura la classificació C , és possible classificar els casos correctament en C . Però en la classificació C' no s'ha enriquit la informació i no és possible distingir entre les classes alternatives c'_1 , c'_2 , i c'_3 . Aquest és el problema de les classes alternatives: la impossibilitat de discriminar entre elles a l'hora de classificar un nou element. Aquest problema ha determinat en algunes ocasions la consecució de resultats no desitjats, ja que els plans que BOLERO aprèn per observació es basen en una classificació C , però en l'aprenentatge per l'experiència BOLERO aprèn una classificació diferent C' (vegeu secció 5.4.2).

5.1.3 Fets irrelevants i coneixements irrelevants

Altres característiques a tenir en compte, a més a més del soroll dels casos, que influeixen l'aprenentatge són la presència de fets irrelevants i de coneixements irrelevants [77]. Els fets irrelevants són aquells que apareixen en els casos però que no són rellevants per a la diagnosi del pacient. Per exemple, el nom del malalt és un fet irrelevant, mentre que les dades de les radiografies són fets rellevants. En el domini de pneumònies és difícil establir quins fets són irrelevants. Per exemple, la data en què un malalt ha contragut la malaltia (i.e. 11 de febrer de 1990) pot semblar irrelevant però no ho és. Les pneumònies es presenten amb més freqüència en unes èpoques determinades de l'any que en altres i, per tant, la data d'inici de la pneumònia, que permet deduir l'època de l'any en qüestió, és una informació rellevant.

Entre els fets irrelevants s'han de distingir els que són irrelevants respecte als coneixements del domini i els que ho són respecte als coneixements de planificació. Per exemple, els fets edat, sexe, dona-embarassada, vell són fets rellevants en els coneixements

de domini, però només els fets vell i dona-embarassada són rellevants des del punt de vista estratègic (i.e. dels coneixements de planificació). La presència de fets estratègicament irrelevants en el procediment de diagnostic de pneumònies perjudiquen les capacitats d'aprenentatge i el rendiment de BOLERO com es demostra amb l'experiment de la secció 5.4.7.

Els coneixements irrelevants són coneixements que utilitza el mètode d'aprenentatge però que en realitat no repercuteixen positivament en el funcionament del sistema, ans al contrari. Per exemple, en PNEUMON-IA* s'usa una xarxa de coneixements del domini en què hi ha nou tipus de relacions entre fets: *pertany*, *inclosa*, *tipus-de*, *membre-de*, *precisa-de*, *especificació-de*, *explicat-per*, *valor-de*, *més-informatiu*. Les relacions entre fets s'utilitzen en els mètodes de generalització i en la propagació de l'activació (definites en les seccions 4.3.2 i 4.4.1). La xarxa de coneixements de domini és, doncs, força extensa i és interessant conèixer si totes les relacions són útils o si n'hi ha algunes d'irrellevants. Per exemple, la relació *més-informatiu* s'ha considerat irrellevant, ja que no és una relació d'abstracció útil ni per al mètode de generalització ni per al mètode de propagació d'activació, i per tant no s'ha usat. Els coneixements irrelevants poden jugar un paper decisiu en el rendiment del sistema, car el seu tractament pot resultar més costós que els beneficis que comporta.

5.2 Mètode d'avaluació

El mètode d'avaluació és experimental [94]. Una prova o experiment consisteix a utilitzar BOLERO per l'aprenentatge plans a partir de $n-1$ casos del total de n casos disponibles en una seqüència $S = [C^1, C^2, \dots, C^n]$ (seqüència d'avaluació) i mesurar la identificació i predicció del sistema. Altres mesures que han estat considerades en un segon terme són el temps d'execució, a més de l'amplitud i la profunditat de l'arbre de plans generat. La identificació i la predicció ens informen de la qualitat dels plans, mentre que la resta de mesures informen de l'esforç realitzat per construir els plans.

Identificació. Suposant que un problema π^i és solucionat per l'expert mestre en un moment t constituint el cas C^i , i que en un moment posterior t' BOLERO aprèn per observació C^i , la identificació expressa la correctesa i precisió¹ del sistema en resoldre el problema π^i en un temps t'' posterior a t' . En unes altres paraules, la identificació ens diu si BOLERO és capaç de resoldre satisfactòriament un problema "idèntic" a un problema anterior, és a dir una experiència (o cas) apresada amb anterioritat. D'aquesta manera, quan diem que el grau d'assoliment de BOLERO en la identificació és del 100% (com ho veurem en la secció 5.3.1), volem dir que BOLERO sempre soluciona amb èxit problemes idèntics a experiències anteriors.

Predicció. La predicció indica la correctesa i precisió del sistema en resoldre un problema π^i del qual no hi ha un cas idèntic en memòria. Això vol dir que BOLERO no ha après en cap moment anterior a l'actual la solució de π^i , que és, per tant, una experiència nova. Quan en els resultats de la secció 5.3 es parli que el grau d'assoliment de BOLERO en la predicció és del 98,30%, s'ha d'interpretar que davant d'una nova experiència els plans que BOLERO construeix contenen el 98,30% de les accions que dicta l'expert mestre.

¹D'acord amb la definició de correctesa explicitada en la secció 4.5, la correctesa fa referència al grau d'assoliment, i la precisió al grau d'enfocament i al grau d'ordenació

La identificació del sistema es mesura utilitzant el cas C^1 (primer element de la seqüència), i la predicció amb el darrer, C^n . Tant C^1 com C^n s'anomenen casos de tests, i la resta de casos que componen una seqüència d'avaluació són els casos d'entrenament². En particular, quan el mètode d'aprenentatge utilitzat és l'aprenentatge per observació, els casos d'entrenament s'anomenen també casos d'instrucció. Les mesures del sistema es prenen quan BOLERO ha après els k primers casos $C^1, \dots, C^k$; després, quan ha après $k + l$; després, quan ha processat $k + 2l$ casos d'entrenament; i així successivament fins que ha après els $n-1$ casos. D'aquesta manera es pot observar com varia el comportament del sistema en funció del nombre de casos d'entrenament.

Per garantir que els resultats obtinguts no depenen de l'ordre amb què els casos d'entrenament són proporcionats a BOLERO ni del cas en particular sobre el qual s'estan avaluant les mesures, s'han generat aleatòriament m diferents seqüències d'avaluació. De cada una de les seqüències d'avaluació s'ha mesurat la identificació i la predicció, i els resultats que es presenten en la secció 5.3 fan referència a la mitjana dels resultats aconseguits en totes les proves.

Cal remarcar que les seqüències d'avaluació s'han generat aleatòriament, però no totes les seqüències han estat vàlides a causa de la presència de casos que anomenem anòmals. Un cas C^i és anòmal en una seqüència S de n casos si existeix una acció a en el pla final $p_{n,i}^i$ de C^i tal que no existeix un altre cas C^j a S amb un pla final $p_{n,j}^j$ que contingui a . És a dir, a C^i hi ha l'únic exemple d' a . Un cas anòmal no pot ser un cas de test, perquè aleshores l'avaluació del sistema no fóra objectiva: si BOLERO no coneix res sobre una acció, no la podrà planificar mai. Així, hem trobat que el cas "g25v1" contenia l'únic exemple de les accions citomegalovirus i criptococ, i s'ha considerat un cas anòmal³. Per tant, aquests casos anòmals o pitjors casos no s'han considerat en els experiments i les corbes d'aprenentatge que es presenten en aquest capítol estan basades en els resultats aconseguits en casos mitjans (*average-case analysis* [44]).

L'avaluació experimental s'ha realitzat sobre diferents configuracions de BOLERO (vegeu capítol 4). És a dir, donada la modularització del sistema envers els mètodes que implementa, una configuració és una combinació dels components de BOLERO. En primer lloc s'ha definit una configuració bàsica amb la qual BOLERO aprèn exclusivament per observació i genera plans. Els resultats de la configuració bàsica estan proporcionats en la secció 5.3. A continuació s'han desenvolupat altres configuracions afegint altres components, com l'aprenentatge per experiència, o modificant característiques, com la importància estratègica dels fets. Les m seqüències de proves són experimentades en totes les configuracions. Avaluant BOLERO en diferents configuracions podem analitzar l'impacte dels components i de les decisions de disseny preses en el desenvolupament de BOLERO en la seva eficiència i resultats. Aquest procés d'avaluació mitjançant l'eliminació o substitució de components del sistema es coneix també com a ablació (*ironing, ablation, o lesion studies* [44, 45, 77]). En la secció 5.4 es mostra l'ablació del sistema quant a l'avaluació multiexpert, l'aprenentatge per experiència, la importància estratègica dels fets, el paràmetre *mdf* i el biaix conservador. Un darrer aspecte avaluat en la secció 5.4 fa referència a les característiques de l'aplicació. Es tracta de l'avaluació de la incidència del nombre de fets definits en l'aplicació en l'eficiència del sistema.

²La tècnica d'avaluar un algorisme d'aprenentatge consistent a fer aprendre al sistema tots els casos menys uns, i amb el que resta avaluar la bondat del sistema, es coneix també com el mètode de *leave one out*.

³S'ha estés el conjunt de casos anòmals a aquells que contenen una acció en el pla final que apareixia en menys d'un 5% dels casos. Per exemple, el cas "m11v1" és un cas anòmal, car conté l'acció *asper*, que apareix només en 2 dels 79 casos.

5.3 Resultats

A continuació es presenten els resultats obtinguts amb BOLERO en l'aprenentatge de plans i planificació de diagnòstics de pneumònia. El mètode emprat per a l'aprenentatge de plans ha estat l'aprenentatge per observació. A més a més s'han efectuat les assignacions següents:

- La variable λ definida pel paràmetre mdf és 2; això vol dir que $mdf = 1 - e^{-2/x}$
- Tots els fets tenen la mateixa importància estratègica (i.e. $w_i = 0.5 \quad \forall f_i \in F$).
- Nombre total de casos: $n = 79$
- Nombre de seqüències d'avaluació realitzades: $m = 10$. La primera mesura s'ha pres quan BOLERO ha processat 8 casos d'aprenentatge ($k = 8$), i les següents cada vegada que BOLERO processava 5 casos més ($l = 5$).

Aquesta configuració de BOLERO és la que anomenem la configuració *bàsica*.

5.3.1 Identificació

El grau d'assoliment de BOLERO en la identificació és del 100%. També el grau d'enfocament de BOLERO és del 100% i els graus d'ordenació, tant el d'enriquiment com el de desencert, són del 0% (i.e. no es produeixen variacions en l'ordre de les accions respecte al pla proposat per l'expert mestre). Per tant, els plans que genera BOLERO en la identificació són correctes i precisos. Això vol dir que un cas que és idèntic a un cas en memòria sempre és reconegut com a tal, i el cas en memòria sempre és seleccionat respecte a tots els altres. Com que BOLERO aprèn només per observació en la configuració bàsica, podem dir que BOLERO resol un cas idèntic a un en memòria exactament tal i com ho fa el mestre. És important remarcar que el nombre de casos en memòria no afecta la identificació del sistema, com ocorre en altres sistemes, com per exemple GRANT [45]. L'explicació és la següent. Disposant d'un nombre més elevat de casos d'entrenament, el nombre de nodes augmenta, tal i com ho mostra aquest gràfic:

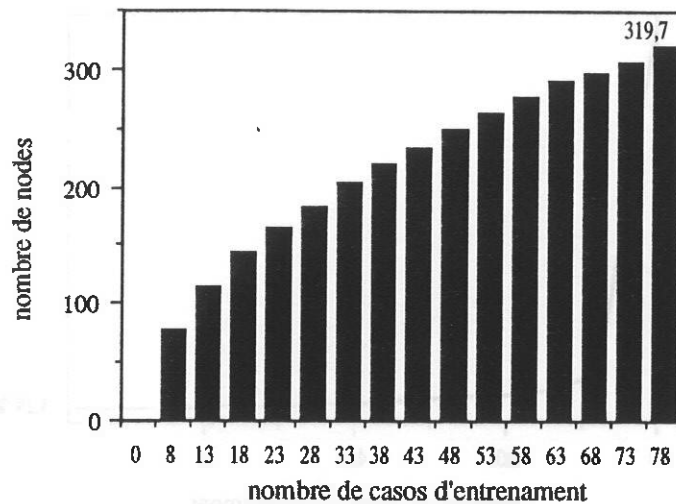


Figura 5.2: Variació del nombre de nodes creats en memòria en funció del nombre de casos d'entrenament. Cal observar que el nombre total de nodes no és elevat si es compara amb el nombre total de nodes possibles. Amb n accions diferents, i suposant que la primera acció de tots els plans és la mateixa, hi hauria $1 + \sum_{j=1}^n \left(\prod_{i=1}^j (n - i) \right)$ nodes no-terminals i la mateixa quantitat de nodes terminals (un descendent per a cada node no terminal). Això significa una quantitat superior a 2^n per a l'aplicació de PNEUMON-IA, on el nombre d'accions és 24.

Un augment del nombre de nodes pot suggerir que en la resolució de problemes s'utilitzi un nombre més elevat de nodes. Usar més nodes per resoldre un problema significa construir plans més llargs i, en conseqüència, es pot produir una pèrdua en el grau d'enfocament. Afortunadament hem observat que això no és així en BOLERO. Hem definit el nombre de nodes en execució per un cas com la raó entre el nombre de nodes que intervenen en la solució d'un problema (i.e. els plans dels quals han estat executats) i el nombre total de nodes que hi ha en memòria, i hem calculat el percentatge de nodes en execució per cas. En el gràfic de la figura 5.3 es mostra com el percentatge de nodes en execució disminueix en funció del nombre de casos d'entrenament.

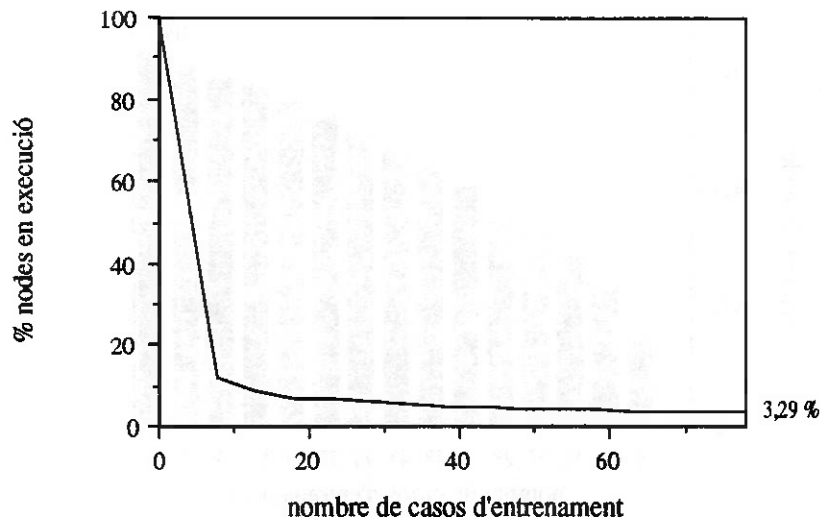


Figura 5.3: Variació del percentatge de nodes en execució en funció del nombre de casos d'entrenament.

Això significa que les generalitzacions de cada node són cada cop més precises i diferenciades, raó per la qual amb pocs nodes es troba la solució d'un problema. És interessant observar que es recupera sempre més d'un node en la resolució d'un cas. Allò esperat en la identificació fóra recuperar un únic node, en particular el node N_t^i que conté el pla final del cas. Però això significaria que des de l'inici de la resolució d'un problema es disposi de tota la informació continguda a v^i . I tenir tanta informació des de l'inici està en contradicció amb les característiques del nostre domini, és a dir, no és possible. Almenys dos nodes intervenen en la resolució d'un cas. El primer node el pla del qual s'executa és el node arrel, que conté el pla per defecte. El pla per defecte és un pla parcial constituït per una única acció: **recollida-dades**. L'execució del pla per defecte fa possible l'obtenció de nova informació que permet recuperar nodes de memòria i ampliar nous plans. El segon node el pla del qual s'executa és un pla recuperat de memòria. Suposem que el primer node recuperat de memòria és un node terminal, N_t , a on està emmagatzemat el cas idèntic al que es vol resoldre. Aleshores si BOLERO executa completament el pla final associat a N_t , BOLERO pot aturar la resolució del problema havent executat el pla de 2 nodes: el node arrel i el node terminal N_t . El que ocorre és que abans de recuperar el node terminal N_t es recuperen nodes no-terminals antecessors de N_t , amb plans parcials.

5.3.2 Predicció

Quant a la predicció, BOLERO, com era d'esperar, no resulta tan bo com la identificació, però és realment ben satisfactori. La predicció del sistema l'hem desglossat en diferents gràfics, distingint el que és la correctesa o grau d'assoliment d'un cas i la precisió (grau d'enfocament i grau d'ordenació). El grau d'assoliment en la predicció evoluciona tal i com mostra el gràfic següent:

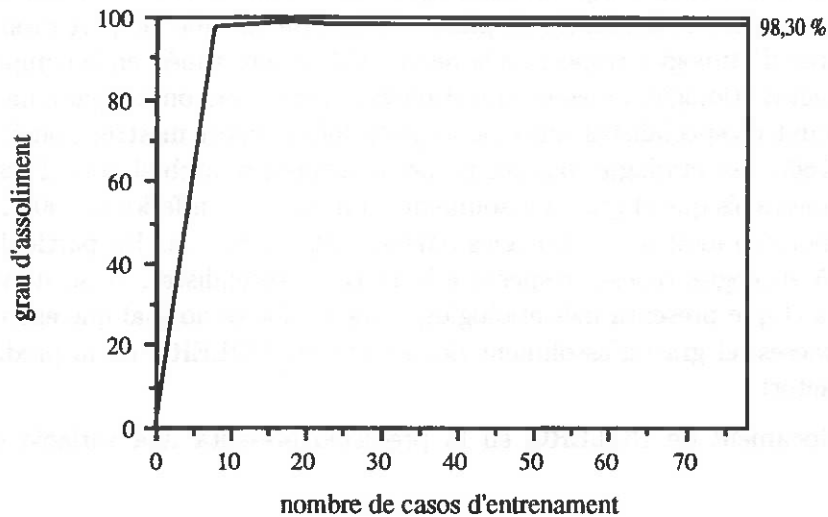


Figura 5.4: Configuració bàsica: grau d'assoliment en la predicció.

Per comprendre els resultats s'ha de tenir en compte que inicialment BOLERO no té cap coneixement sobre plans, ni coneix quines són les accions possibles pel domini en el qual està treballant. De fet, inicialment només coneix el pla per defecte [recollida-dades]. La resta de les accions són apreses a mesura que s'incorporen casos en memòria, tal i com es mostra a continuació:

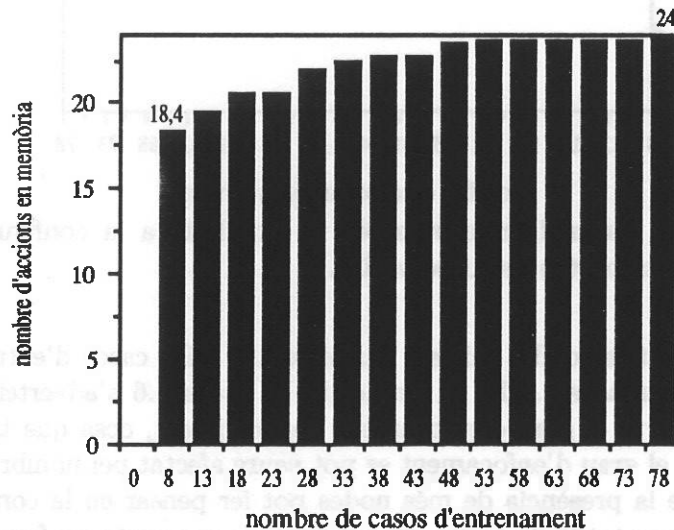


Figura 5.5: Nombre d'accions en memòria en funció del nombre de casos d'entrenament.

En el gràfic de la figura 5.5 s'observa que els plans finals dels casos són llargs, en el sentit que contenen moltes accions. Car amb la incorporació de 8 casos en memòria, BOLERO ja coneix el 76,67% de les accions possibles en la tasca de diagnosticar pneumònies (24).

El fet que el grau d'assoliment sigui del 98,30% també ve justificat per les característiques del nostre expert mestre. Hem comparat els diagnòstics del nostre expert mestre amb la resta d'experts que intervenen en l'avaluació de plans i hem observat que l'expert mestre presenta un nombre més gran d'etiologies respecta a la resta [167]. A més a més, en la comparació entre experts s'han detectat etiologies omeses: una etiologia es considera omesa per a un determinat cas quan del conjunt d'especialistes (els cinc experts més l'expert mestre) només n'hi ha un que no la dona [166]. Les etiologies omeses poden relacionar-se amb el grau d'assoliment: si hi ha etiologies omeses, és que el grau d'assoliment d'un expert és inferior al 100%. Existeixen experts amb un nombre molt alt d'etiologies omeses (19, 12, 8, ...). En particular, l'expert mestre presenta 5 etiologies omeses respecte a la resta d'especialistes. Si se suma el fet que l'expert mestre és el que presenta més etiologies, i que també és normal que entre experts hi hagi etiologies omeses, el grau d'assoliment del 98,30% de BOLERO en la predicció es pot considerar satisfactori.

El grau d'enfocament de BOLERO en la predicció presenta una variació com la que segueix:

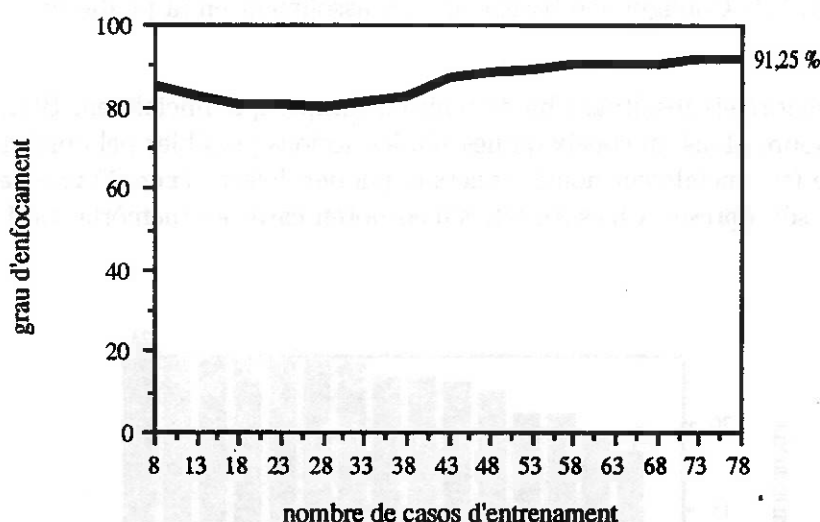


Figura 5.6: Grau d'enfocament en la predicció a la configuració bàsica (només aprenentatge per observació).

No hem proporcionat les dades inicials (dades entre 0 i 8 casos d'entrenament) perquè aquestes no són representables⁴. En el gràfic de la figura 5.6 s'adverteix que el nombre de casos apresos no afecta el grau d'enfocament negativament, cosa que també passa en la identificació. És a dir, el grau d'enfocament es pot veure afectat pel nombre de nodes que hi ha en memòria, ja que la presència de més nodes pot fer pensar en la construcció de plans més llargs. Però el percentatge de nodes en execució no augmenta en funció del nombre de nodes en memòria, ans al contrari. Si observem en el gràfic de la figura 5.7 el percentatge del nodes en execució disminueix en funció del nombre de casos d'entrenament.

⁴Si suposem que BOLERO no té cap coneixement de com dur a terme la planificació, no planifica res. El grau d'enfocament no és aplicable al pla nul.

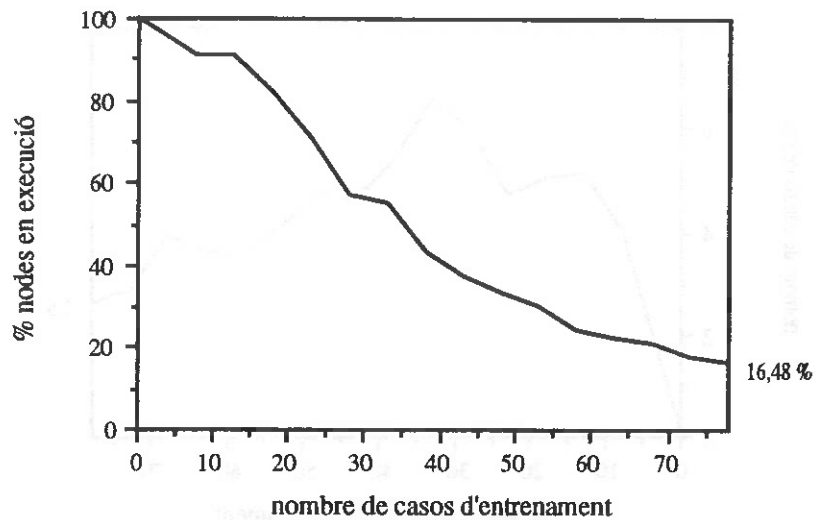


Figura 5.7: Variació del percentatge del nombre de nodes en execució en funció del nombre de casos d'entrenament.

El percentatge del nombre de nodes en execució disminueix en funció del nombre de casos d'entrenament, de manera que cada cop intervenen menys nodes en la resolució d'un cas. En considerar menys nodes en la resolució d'un cas, hi ha menys opcions d'incloure accions no necessàries (espúries o sobreres) en els plans. Això fa que el grau d'enfocament augmenti en funció del nombre de casos en memòria. A més a més, podem hipotetitzar que, disposant d'un nombre més elevat de casos d'entrenament, el nombre de nodes en execució per cas ha de disminuir i, en conseqüència, el grau d'enfocament ha d'augmentar encara més. El resultat actual, un grau d'enfocament del 91,25% en predicció, però, és ben satisfactori.

El grau d'enfocament també està relacionat amb el nombre de canvis de plans que es produeixen durant la resolució d'un problema, cosa que anomenem *inconstància*. Per canvi de pla volem dir que el pla p_i donada la situació s_i , no presenta cap relació d'inclusió amb el pla p_{i-1} executat en la situació s_{i-1} . Si el comportament de BOLERO resulta inconstant, significa que no executa completament un pla final, i.e. un pla que porti a una solució. Si no s'executa un pla final, és molt difícil que es compleixin les condicions de terminació (perquè comencin a activar-se, tal i com ha estat detallat en la secció 4.4.3, s'ha de completar un pla final). Aleshores la resolució d'un problema és més llarga del que seria desitjable. La mitjana de canvis de pla que té lloc en la resolució d'un nou problema, cosa que podem anomenar *grau d'inconstància*, evoluciona de la manera següent:

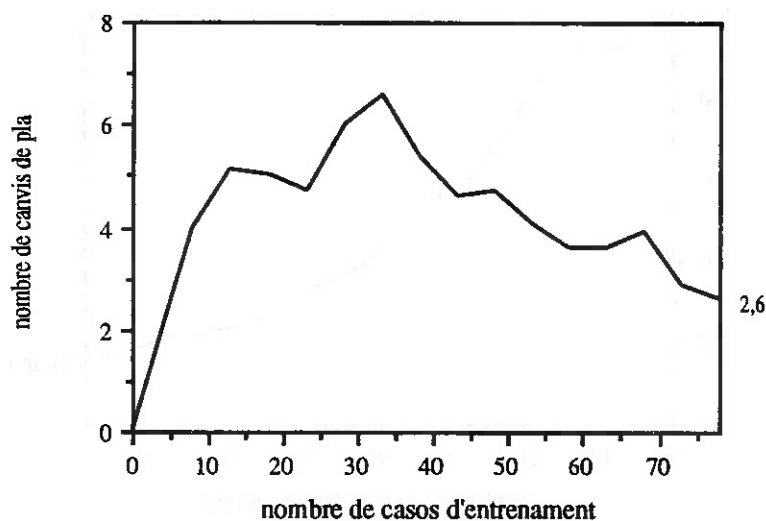


Figura 5.8: Evolució de la mitjana de nombre de canvis de pla que té lloc en la resolució d'un problema respecte al nombre de casos en memòria (i.e. casos d'entrenament).

Observem que inicialment (en els 25 primers casos) BOLERO té un grau d'inconstància al voltant del 4-5 canvis de pla per cas. Això es causat pel fet que BOLERO té pocs casos en memòria (vegeu figura 5.10) i, consegüentment, no es produeixen massa canvis de pla per resoldre un cas. El grau d'inconstància és, doncs, petit inicialment. Després d'adquirir més casos, quan hi ha en memòria entre 25 i 40 casos, el grau d'inconstància és força elevat: BOLERO canvia fins i tot 7 vegades de pla per resoldre un problema. El nombre de plans en memòria ha augmentat, per la qual cosa hi ha més possibilitats que BOLERO, per resoldre un cas del qual no coneix la solució (i.e. no hi ha cap cas idèntic en memòria), faci combinacions amb més d'un pla per resoldre un problema. A mesura que BOLERO aprèn més sobre plans (a partir de 40 casos en memòria), la inconstància disminueix fins a arribar a 2,6 canvis de pla per cas. El nombre de plans en memòria ha augmentat prou com perquè hi hagi més possibilitats de trobar un cas semblant en memòria i, amb pocs canvis, elaborar el pla adequat. Donades les característiques del domini, no esperem arribar a una quota d'1. Com ja ha estat dit en la secció anterior, una quota d'1 significa que la informació inicial disponible sobre un problema és totalment completa i precisa, cosa que no succeeix en el domini mèdic en el qual hem aplicat BOLERO. Si la informació fos completa i precisa, no tindria sentit el disseny d'un planificador reactiu. Per tant, un grau d'inconstància del 2,6 és molt bo. No obstant això, seria interessant intentar mesurar el grau d'inconstància del mestre i comparar-lo al de BOLERO.

Resumint els resultats del grau d'assoliment i el grau d'enfocament, podem observar que ja amb pocs casos apresos BOLERO és capaç de resoldre problemes amb un grau d'assoliment del 97% (aprox.) i un grau d'enfocament del 80% (aprox.). El grau d'ordenació en la predicció, donat pels graus d'enriquiment i de desencert, no és certament molt important (12,90% i 12,15% respectivament). És a dir, recordem que el grau d'ordenació ens dona informació del nombre d'accions que han estat ordenades en el pla en posicions diferents a les posicions que ocupen en el pla de l'expert mestre. En particular, el grau d'enriquiment mesura les accions

del pla generat per BOLERO que tenen un ordre millor respecte a les posicions que ocupen en el pla del mestre. I el grau de desencert mesura les accions que han estat ordenades per BOLERO amb un ordre pitjor (vegeu secció 4.5.2). Els resultats obtinguts, doncs, relaten que un 12,90% de les accions d'un pla generat per BOLERO han estat posicionades en millor ordre, mentre que un 12,15% en un ordre inadequat. Si suposem que la mitjana de la llargària dels plans és de 13 accions aproximadament (vegeu secció 5.3.4), un 12,90% no arriba ni a 2 accions per pla. Respecte als resultats del grau de desencert, un 12,15% de les accions (i.e. una mitjana d'1,5 accions per pla) estan mal ordenades. Cal observar, però, la seva evolució en el gràfic següent:

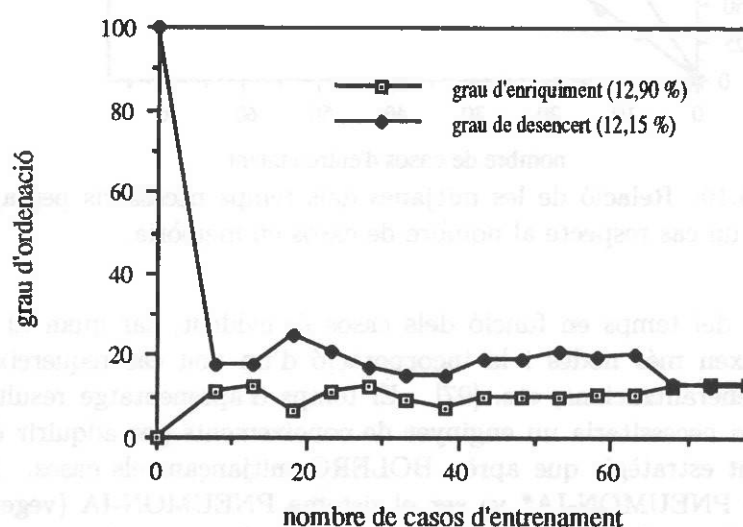


Figura 5.9: Evolució del grau d'enriquiment i del grau de desencert en la predicció.

Quan s'afegeixen més casos en memòria, el grau d'enriquiment augmenta fins al punt d'agafar al grau de desencert i fins i tot superar-lo. Això vol dir que BOLERO tendeix a resoldre els problemes construint plans on l'ordre de les accions és poc diferent a l'ordre de les accions dels plans construïts pel mestre, i en qualsevol cas tendeix a millorar-lo.

5.3.3 Temps d'execució

El temps que consumeix BOLERO en l'aprenentatge d'un cas augmenta a mesura que s'incorporen nous casos en memòria (i en conseqüència el nombre de nodes, com s'ha il·lustrat en el gràfic de la figura 5.1), com es mostra el gràfic de la figura 5.10:

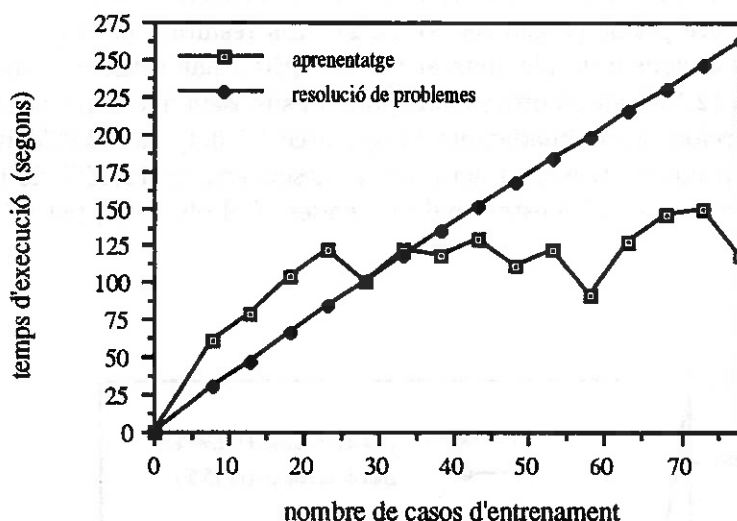


Figura 5.10: Relació de les mitjanes dels temps necessaris per aprendre i resoldre un cas respecte al nombre de casos en memòria.

Aquest augment del temps en funció dels casos és evident, car quan hi ha més casos en memòria, existeixen més nodes i la incorporació d'un nou cas requereix més nombre de comparacions, generalitzacions, etc. [97]. El temps d'aprenentatge resulta, però, molt inferior al temps que necessitaria un enginyer de coneixements per adquirir del seu expert humà el coneixement estratègic que aprèn BOLERO mitjançant els casos. En particular, la versió original de PNEUMON-IA* va ser el sistema PNEUMON-IA (vegeu apèndix C). PNEUMON-IA incloïa en el seu coneixement un conjunt de metaregles que eren utilitzades per a planificar la diagnosi del pacient. L'expert va trigar aproximadament l'equivalent a dues setmanes (comptant 8 hores diàries de treball) per desenvolupar les metaregles. BOLERO, després d'haver après 77 casos, triga aproximadament 2' per aprendre el cas 78. El guany és de 42 ordres de magnitud aproximadament⁵. D'altra banda, cal considerar el temps per reunir els casos que BOLERO utilitza per aprendre. Tanmateix, en PNEUMON-IA també calia reunir aquests casos, car les metaregles codificades manualment es van verificar i incrementalment modificar (d'aquí el temps de 2 setmanes) fins a assolir una conducta estratègica acceptable. La mesura de comparació és, doncs, plenament vàlida i il·lustra el potencial benecifi d'usar tècniques d'aprenentatge en l'adquisició d'estratègies.

Respecte al temps de resolució de problemes⁶, l'augment és lineal, amb un pendent aproximat de 3.5. Cal distingir una petita diferència establerta entre el temps necessari per resoldre un cas que ja està en memòria (identificació) i el temps requerit per resoldre un nou problema (predicció). En el segon cas hi ha un empitjorament molt lleuger, com es mostra en la taula 5.1:

⁵Suposant el pitjor cas, és a dir 2' per aprendre un cas quan hi ha 78 casos en memòria, BOLERO triga 2,6 hores. L'expert triga 8 hores diàries durant 2 setmanes (comptant 7 dies per setmana).

⁶Cal distingir el temps de construcció d'un pla en BOLERO del temps de resoldre un problema en BOLERO-SBC, i.e. construir n plans.

casos	identificació	predicció
10.36	30.78	34.97
16.67	47.31	53.00
23.08	66.98	73.25
29.49	84.12	91.64
35.90	101.30	108.36
42.31	118.10	125.93
48.72	135.18	142.67
51.13	151.63	157.26
61.54	167.21	172.78
67.95	182.78	188.55
74.36	198.10	203.16
80.77	215.49	220.92
87.18	229.26	234.13
93.59	246.54	250.85
100	262.43	265.77

Taula 5.1. Temps necessari (mitjana) per resoldre un cas en la identificació i en la predicció. En la columna "casos" s'indica el percentatge de casos (sobre el total de 79) que BOLERO té en memòria quan es realitzen les medicions. La columna "identificació" conté els temps de resolució d'un cas que ja està en memòria. La columna "predicció" recull les dades del temps necessari perquè BOLERO resolgui un nou problema.

Hem comparat el temps de resolució de problemes de BOLERO amb el de PNEUMON-IA, la versió original de PNEUMON-IA* esmentada amb anterioritat (vegeu apèndix C). PNEUMON-IA triga de l'ordre de 5.07 segons de mitjana per solucionar un cas. La diferència entre BOLERO i PNEUMON-IA és, doncs, substancial. Malgrat que la diferència és perceptible en un entorn *batch*, no és evident en un entorn *on-line*. És a dir, els resultats que es mostren han estat calculats de forma *batch* i les seqüències d'avaluació que s'han dut a terme per tal d'avaluar experimentalment BOLERO han estat realitzades sense la presència d'un usuari. Però, de fet, el sistema resultant BOLERO-SBC interacciona amb un usuari. Cada fet que *SBC^B* (i.e. PNEUMON-IA*) requereix per dur a terme les seves inferències és qüestionat a l'usuari. Afortunadament, el temps de resposta de BOLERO quan l'usuari està interaccionant amb el sistema BOLERO-SBC s'ha estimat, quan hi ha 78 casos en memòria, en menys de 5 segons⁷, que considerem un temps de resposta raonable. És a dir, l'usuari és incapaç de distingir si està treballant amb els sistema PNEUMON-IA o amb BOLERO-SBC quant al temps de resposta del sistema.

El que seria interessant estudiar és la relació entre el nombre de casos d'aprenentatge i el cost (temps) que necessita BOLERO-SBC per resoldre problemes nous [77]. Quan BOLERO ha processat pocs casos (8 casos), ja té un grau d'assoliment i un grau d'enfocament prou

⁷Si suposem que per a cada problema es presenta de mitjana un nombre de 54,20 preguntes a l'usuari, i quan hi ha 78 casos en memòria BOLERO triga al voltant de 262,43 segons a resoldre el cas complet, podem pensar que el temps de resposta del sistema entre pregunta i pregunta és aproximadament $262,43/54,20=4,85$. Aquest és el temps que requereix BOLERO per construir un pla, i.e. el temps de resposta a un canvi en la informació.

bons (97,92% i 85,22% respectivament). Amb vuit casos processats, el temps de resposta del sistema és inferior a un minut. Quan hi ha 48 casos en memòria, el grau d'assoliment ha arribat ja al 98,30% i el grau d'enfocament al 87,92%, mentre que ja triga 172,78 segons (quasi 3 minuts) a resoldre un problema. Per tant, fóra bo desenvolupar un mecanisme de decisió que avalués la conveniència d'incorporar plans nous en memòria en lloc d'incorporar-los tots com ara. Com es veurà en la secció 5.3.4, molts plans només presenten diferències entre l'ordre de les accions i realment podrien considerar-se equivalents. En qualsevol cas, aquest és un problema d'optimització del sistema que esperem resoldre en un futur.

5.3.4 Mesures relatives a l'arbre de plans

A més a més de la quantitat de nodes i accions en memòria (de les quals s'acaben de proporcionar els gràfics en la secció anterior), en cada una de les seqüències d'avaluació s'ha mesurat la profunditat i l'amplada de l'arbre que constitueix la memòria de plans de BOLERO. Els resultats obtinguts són:

- Profunditat mitjana = 12,9367 nodes (variança = 4,6179)
- Amplitud mitjana = 1,13 fill per node (variança 0,5263)

Els plans finals són doncs, en general, llargs (50% de totes les accions possibles en el domini de pneumònies). Remarquem que existeixen poques ramificacions, la major part concentrades al voltant de l'arrel de l'arbre. Quan més ens acostem als nodes terminals, els plans semblen estar ja fortament diferenciats (i.e. hi ha més informació i, per tant, menys indecisió en els plans). Però si s'observa amb detall la jerarquia, en realitat els plans són força semblants, i la diferència entre ells radica en la presència o l'ordre d'alguna de les accions. En particular, en la jerarquia s'observen el que anomenem *camins rectes*. Un camí recte és un camí parcial de nodes no terminals que només tenen un descendent (també no terminal). Per exemple, en la figura 5.11 un camí recte el formen els nodes $[N^{14}, N^{15}, \dots, N^{25}]$. Un altre camí recte és $[N^{39}, N^{40}, N^{41}, N^{42}]$.

Els camins rectes ens indueixen a pensar en una compactació o *chunking* dels nodes. La compactació consisteix en la substitució d'un camí recte $[N^{k_1}, N^{k_2}, \dots, N^{k_m}]$ per un únic node N . El pla parcial p associat al node N es diferencia del pla parcial p^{k_0} del seu pare n_0 no en una acció sino en un *subplà*. És a dir, si $p^{k_1} = p^{k_0}; a_{k_1}$, $p^{k_2} = p^{k_1}; a_{k_2}$, $\dots$, i $p^{k_m} = p^{k_{m-1}}; a_{k_m}$, aleshores $p = p^{k_0}; \rho$ on ρ és el subplà $\rho = [a_{k_1}, a_{k_2}, \dots, a_{k_m}]$.

Compactant els camins rectes obtenim una jerarquia més senzilla, amb menys quantitat de nodes (figura 5.12). Una jerarquia més simple implica una reducció del nombre d'indexacions i, en conseqüència, un millor temps de resposta de BOLERO en la resolució de problemes. A més a més, observem que existeixen subplans distribuïts en diferents punts de la jerarquia, la qual cosa fa pensar que l'expert treballa amb conjunts d'accions (subplans) en el procediment que segueix per tal de diagnosticar un pacient més que amb accions individuals. Per exemple, $[N^{39}, N^{40}, N^{41}, N^{42}]$ constitueix un subplà idèntic al subplà de $[N^{26}, N^{32}, N^{33}]$.

La compactació, però, no només presenta avantatges. Una rutina de compactació comportaria un mètode d'aprenentatge més complex. La raó principal resideix en el mecanisme de *descompactació* que necessàriament s'hauria de desenvolupar. La descompactació és una tasca difícil que resta com a problema obert. Tanmateix, si es volgués construir un sistema expert (que no fes més aprenentatge, i.e. es considerés un sistema acabat per ser utilitzat per usuaris no experts), podria fer-se la compactació únicament. Això podria millorar molt l'eficiència de BOLERO en la resolució de problemes.

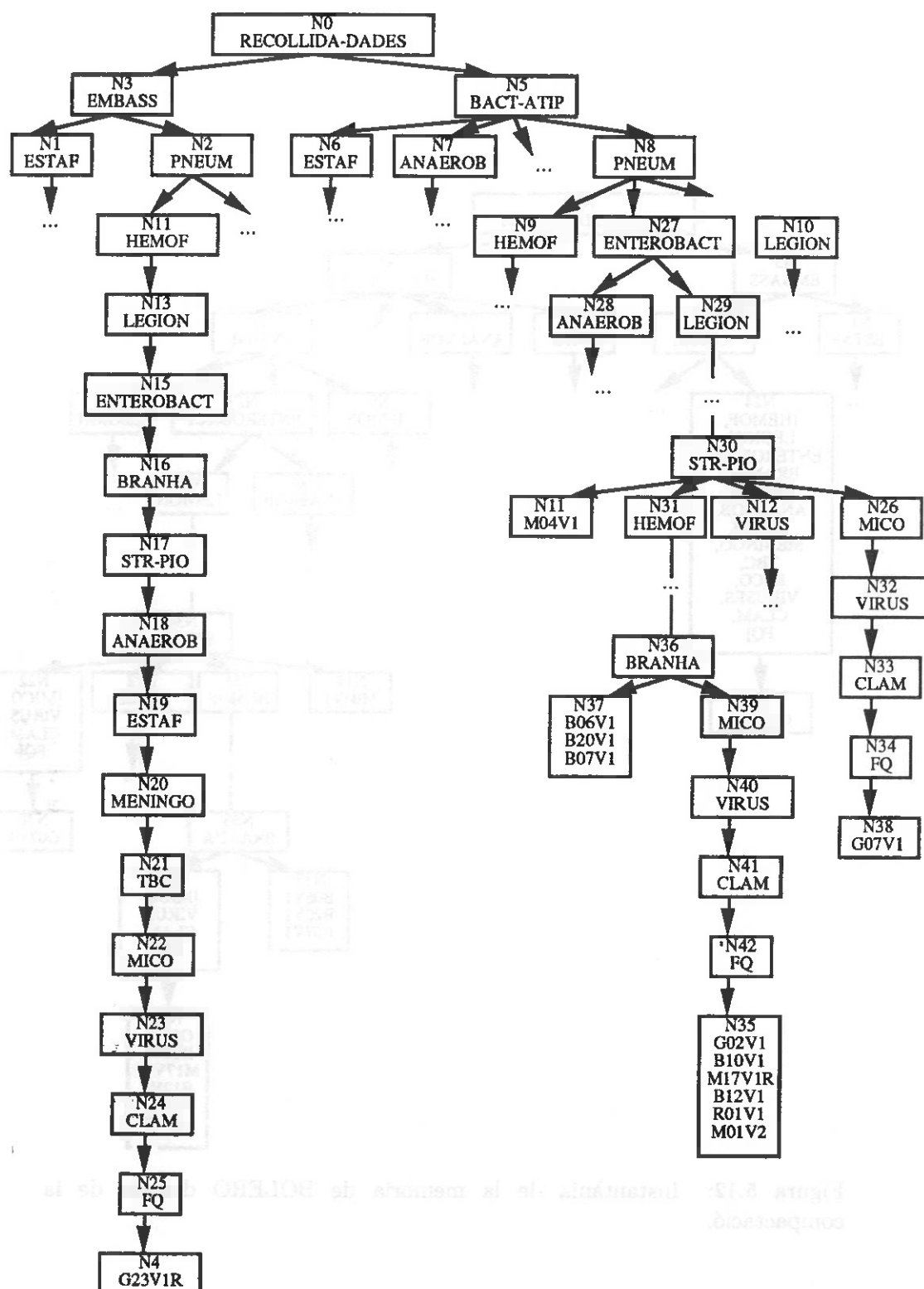


Figura 5.11: Instantània (*snapshot*) de la memòria de BOLERO quan ha après per observació 79 casos. Per qüestió de claredat, en cada node només s'ha dibuixat l'identificador del node i la darrera acció del pla que diferencia el node del seu pare.

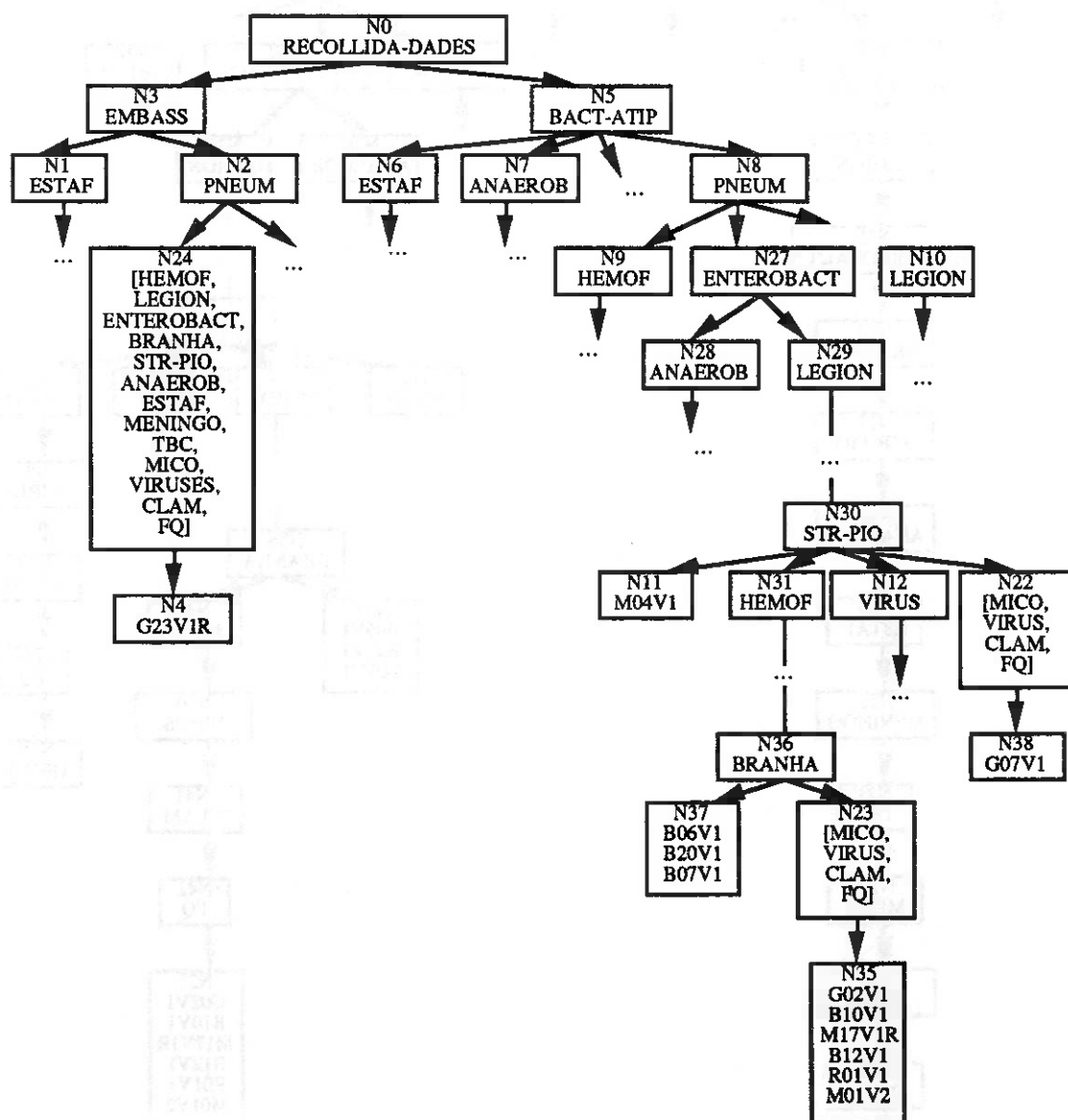


Figura 5.12: Instantània de la memòria de BOLERO després de la compactació.

5.4 Variacions

En aquesta secció presentem un conjunt de variacions fetes sobre la configuració bàsica per tal de mesurar l'impacte de les característiques del sistema en els seus resultats. S'han modificat diferents característiques del sistema, creant fins a 7 configuracions diferents (sense comptar la bàsica). A continuació es mostren els resultats aconseguits seguint el aquest patró: descripció de la configuració, hipòtesi que es pretén verificar i resultats.

5.4.1 Avaluació de plans multiexpert

Configuració 1. Per avaluar els efectes d'una avaluació basada en diferents experts hem substituït el mètode d'avaluació multiexpert per l'avaluació basada exclusivament per l'expert mestre. Un patró d'avaluació en la configuració 1 és $PA(C^i) = \langle PA_p(C^i), \emptyset \rangle$, mentre que en la configuració bàsica és $PA(C^i) = \langle PA_p(C^i), PA_d(C^i) \rangle$.

Hipòtesi. L'avaluació multiexpert afavoreix que els plans siguin més llargs que els que elabora l'expert. El fet que accions que l'expert mestre no considera necessàries s'acceptin constitueix una estratègia més conservadora (va sobre segur), però va en contra del criteri de parsimònia. Acceptar més accions per pla amb una avaluació multiexpert condueix a augmentar el grau d'enfocament dels plans, ja que l'avaluació multiexpert permet acceptar un nombre més ampli d'accions per pla. En la configuració 1, quan només s'accepta l'opinió de l'expert mestre en l'avaluació de plans, el nombre d'accions acceptades en un pla disminueix. Conseqüentment, és d'esperar que el grau d'enfocament en la configuració ha de ser pitjor al valor obtingut en la configuració bàsica.

Resultats. BOLERO continua essent correcte i precís en la identificació. El grau d'assoliment en la predicció també es manté en el 98,30%. Observem, però, una davallada del grau d'enfocament en la predicció com ho mostra la taula 5.2:

casos	multiexpert	expert mestre
10.36	85.22	82.86
16.67	82.23	80.01
23.08	80.24	78.02
29.49	80.30	78.20
35.90	79.88	78.21
42.31	80.93	79.26
48.72	82.43	80.76
51.13	87.01	85.35
61.54	97.92	86.25
67.95	89.17	87.5
74.36	90.00	88.33
80.77	90.42	88.75
87.18	90.42	88.75
93.59	91.25	89.21
100	91.25	89.58

Taula 5.2. Grau d'enfocament en la predicció. En la columna *casos* s'indica el percentatge de casos (sobre el total de 79) que BOLERO té en memòria quan es realitzen els mesuraments. En la columna *multiexpert*, es donen els graus d'enfocament obtinguts si en l'avaluació es té en compte l'opinió d'altres experts a més de la del mestre (configuració bàsica). I en la tercera columna, *expert mestre*, apareixen els resultats aconseguits quan en l'avaluació només es té en compte l'opinió del mestre (configuració 1).

Per tant la nostra hipòtesi es confirma: els plans que BOLERO construeix són més llargs que els que construeix l'expert mestre. Per exemple, mentre que BOLERO ha resolt el cas "b05v1" mitjançant el pla:

[recollida-dades, bact-atip, pneumococ, micoplasma,
virus, clamídia, legionel·la, enterobacteriàcies,
s-piògen, hemòfil, branhamel·la, febre-q, tuberculosi,
anaerobi, estafilococ, meningococ]

el mestre ha construït el pla:

[recollida-dades, bact-atip, pneumococ, legionel·la,
enterobacteriàcies, anaerobi, hemòfil, estafilococ,
s-piògen, meningococ, branhamel·la, tuberculosi]

Ambdós plans són correctes: les accions micoplasma, virus, clamídia, i febre-q no considerades pel mestre han estat considerades per un expert i, per tant, són correctes. Les noves accions no considerades pel mestre són acceptades com a correctes si hi ha qualche expert que les hi ha considerades (accions que a partir d'ara anomenem accions *admeses*). Per aquesta raó, amb una avaluació multiexpert el grau d'enfocament és més elevat que amb una avaluació basada exclusivament en l'expert.

L'avaluació multiexpert presenta una altra conseqüència, que és la impossibilitat d'establir un ordre total per accions admeses. Els criteri d'ordre de les accions descrit en el capítol 4 (secció 4.5.2) no és aplicable a les accions admeses per dos motius principals. En primer lloc, els resultats de les accions admeses són fets difusos amb valors molt poc significatius (*desconegut*, *gens-p*). Això és degut al fet que la informació disponible sobre un cas es basa

en la recollida d'informació que ha fet el mestre, i si aquest no ha considerat una acció a , és natural pensar que no hi ha fets com per a dur a terme a . Per exemple, en el cas "b05v1", l'expert mestre no ha considerat l'acció *micoplasma*. Però sí que l'han considerat altres experts, diagnosticant el fet (*micoplasma, possible*). BOLERO també considera l'acció *micoplasma* quan resol el cas "b05v1", encara que no obté més que el fet (*micoplasma, desconegut*). Ni els coneixements de domini disponibles, ni la informació continguda en el cas "b05v1" permeten deduir cap valor significatiu per a l'objectiu *micoplasma*. En segon lloc, el valor poc significatiu d'una acció admesa no condiciona, seguint els criteris d'ordenació (menys evidència en els resultats d'una acció, execució més tard), que hagi de realitzar-se en darrer lloc. BOLERO ha executat *micoplasma* abans que *legionel·la*, i ha obtingut resultats de les accions són (*micoplasma, desconegut*), i (*legionel·la, gens-possible*) respectivament. Aquesta informació no ens pot dur a pensar que l'ordre és inadequat, és a dir, suposar que com *gens-possible* > *desconegut* *legionel·la* ha de ser executada abans que *micoplasma*. La solució de domini consensuada $PA_d(b05v1)$ ens indica que el resultat de *micoplasma* pot ser (*micoplasma, possible*), i per tant l'ordre seguit per BOLERO (i.e. primer *micoplasma* i després *legionel·la*) podria ésser correcte. Ens trobem, doncs, amb la incapacitat de suggerir un ordre per a les accions admeses, i hem convingut que l'ordre amb què BOLERO ha planificat les accions admeses és el correcte. Així, en l'exemple del cas "b15v1", l'ordre de les accions admeses *micoplasma*, *virus*, *clamídia*, i *febre-q* seguit per BOLERO és un ordre adequat.

5.4.2 Aprenentatge per observació versus aprenentatge per experiència

Configuració 2. Per avaluar l'impacte en els resultats de l'aprenentatge per experiència s'ha passat a substituir l'aprenentatge per observació de la configuració bàsica per l'aprenentatge per experiència. En particular, s'han utilitzat els mètodes d'aprenentatge pel reeiximent i pel fracàs del pla (i.e. els mètodes descrits a les seccions 4.6.1 i 4.6.2 respectivament). L'aprenentatge de la importància estratègica dels índexs ha estat avaluat en una altra configuració i els resultats es mostren en la secció 5.4.4. En la configuració 2, doncs, BOLERO sempre intenta resoldre un cas, i només si no ho soluciona bé acudeix al mestre perquè li proporcioni el cas correcte. Per tant, el biaix de la configuració 2 és preferir la solució pròpia a la solució del mestre.

Hipòtesi. De la configuració 1 hem observat que una avaluació multiexpert afavoreix l'execució de plans llargs. Aprendre de l'experiència pròpia significa aprendre plans vàlids d'acord amb l'avaluació multiexpert. Aleshores, la llargària mitjana dels plans que hi hagi en memòria en la configuració augmentarà respecte de la llargària dels plans en la configuració bàsica. En la identificació no s'espera cap variació, car si un pla p generat per BOLERO per resoldre un cas C és correcte i precís, i s'incorpora en la memòria, és de suposar que qualsevol cas idèntic a C es resoldrà amb p d'una manera correcta i precisa (altrament estaríem suposant que els criteris d'avaluació varien). Ara bé, en la predicció és possible que es generin accions sobrerres per resoldre plans. És a dir, per poder finalitzar l'execució d'un cas és necessari que almenys hi hagi un pla d'un node final completat. Si els plans en memòria són més llargs, els plans generats a partir d'aquest seran consegüentment més llargs. Aleshores és raonable pensar que en els plans generats per a un cas no vist amb anterioritat hi apareiguin accions sobrerres. En conseqüència, el grau d'enfocament en la predicció ha de disminuir.

Resultats. La identificació en la configuració 2 no varia respecte a la configuració bàsica: el grau d'assoliment és del 100%, el grau d'enfocament del 100%, el grau d'enriquiment del 0% i el grau de desencert del 0%. BOLERO continua essent correcte i precís en la identificació. El grau d'assoliment en la predicció es manté en el 98,30%. El grau d'enfocament en la predicció varia en la configuració 2 respecte de la configuració bàsica, tal i com es mostra en la taula 5.3.

casos	observació	experiència
10.36	85.22	82.86
16.67	82.23	80.01
23.08	80.24	78.02
29.49	80.30	79.86
35.90	79.88	78.63
42.31	80.93	79.68
48.72	82.43	82.43
51.13	87.01	86.24
61.54	97.92	87.50
67.95	89.17	87.92
74.36	90.00	88.75
80.77	90.42	89.15
87.18	90.42	88.33
93.59	91.25	89.58
100	91.25	90.00

Taula 5.3. Grau d'enfocament en la predicció. En la columna *casos* s'indica el percentatge de casos (sobre el total de 79) que BOLERO té en memòria quan es realitzen els mesuraments. En la segona columna, *observació*, es donen els graus d'enfocament obtinguts per la configuració bàsica. I en la tercera columna, *experiència*, apareixen els resultats aconseguits en la configuració 2.

Els resultats confirmen la nostra hipòtesi, puix que mentre que en la identificació els resultats es mantenen inalterats, en la predicció s'observa una pèrdua lleugera del grau d'enfocament respecte de la configuració bàsica. L'evolució del grau d'enfocament en la predicció és, però, correcta en el sentit que el grau d'enfocament augmenta amb el nombre de casos d'entrenament. La devallada del grau d'enfocament en la predicció és una conseqüència de l'avaluació multiexpert dels plans, però a més a més el problema de les classes alternatives hi contribueix (veure secció 5.1.2). Per una banda, l'avaluació multiexpert afavoreix la creació de plans finals més llargs, com ja ha estat esmentat amb anterioritat, però també diferents als que ha creat el mestre, tot generant una jerarquia de plans més fina que en l'aprenentatge per observació. És a dir, si BOLERO soluciona els problemes correctament amb plans diferents al del mestre, aleshores BOLERO aprèn més plans que els que aprendria per l'aprenentatge per observació. Si suposem que cada node terminal determina una classe de casos d'acord amb el seu pla final (com ja s'ha suposat altres vegades), BOLERO aprèn classes noves diferents a les que aprèn amb el mestre. Per tant, es creen classes noves. D'aquesta manera observem com el nombre de nodes terminals (i.e. classes) augmenta en l'aprenentatge per experiència respecte a l'aprenentatge per observació:

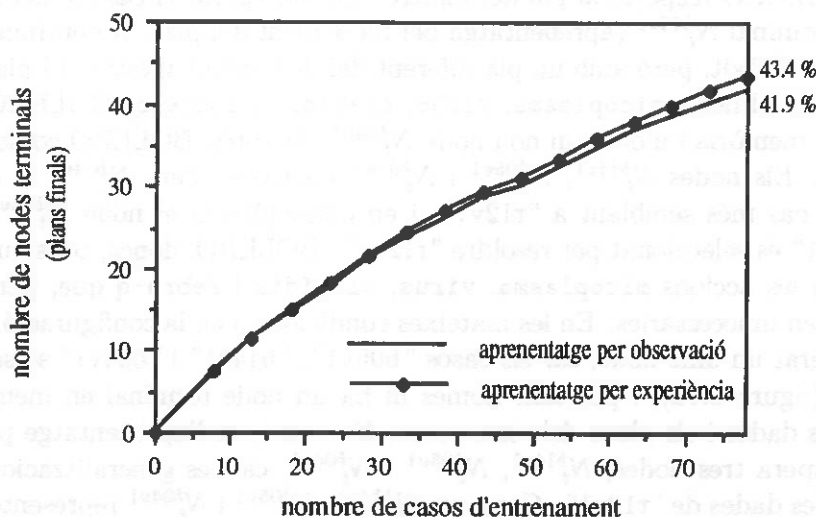


Figura 5.13: Nombre de nodes terminals (i.e. plans finals diferents o classes) en memòria, quan BOLERO aprèn per observació (configuració bàsica) i quan aprèn per experiència (configuració 2), respecte al nombre de casos d'entrenament.

La construcció de plans nous no només afavoreix la creació de classes noves, sinó que també es produeixen canvis de classe. És a dir, suposem que en l'aprenentatge per observació s'obté una classificació dels casos C , i en l'aprenentatge per experiència una classificació C' . Suposem que un cas C^1 es classifica en la classe c_1 de C d'acord amb el seu pla final, on també estan classificats els casos C^2 i C^3 ; i C^4 està classificat en la classe c_2 de C . Suposem també que BOLERO resol C^1 amb un pla final diferent als plans de C^2 i C^3 , però igual al pla final de C^4 . Aleshores en C' , el cas C^1 i C^4 pertanyen a una classe c'_1 , mentre que els casos C^2 i C^3 pertanyen a una altra classe c'_2 . Diem que el cas C^1 ha canviat de classe.

Per altra banda, la creació de classes noves i els canvis de classe fa palès el problema de les classes alternatives en la generació de plans (vegeu secció 5.1.2). La classificació dels casos canvia, però la informació continguda en els casos no canvia, raó per la qual la discriminació entre classes alternatives es fa difícil. Atès que una classe correspon a un node terminal, el problema de les classes alternatives ens indica que no hi ha discriminació entre nodes terminals "alternatius". Això causa que BOLERO, en la configuració 2, recuperi més nodes per generar un pla que la quantitat de nodes recuperats en la configuració bàsica. Però alguns dels nodes són recuperats erròniament, com s'observa en el exemple següent.

Suposem els casos "b05v1", "b14v1", "r12v1" i "b04v1" introduïts a la secció 5.1.2, que presenten el problema de les classes alternatives. Suposem també que en la seqüència d'avaluació els casos "b14v1", "b05v1" i "b04v1" són d'entrenament, i el cas "r12v1" és un cas de test. En la configuració 2, el cas "b14v1" és solucionat incorrectament (és el primer cas en la seqüència i BOLERO no disposa encara de coneixements de planificació de pneumònies) i per tant se sol·licita al mestre el cas correcte (aprenentatge pel fracàs del pla). El cas "b14v1" és incorporat en memòria, i el seu pla final està representat en el node terminal N_t^{b14v1} . Després d'entrenar-se amb un cert nombre de casos, BOLERO resol el cas "b05v1".

En aquest cas BOLERO encerta i resol correctament el cas, però amb un total de quatre accions admeses, car hi ha qualche expert que ha considerat quatre accions que hi ha de més en el pla de BOLERO respecte al pla del mestre. En incorporar el cas en memòria es genera un nou node terminal N_t^{b05v1} (aprenentatge pel reeiximent del pla). A continuació BOLERO resol "b04v1" amb èxit, però amb un pla diferent del de l'expert mestre. El pla de BOLERO conté les accions admeses micoplasma, virus, clamídia i febre-q. BOLERO incorpora el cas "b04v1" en memòria i n'obté un nou node N_t^{b04v1} . Després, BOLERO comença a resoldre el cas "r12v1". Els nodes N_t^{b14v1} , N_t^{b05v1} i N_t^{b04v1} s'activen. Però N_t^{b04v1} és el més activat puix que és el cas més semblant a "r12v1", i en conseqüència el node N_t^{b04v1} on és el pla final del "b04v1" és seleccionat per resoldre "r12v1". BOLERO, doncs, construeix un pla per a "r12v1" amb les accions micoplasma, virus, clamídia i febre-q que, per al cas actual "r12v1", resulten innecessàries. En les mateixes condicions, i en la configuració bàsica, només s'hauria recuperat un únic node, car els casos "b05v1", "b14v1" i "b04v1" se solucionen amb el mateix pla (figura 5.1a), i per tant només hi ha un node terminal en memòria on estan representats les dades i els plans dels tres casos. En canvi, en l'aprenentatge per experiència BOLERO recupera tres nodes, N_t^{b14v1} , N_t^{b05v1} i N_t^{b04v1} , car les generalitzacions de tots ells són similars a les dades de "r12v1". Com que N_t^{b14v1} , N_t^{b05v1} i N_t^{b04v1} representen 3 classes de plans que presenten el problema de les classes alternatives, BOLERO és incapaç de discriminar entre elles, i finalment escull N_t^{b04v1} , atès que "b04v1" és el cas més semblant a "r12v1".

Cal remarcar que, a més a més, en l'aprenentatge per l'experiència l'ordre en què es proporcionen els casos a BOLERO és important. Si abans d'intentar resoldre el cas "r12v1" s'hagués introduït en memòria el cas "b02v1", que és pràcticament idèntic a "r12v1", BOLERO hauria recuperat el node N_t^{b02v1} , resolent amb un pla correcte i precís el cas "r12v1"⁸.

El temps d'execució (mesurat en segons de CPU) i, doncs, la resposta del sistema, es veuen fortament afectats si substituïm l'aprenentatge per observació per l'aprenentatge per experiència. Observem-ho en el gràfic de la figura 5.14:

⁸Cal recordar també que els resultats que es mostren en tot aquest capítol és la mitjana de 10 seqüències de prova diferents. Per tant, suposem que els resultats presentats són independents de la seqüència de prova.

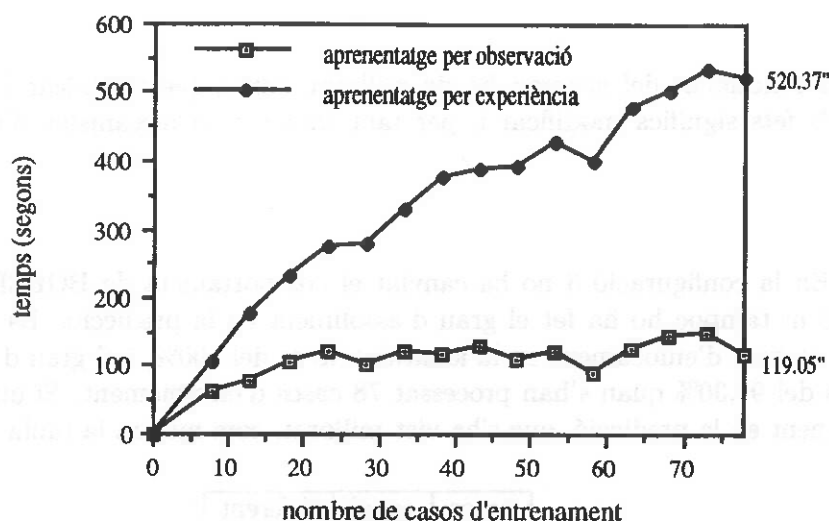


Figura 5.14: Temps (en segons, mitjana) necessari per aprendre un cas mitjançant l'aprenentatge per observació (configuració bàsica) i l'aprenentatge per experiència (configuració 2).

Aquest augment del temps d'execució és comprensible si observem que en el mètode d'aprenentatge per experiència s'inclou la major part del mètode d'aprenentatge per observació (inclusió d'un cas en memòria).

Abans de finalitzar aquesta secció cal observar que s'ha fet molt d'èmfasi en la justificació d'una davallada de 1,25 punts en el grau d'enfocament en la predicció. Però la corba d'aprenentatge (i.e. l'evolució del grau d'enfocament en la predicció) ens mostra que el grau d'enfocament tendeix a millorar en funció del nombre de casos en memòria, com ocorre en la configuració bàsica. A més a més, la resta de mesures es mantenen: el grau d'assoliment en la identificació és del 100% i en la predicció del 98,30%; el grau d'enfocament en la identificació és del 100%.

5.4.3 Importància estratègica dels fets

Configuració 3. La següent alteració realitzada en BOLERO ha estat la d'assignar una importància estratègica diferent a cada fet. L'assignació s'ha realitzat mitjançant l'organització dels fets en classes dins de la jerarquia proporcionada pels coneixements de domini. Així, per exemple, als fets que pertanyen a la classe *semiologia* els hi ha estat adjudicada una importància estratègica de 0.2, mentre que als fets de la classe *dades-de-laboratori* se'ls ha adjudicat una importància estratègica de 0.9. La importància de les classes ha estat establerta per l'expert mestre (el nostre expert en pneumònies) d'acord amb la confiança que diposita en els fets de cada classe a l'hora d'elaborar el seu procediment de diagnòstic. Així, les dades semiològiques són molt subjectives, poden ser interpretades de moltes maneres. Per això els fets de la classe *semiologia* tenen una importància estratègica petita. Les dades de laboratori són, al contrari, molt més objectives i fiables a l'hora de formular un diagnòstic. Per tant, el mestre ha assignat un valor estratègic important als fets

d'aquesta classe.

Hipòtesi. El rendiment del sistema ha de millorar donat que modificar la importància estratègica dels fets significa modificar i, per tant millorar, el mecanisme d'indexació dels casos.

Resultats. En la configuració 3 no ha canviat el comportament de BOLERO respecte a la identificació ni tampoc ho ha fet el grau d'assoliment en la predicció. És a dir, el grau d'assoliment i el grau d'enfocament en la identificació és del 100%, i el grau d'assoliment en la predicció és del 98,30% quan s'han processat 78 casos d'entrenament. Sí que ha variat el grau d'enfocament en la predicció, que s'ha vist millorat, com mostra la taula 5.4.

casos	igual	diferent
10.36	85.22	85.22
16.67	82.23	81.73
23.08	80.24	81.85
29.49	80.30	83.99
35.90	79.88	81.71
42.31	80.93	83.43
48.72	82.43	87.43
51.13	87.01	88.74
61.54	97.92	90.83
67.95	89.17	92.08
74.36	90.00	91.67
80.77	90.42	91.67
87.18	90.42	93.33
93.59	91.25	94.17
100	91.25	94.17

Taula 5.4. Configuració 3: variació del grau d'enfocament en la predicció. La columna *igual* conté les dades referents als resultats obtinguts quan tots els fets tenen la mateixa importància estratègica (configuració bàsica); la columna *diferent* correspon als resultats aconseguits quan la importància estratègica és diferent per a cada classe de fets (configuració 3).

Això significa que la importància estratègica dels fets juga un paper molt important en la indexació dels casos (fets), com era d'imaginar. Fets amb una importància estratègica alta són fets molt rellevants; fets amb una importància estratègica baixa són fets irrelevants que podrien suprimir-se. L'eliminació de fets irrelevants comporta un nombre menor d'indexacions i, en conseqüència, una disminució del temps de resposta de BOLERO, com es veurà amb l'experiment detallat a la secció 5.4.7.

Assignar una importància estratègica diferent a cada fet no és una tasca fàcil, fins i tot per a experts humans. Per aquesta raó hem intentat en la següent configuració de BOLERO aprendre la importància estratègica dels fets per experiència.

5.4.4 Aprenentatge de la importància estratègica dels fets

Configuració 4. En la tercera alteració (configuració 3) s'ha utilitzat l'algorisme d'aprenentatge de la importància estratègica dels fets per experiència, algorisme que ja ha estat explicat en la secció 4.6.3, en substitució de l'aprenentatge per observació de la configuració bàsica. Inicialment, la importància estratègica dels fets és idèntica (i.e. $w_i = 0.5 \forall f_i \in F$), i aquest valor es va corregint per l'experiència de BOLERO en la construcció de plans.

Hipòtesi. Els valors de la importància estratègica dels fets han de tendir a acostar-se als valors assignats pel mestre en la configuració anterior (configuració 3) i, per tant, a obtenir uns resultats similars.

Resultats. De nou, la identificació de BOLERO no es veu afectada. BOLERO continua essent correcte i precís al 100%. A la taula 5.5 es donen les dades obtingudes pels graus d'assoliment i d'enfocament en la predicció que sí varien.

casos	grau d'assoliment		grau d'enfocament	
	observació	apr.importància	observació	apr.importància
10.36	91.19	97.19	85.22	82.86
16.67	97.75	97.75	82.23	80.01
23.08	98.89	98.89	80.24	78.02
29.49	98.33	98.33	80.30	79.86
35.90	98.33	98.33	79.88	78.21
42.31	98.33	98.33	80.93	80.10
48.72	98.33	98.33	82.43	83.68
51.13	98.33	98.33	87.01	85.76
61.54	98.33	98.33	97.92	86.67
67.95	98.33	98.33	89.17	87.92
74.36	98.33	98.33	90.00	88.75
80.77	98.33	98.33	90.42	89.17
87.18	97.75	98.30	90.42	89.58
93.59	98.30	98.86	91.25	91.67
100	98.30	98.86	91.25	91.25

Taula 5.5. Variació del grau d'assoliment i el grau d'enfocament en la predicció. Les columnes "observació" corresponen a les dades aconseguïdes en la configuració bàsica, i les columnes "apr.importància" contenen les dades de la configuració 4.

Advertim que existeix un petit guany en el grau d'assoliment en la predicció respecte a la configuració bàsica, cosa que no observàvem en la configuració 3, quan la importància estratègica dels fets era assignada per l'expert mestre. Això vol dir que l'aprenentatge de la importància estratègica dels fets per experiència aconseguix una adjudicació més acurada de la importància estratègica dels fets que la proporcionada per l'expert. El grau d'enfocament en la predicció evoluciona diferentment, i no arriba a la mateixa quota que en la configuració anterior (en la configuració 3 s'arriba al 94,17%, tal i com es mostra en la taula 5.4), encara

que arriba a les quotes de la configuració bàsica (91,25%). Si avaluem el guany diferencial entre la configuració 4 i la 3 ($94,17 - 91,25 = 2,92$), veiem que no és un guany molt significatiu. En canvi, el que resulta important és que amb l'aprenentatge de la importància estratègica deslliurem l'expert de la tasca de proporcionar-nos-la. A més a més, cal recordar que disposem de relativament pocs casos d'entrenament i, possiblement, i d'acord amb l'evolució dels resultats en funció del nombre dels casos, amb un conjunt de casos d'entrenament més gran s'aprendria una importància estratègica més adequada a cada fet i, en conseqüència, un grau d'enfocament en la predicció més satisfactori.

Quant al grau d'ordenació en la predicció en l'aprenentatge per observació i en l'aprenentatge de la importància estratègica dels fets, existeixen les diferències següents:

casos	enriquiment		desencert	
	observació	apr.importància	observació	apr.importància
10.36	10.82	9.77	17.94	17.416
16.67	12.37	11.31	17.86	16.81
23.08	6.63	5.58	24.79	23.74
29.49	10.98	10.95	21.11	20.58
35.90	12.16	12.16	16.89	17.89
42.31	9.40	9.95	15.27	15.77
48.72	7.28	7.19	15.13	15.55
51.13	9.51	8.93	18.70	17.16
61.54	9.49	8.90	18.99	17.45
67.95	10.06	9.56	20.99	20.10
74.36	10.20	10.15	19.62	19.47
80.77	10.23	9.70	20.14	20.02
87.18	12.20	15.60	12.77	14.77
93.59	12.87	16.79	12.15	15.30
100	12.90	17.17	12.15	8.52

Taula 5.6. Variació del grau d'enriquiment i el grau de desencert en la predicció. Les columnes "observació" correspon a les dades aconseguïdes en la configuració bàsica, i les columnes "apr.importància" contenen les dades de la configuració 4.

El grau d'enriquiment es veu millorat en una proporció important. Però amb els resultats obtinguts no es pot pronosticar res sobre el grau de desencert, perquè oscilla. Per la qual cosa no es pot valorar d'una manera global l'impacte en el grau d'ordenació de la configuració 4 respecte de la configuració bàsica.

5.4.5 Efectes del paràmetre *mdf*

Configuració 5. L'avaluació experimental de BOLERO no pot deixar de banda l'impacte de la variació del paràmetre *mdf* que condiciona la terminació de la resolució dels problemes. Recordem que el paràmetre *mdf* està definit com segueix (secció 4.4.3):

$$mdf = 1 - e^{-\frac{\lambda}{n}}$$

on *n* és el nombre de nodes terminals en memòria i λ un valor determinat heurísticament. Existeixen dues avaluacions experimentals possibles de l'impacte del paràmetre *mdf*: la

avaluació de la definició de mdf i l'avaluació de mdf en funció de λ . Donat l'amalgama de possibilitats que ofereix l'avaluació de la definició de mdf, ens hem restringit a avaluar l'impacte de la variació de mdf en funció de λ . El paràmetre mdf varia respecte a λ tal i com es mostra en el gràfic de la figura 5.15.

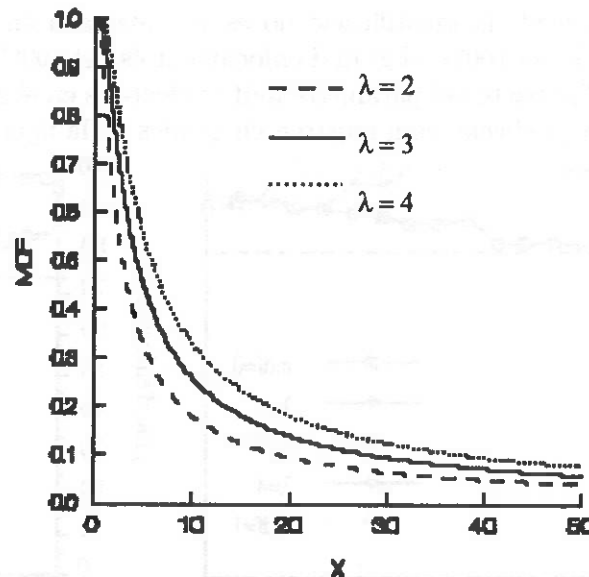


Figura 5.15: Variació del paràmetre mdf respecte a λ i en el nombre de casos en memòria (i.e. nodes terminals).

Com més gran és λ , mdf decreix més lentament. Això significa que, en la resolució de problemes BOLERO exigeix una dissimilitud menys estricta entre les generalitzacions dels nodes per decidir aturar la resolució d'un problema. Hem construït cinc noves configuracions de BOLERO que creiem que són significatives respecte al paràmetre mdf:

Configuració 5.a: El paràmetre mdf val sempre 0 independentment del nombre de nodes terminals.

Configuració 5.b: $\lambda = 2$

Configuració 5.c: $\lambda = 3$

Configuració 5.d: $\lambda = 4$

Configuració 5.e: El paràmetre mdf val sempre 1 independentment del nombre de nodes terminals.

Hipòtesi. Un valor de λ petit significa que mdf decreix molt ràpidament, raó per la qual BOLERO, amb pocs casos en memòria, detecta dissimilituds entre casos abans no ho faria amb un valor de λ gran. Això vol dir que amb un valor petit de λ BOLERO finalitza probablement un cas abans que amb un valor de λ més gran. Conseqüentment, el risc de fracàs del pla (i.e. no executar totes les accions que dicta l'expert mestre a $PA_p(C^i)$) és més gran. Per tant, amb una λ petita el grau d'assoliment ha de disminuir.

L'inversa també és d'esperar. És a dir, amb un valor de λ gran BOLERO triga més a finalitzar la solució dels casos i, per tant, hi ha menys risc que no executi alguna acció de $PA_p(C^i)$. Però, a la vegada, amb un valor de λ gran l'expectació és que BOLERO executi més accions que les desitjades. En conseqüència, el grau d'enfocament ha de disminuir. El valor de λ és, doncs, fonamentalment un biaix que pot primar el grau d'assoliment (i.e. la correcció dels

plans) enfront del grau d'enfocament. El grau d'assoliment augmenta en relació directament proporcional a l'augment de λ . El grau d'enfocament disminueix en relació inversament proporcional a l'augment de λ .

Resultats. Novament, la identificació no es veu afectada en les noves configuracions: el grau d'assoliment és del 100%, el grau d'enfocament és del 100% i el grau de desencert és del 0%. On realment l'impacte del paràmetre mdf té efecte és en el grau d'assoliment i en el grau d'enfocament en la predicció, com mostren els gràfics de la figura 5.16:

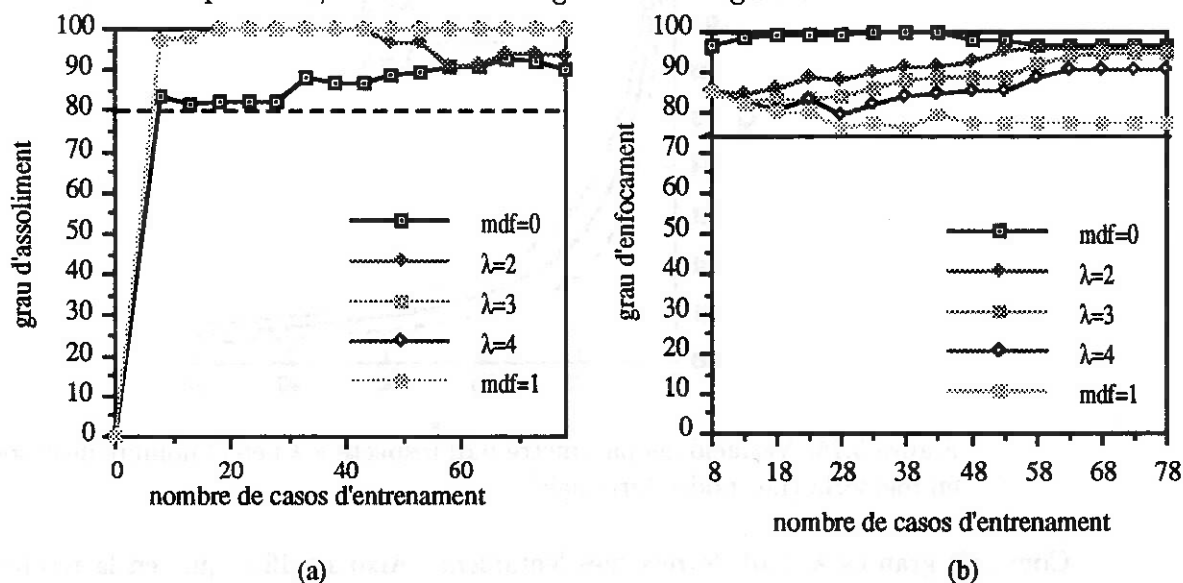


Figura 5.16: Variació del grau d'assoliment (a) i el grau d'enfocament (b) en la predicció en funció del paràmetre mdf (configuracions 5.a, 5.b, 5.c, 5.d, i 5.e).

Quan mdf pren el valor mínim ($mdf = 0$), el grau d'assoliment és el pitjor de tots. Això vol dir que les condicions de terminació que depenen de mdf són massa estrictes i s'atura massa aviat la resolució d'un problema (encara que el grau de semblança entre dos nodes estigui molt a prop, si no són idèntics es consideren suficientment diferenciats com per aturar la resolució d'un problema). A mesura que el valor de mdf augmenta, el grau d'assoliment millora. Fins a arribar al seu valor màxim, $mdf = 1$, en què el grau d'assoliment és del 100% a partir d'un nombre molt petit de casos apresos (18 aprox.). Les nostres expectatives sobre el grau d'assoliment, doncs, es confirmen.

Quant al grau d'enfocament, passa exactament el contrari, tal com s'esperava. Quan mdf pren el valor més petit, $mdf = 0$, el grau d'enfocament és el millor i resulta en algunes ocasions fins i tot l'òptim (quan hi ha entre 33 i 43 casos en memòria aproximadament no fa cap acció de més). El grau d'enfocament es va deteriorant a mesura que mdf augmenta, i arriba a ser el pitjor pel seu valor màxim, $mdf = 1$. Hem comprovat, doncs, l'efecte del biaix de λ en la conducta del sistema.

5.4.6 Efectes del biaix conservador

Configuració 6. Una altre factor molt important a tenir en compte en l'avaluació d'un algorisme d'aprenentatge són els biaixos [154, 155]. En BOLERO hem parlat de la presència

d'un biaix conservador en el mètode d'aprenentatge (veure secció 4.5): totes les accions considerades per l'expert mestre han d'estar incloses en el pla final construït per BOLERO per resoldre un problema d'una manera correcta. Aquest biaix és adequat en el domini mèdic en què ha estat desenvolupat BOLERO. En aquest domini és molt important no eliminar hipòtesis (sobretot si es tracta de malalties perilloses, on el que es vol és demostrar-ne la falsedat), i per tant obtenir resultats com ara (*micoplasma*, *gens-possible*) no es pot interpretar com una acció (en aquest cas *micoplasma*) que ha fracassat. En les mateixes circumstàncies, caldria considerar la hipòtesi de *micoplasma* i no pas eliminar-la⁹.

Per mesurar l'impacte del biaix conservador hem construït una nova configuració (la 6) on no obliguem el sistema a considerar totes les accions que l'expert mestre dicta. El grau d'assoliment en la configuració 6 ve determinat pel nombre d'accions que han estat considerades i que pertanyen al patró d'avaluació $PA_p(C^i)$, més les accions que no ha considerat però que si hagués considerat fracassarien. Per tant, podem pensar que en la configuració 6 tenim el següent patró d'avaluació:

$$PA'_p(C^i) = PA_p(C^i) \setminus \{a | a \in PA_p(C^i) \wedge fracassa(a)\}$$

on *fracassa(a)* és un predicat que és cert quan l'acció *a* fracassa. Una acció *a* fracassa si:

$$\exists f \in results(a), \text{ tal que } valor(f) > lleugerament - possible$$

Una acció *a* és *assolida* acció si:

$$[\exists f \in result(a) \text{ tal que } valor(f) > lleugerament - possible] \vee result(a) \subset PA_d$$

Hipòtesi. El grau d'assoliment en la predicció augmenta respecte al de la configuració bàsica per tal com es redueix el nombre d'accions que necessàriament s'han d'incloure en els plans que BOLERO genera. És a dir, el nombre d'accions del patró $PA'_p(C^i)$ és inferior al nombre d'accions de $PA_p(C^i)$ utilitzat en la configuració bàsica. La resta de mesures es mantindrà indiferent perquè l'alteració no les afecta.

Resultats. Els resultats de la configuració 6 són aquests: el grau d'assoliment, tant en la identificació com en la predicció, és del 100% amb pocs casos d'entrenament (10 casos aprox). La resta de mesures romanen inalterades.

Per donar una avaluació més precisa de l'impacte del biaix conservador, hem definit l'*enfocament* d'un pla final. L'enfocament (que cal distingir del grau d'enfocament) es defineix com

$$\frac{\text{nombre d'accions assolides del pla final}}{\text{nombre d'accions del pla final}}$$

L'enfocament s'ha calculat pels casos resolts per BOLERO, i els resultats es mostren al gràfic de la figura 5.17:

⁹En altres dominis, com ara en la trajectòria de robots, podríem pensar a aprendre a no repetir accions fracassades.

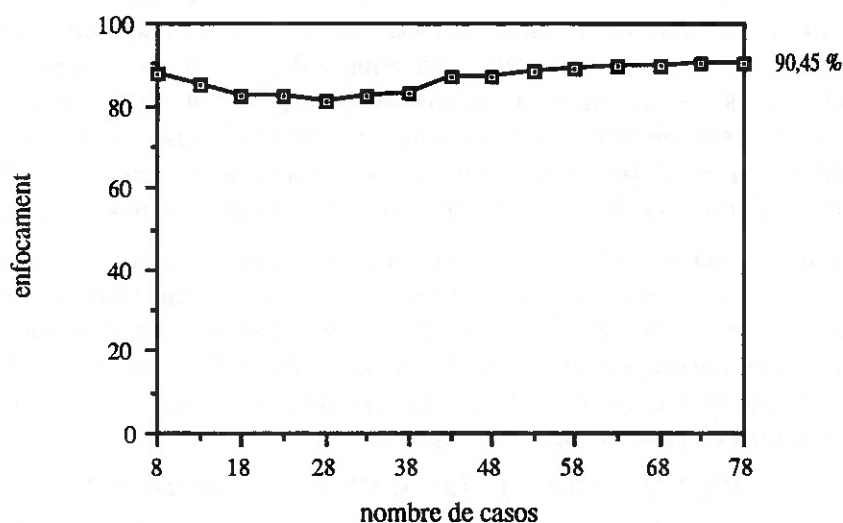


Figura 5.17: Configuració 6: Evolució de l'enfocament ($\neq$ grau d'enfocament) respecte al nombre de casos d'entrenament.

Quan BOLERO ha estat entrenat amb 79 casos, construeix plans el 90,45% de les accions dels quals s'assoleixen. És a dir, que en unes altres circumstàncies (no en un domini mèdic) podríem dir que un 10% aproximadament de l'esforç que fa BOLERO podria ser estalviat. Això vol dir que el biaix conservador causa un 10% d'*over-head* en el procediment de diagnòstic. En altres dominis, diferents al de la diagnosi mèdica, com pot ser la trajectòria de robots, BOLERO podria aprendre tenint en compte la definició del patró $PA'_p(C^*)$ de la configuració 6 i obtenir fins i tot un comportament més eficient que el del mestre.

5.4.7 Reducció del conjunt de fets

Configuració 7. La darrera configuració l'hem creada per avaluar l'impacte del vocabulari usat per BOLERO en l'aprenentatge, i.e la quantitat i qualitat dels coneixements del domini. Hem escollit el conjunt de fets estratègicament rellevants donats per l'expert mèdic i n'hem eliminat els altres fets de la consideració de BOLERO. Això fa que BOLERO passi d'usar 460 fets a usar-ne 19 (el 4,1%).

Hipòtesi. Per formular la primera hipòtesi de treball cal recordar que el mecanisme de recuperació de nodes de la memòria es basa en una ponderació de la importància estratègica dels fets. En la configuració 7, com en la configuració bàsica, la importància estratègica és per a tots els fets la mateixa: 0.5. És a dir, qualsevol fet activa en la mateixa proporció nodes en memòria. Aleshores podem pensar que en la configuració 7, on el conjunt de fets està reduït però constituït per fets estratègicament rellevants, es recuperaran menys nodes de memòria que en la configuració bàsica, on hi ha un nombre més elevat de fets; però els nodes recuperats en la configuració 7 han de ser més rellevants al cas que s'està resolent que els nodes recuperats en la configuració bàsica. Per tant, és lògic esperar que la predicció millori.

Atès que la reducció del conjunt de fets reduirà la indexació, esperem també que el temps de resposta del sistema es redueixi.

Resultats. En la nova configuració, el grau d'assoliment i el grau d'enfocament en la identificació continuen sense variar (i.e. són del 100%). El grau d'assoliment en la predicció varia i arriba al 100% només amb el 23,08% dels casos introduïts en memòria, com es mostra en la taula 5.7:

casos	complet	reduït
10.36	97.19	97.19
16.67	97.75	98.30
23.08	98.89	100.00
29.49	98.33	100.00
35.90	98.33	100.00
42.31	98.33	100.00
48.72	98.33	100.00
51.13	98.33	100.00
61.54	98.33	100.00
67.95	98.33	100.00
74.36	98.33	100.00
80.77	98.33	100.00
87.18	97.75	100.00
93.59	98.30	100.00
100	98.30	100.00

Taula 5.7. Grau d'assoliment en la predicció. La columna "complet" conté les dades referents a la configuració bàsica on el domini de pneumònia es representa amb 460 fets. Mentre que la columna "reduït" correspon a les dades de la configuració 7, on el domini està representat per un conjunt de 19 fets.

També el grau d'enfocament es veu sensiblement millorat, com mostra la taula 5.8:

casos	complet	reduït
10.36	85.22	85.22
16.67	82.23	82.76
23.08	80.24	83.58
29.49	80.30	83.11
35.90	79.88	83.82
42.31	80.93	85.76
48.72	82.43	87.93
51.13	87.01	88.35
61.54	97.92	88.75
67.95	89.17	88.75
74.36	90.00	92.08
80.77	90.42	93.75
87.18	90.42	95.00
93.59	91.25	95.00
100	91.25	95.00

Taula 5.8. Grau d'enfocament en la predicció. La columna "complet" conté les dades referents a la configuració bàsica (460 fets), i la columna "reduït" correspon a les dades de la configuració 7 (19 fets).

Una de les raons principals que justifiquen aquest millorament és que els 19 fets representats en la configuració 7 són realment significatius per a la construcció de plans. L'eliminació de fets irrelevants dona un millor biaix de vocabulari per a l'aprenentatge al sistema.

El grau d'ordenació en la identificació varia per primera vegada respecte de la configuració bàsica. També varia el grau d'ordenació en la predicció. En els gràfics de la figura 5.18 es veu l'evolució del grau d'ordenació (i.e. els graus d'enriquiment i de desencert) tant en identificació com en predicció.

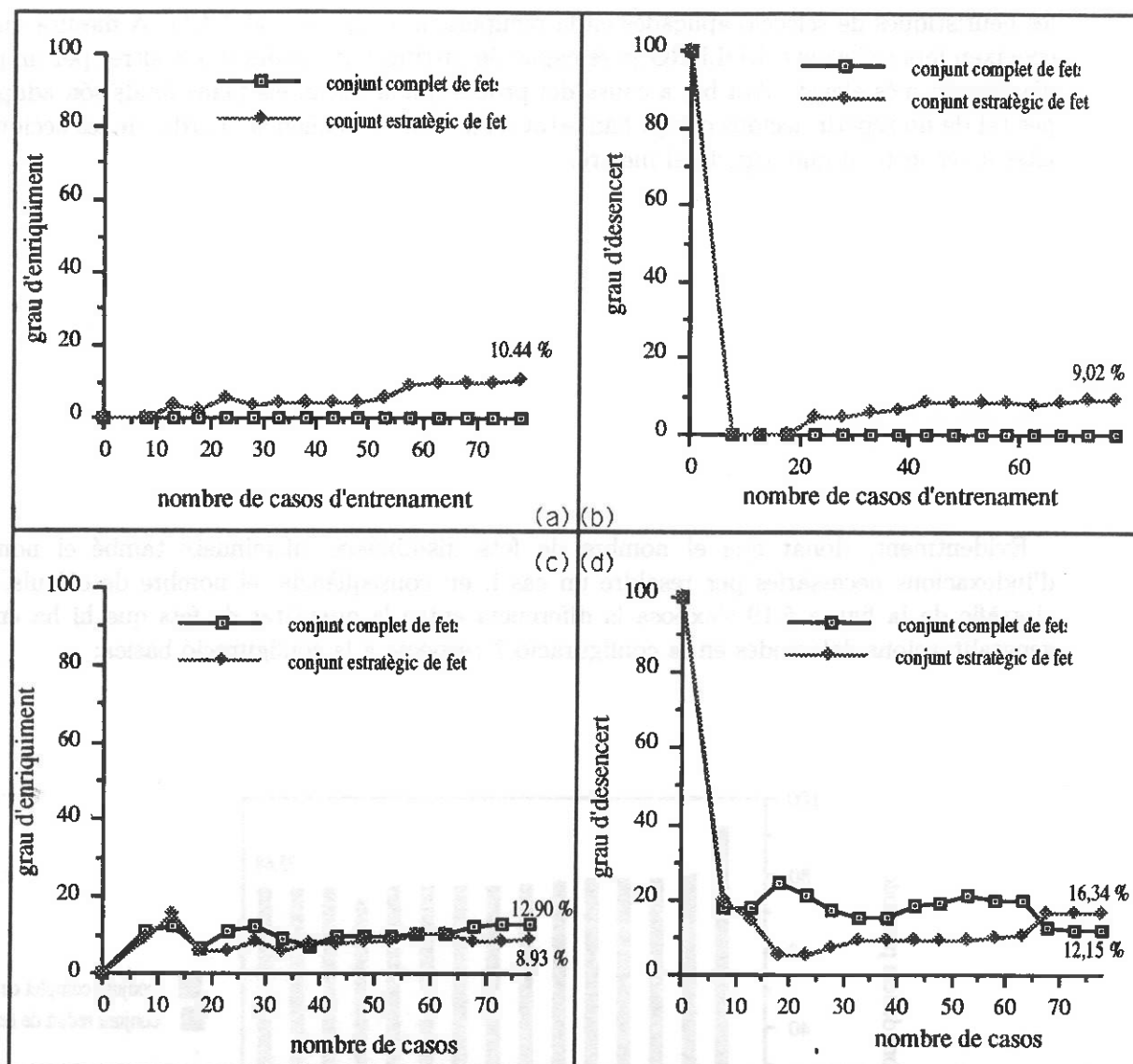


Figura 5.18: Variació del grau d'ordenació en la configuració 7 (conjunt reduït) respecte de la configuració bàsica (conjunt complet). (a) El grau d'enriquiment en la identificació. (b) El grau d'error en la identificació. (c) El grau d'enriquiment en la predicció. (d) El grau d'error en la predicció.

La raó que ens explica l'alteració del grau d'ordenació resideix en la reducció dels fets en els casos, de la mateixa manera que hem reduït el nombre de fets que representen el domini. En efecte, BOLERO s'ha després de fets irrelevants (veure secció 5.1.3) com ara el sexe, o si el pacient té tos, etc., per disposar de dades que requereixen una elaboració, com ara les dades de raig-x, d'hemocultius, etc. Els fets disponibles són pocs i no s'obtenen en un primer moment sinó després de l'execució d'alguna acció. Quan BOLERO comença a resoldre un cas, inicialment disposa de pocs fets del conjunt reduït. Amb poques dades tots els casos són versemblants, amb un grau d'activació molt baix. Aleshores els plans que construeix BOLERO es basen en nodes seleccionats, més que pel seu grau d'activació, per

les heurístiques de selecció aplicades en la recuperació (vegeu secció 4.4.1). A mesura que es coneixen fets rellevants, BOLERO ja és capaç de distingir uns nodes d'uns altres per un grau d'activació més elevat. Ara bé, a causa del procés d'adaptació, els plans finals són adaptats per tal de no repetir accions que ja han estat fetes. En conseqüència, l'ordre de les accions és alterat respecte al que segueix el mestre.

Evidentment, donat que el nombre de fets disminueix, disminueix també el nombre d'indexacions necessàries per resoldre un cas i, en conseqüència, el nombre de càlculs. En el gràfic de la figura 5.19 s'exposa la diferencia entre la quantitat de fets que hi ha en les generalitzacions dels nodes en la configuració 7 respecte a la configuració bàsica:

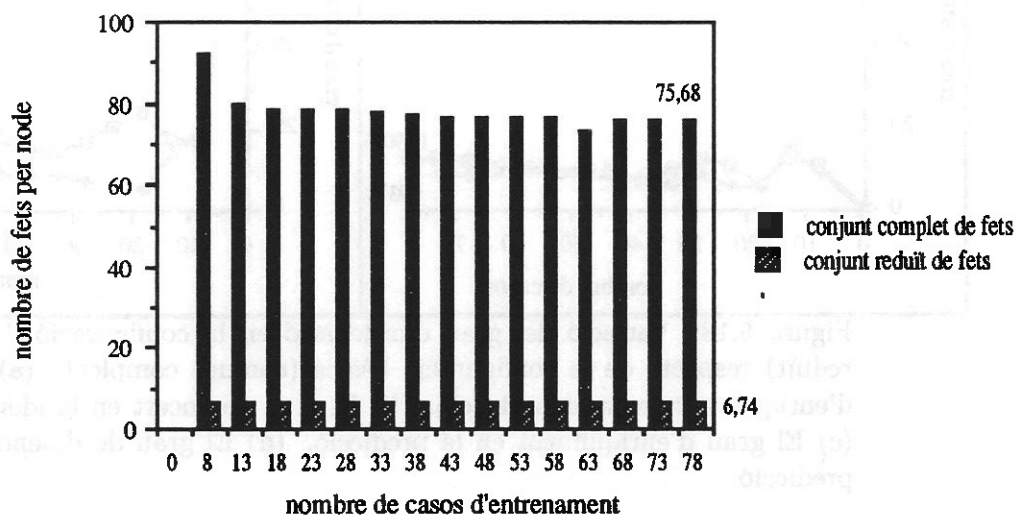


Figura 5.19: Diagrama de barres on es reflecteix, en funció del nombre de casos en memòria, la variació del nombre de fets per node (mitjana), quan BOLERO disposa del conjunt de fets complet (configuració bàsica) i el conjunt de fets reduïts (configuració 7).

Això es veu reflectit en el temps de resposta del sistema. En el gràfic següent es mostra la relació del temps de CPU necessari (en segons) per a l'aprenenatge de casos en la configuració bàsica i en la 6.

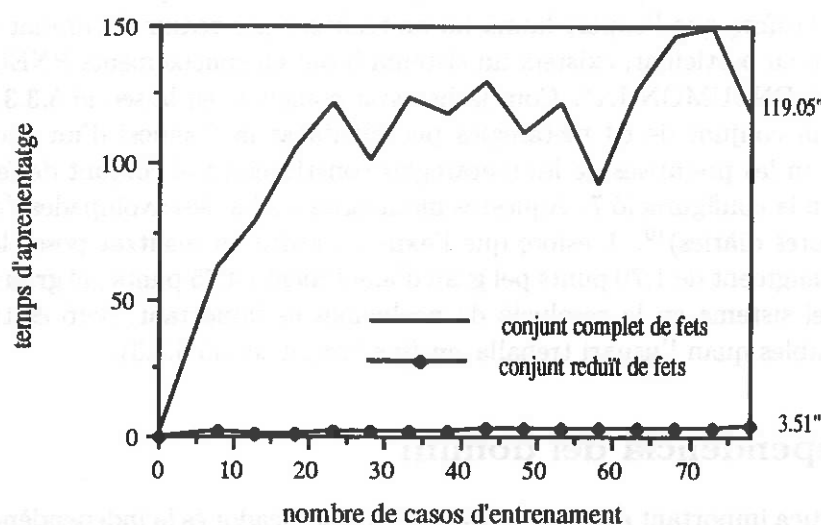


Figura 5.20: Temps de CPU (mitjana en segons) consumit per BOLERO per aprendre un cas per observació depenent del nombre de casos d'entrenament, i de si el conjunt de fets representats és complet (configuració bàsica) o reduït (configuració 7).

El temps (segons CPU) que consumeix BOLERO en la resolució d'un problema també es veu fortament afectat, com mostra el gràfic de la figura 5.21:

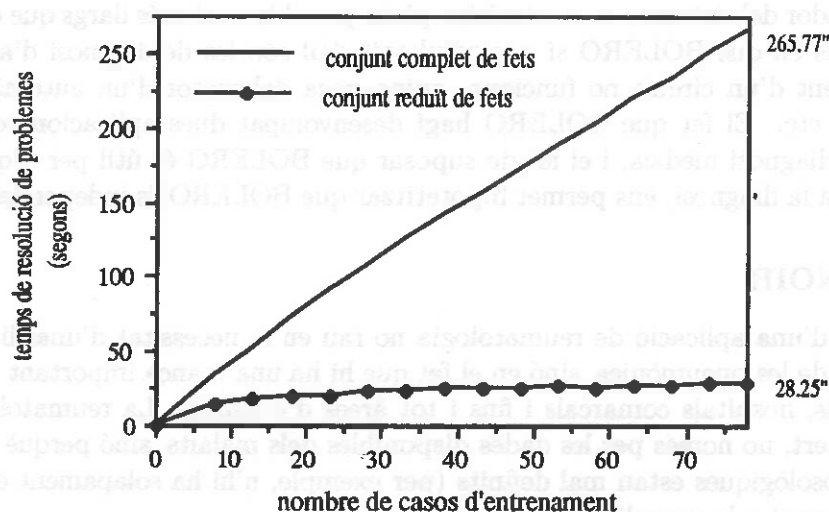


Figura 5.21: Temps de CPU (mitjana en segons) consumit per resoldre un cas en BOLERO-SBC en funció del nombre de casos en memòria, i la mida del conjunt dels fets del domini. La configuració bàsica està representada per la corba del conjunt complet de fets, i la configuració 7 per la corba del conjunt reduït de fets.

Mentre que en la configuració bàsica, BOLERO necessita cada vegada més temps per

resoldre un cas (lineal), en la configuració 7 el temps sembla estabilitzar-se al voltant dels 28". En l'avaluació de l'impacte de la configuració 7 en el temps de resposta cal tenir en compte, però, l'esforç que l'expert humà ha de realitzar per reduir el conjunt de fets. En la nostra aplicació en particular, existeix un sistema basat en coneixements PNEUMON-IA, del qual s'ha extret PNEUMON-IA*. Com ja ha estat comentat en la secció 5.3.3, PNEUMON-IA disposa d'un conjunt de 64 metaregles per planificar la diagnosi d'un pacient. Els fets que apareixen en les premisses de les metaregles constitueixen el conjunt de fets reduïts que s'ha utilitzat en la configuració 7. Aquestes metaregles les ha desenvolupades l'expert en dues setmanes (8 hores diàries)¹⁰. L'esforç que l'expert mestre ha realitzat possiblement no està justificat per l'augment de 1,70 punts pel grau d'assoliment i 4,75 punts pel grau d'enfocament. La resposta del sistema en la resolució de problemes és important, però es troba dins dels marges acceptables quan l'usuari treballa *on-line* (vegeu secció 5.3.3).

5.5 Independència del domini

Una característica important a provar en qualsevol planificador és la independència del domini. Així, BOLERO ha estat aplicat per aprendre el procediment de diagnosi de collagenosi i artropaties inflamatòries (reumatologia) [16], com es mostra a continuació en la secció 5.5.1.

Però en general BOLERO ha estat dissenyat per ser aplicat en la diagnosi mèdica, de manera que alguns dels mètodes descrits en el capítol 4 tenen incorporats biaixos per dur a terme la planificació basada en casos en aquest tipus d'aplicacions. Altres tipus de dominis mèdics, diferents a la diagnosi, com ara la terapèutica, no són aplicables en BOLERO principalment per la hipòtesi de monocitat de les situacions sota la qual s'han desenvolupat alguns mètodes. En altres dominis fora de l'àmbit mèdic, com per exemple la planificació de trajectòries de robots, els problemes no tindrien una resolució brillant en BOLERO. Donat el biaix conservador del sistema, es construirien plans possiblement més llargs que els desitjables.

Aplicacions en què BOLERO sí que resultaria útil són les de diagnosi d'avaries: trobar quin component d'un circuit no funciona, quina peça del motor d'un automòbil ha deixat de funcionar, etc. El fet que BOLERO hagi desenvolupat dues aplicacions diferents en el domini de la diagnosi mèdica, i el fet de suposar que BOLERO és útil per a qualsevol tipus d'aplicacions a la diagnosi, ens permet hipotetitzar que BOLERO és independent del domini.

5.5.1 RENOIR

La motivació d'una aplicació de reumatologia no rau en la necessitat d'una diagnosi precoç, com en el cas de les pneumònies, sinó en el fet que hi ha una manca important d'especialistes en àrees rurals, hospitals comarcals i fins i tot àrees d'urgència. La reumatologia és també un domini incert, no només per les dades disponibles dels malalts, sinó perquè els límits de les entitats nosològiques estan mal definits (per exemple, n'hi ha solapament entre malalties com la collagenosi o la vasculitis).

Per a l'aprenentatge de plans en el domini de collagenosi i artropaties inflamatòries es disposa de 81 casos. Cada cas té una mitjana d'uns 59,44 fets, el 46,51% dels quals són fets amb valor desconegut. El total de fets del domini representats és de 978 (més del doble de pneumònia), i estan repartits en 93 classes diferents (mentre que en pneumònia hi ha

¹⁰S'ha de tenir en compte que les dues setmanes estan espaiades en mesos perquè és molt difícil de disposar d'un expert de forma exclusiva durant tant de temps.

només 12 classes). Existeixen 37 accions diferents, és a dir, el 54,17% d'accions més que en pneumònia, la qual cosa comporta un nombre més alt de possibilitats de construcció de plans. Per dur a terme l'avaluació de plans es compta amb l'opinió de 12 experts. I per executar els plans es disposa del sistema basat en regles RENOIR* [16], implementat en MILORD [142] anàlogament a PNEUMON-IA* (per a més detalls vegeu apèndix C). Les característiques de l'aplicació (dimensió i complexitat) representen un desafiament per a BOLERO.

L'aplicació de reumatologia en BOLERO no ha estat experimentada totalment. Però hi ha una versió de BOLERO aplicada a la diagnòsi de collagenosi i artropaties inflamatòries en funcionament.

5.6 Conclusions

L'avaluació experimental detallada en aquest capítol permet assegurar que el sistema BOLERO ha resultat un sistema basat en casos realment factible en una aplicació real. BOLERO té una identificació correcta i precisa, independentment del nombre de casos en memòria. La predicció de BOLERO depèn del nombre de casos en memòria, però les corbes d'aprenentatge exposades en aquest capítol mostren uns resultats ben satisfactoris: 98,30% pel grau d'assoliment i 91,25% pel grau d'enfocament. Els resultats obtinguts en les diferents configuracions, on s'avaluen els diferents components del sistema, no varien substancialment. Es detecten alteracions poc significatives, i totes han estat analitzades i explicades dins del comportament de BOLERO i les característiques de l'aplicació utilitzada per avaluar el sistema.

Un resum dels resultats obtinguts en l'ablació del sistema és el següent:

1. La identificació del sistema ens indica que els plans construïts per BOLERO per resoldre un cas vist amb anterioritat són sempre correctes i precisos.
2. La predicció mostra també uns resultats generalment no inferiors al 98,20% aconseguit en el grau d'assoliment, i del 91,25% aconseguit pel grau d'enfocament. En particular:
 - (a) L'avaluació multiexpert afavoreix la construcció de plans diferents i més llargs que els que genera l'expert mestre.
 - (b) L'aprenentatge per experiència redueix el grau d'enfocament en la predicció a causa principalment de l'avaluació multiexpert i del problema de les classes alternatives, però manté l'assoliment en predicció.
 - (c) La importància estratègica dels fets juga un paper fonamental. Quan el mestre assigna a cada fet una importància estratègica d'acord amb la confiança que disposa en el fet per decidir el procediment de diagnòstic, els resultats han millorat substancialment. Però per deslliurar l'expert humà d'un treball en l'adquisició de coneixements, és millor utilitzar un mecanisme d'aprenentatge de la importància estratègica dels fets.
 - (d) El paràmetre *mdf* determina un biaix en vers quan es té una solució i cal terminar. Això afecta en gran mesura la predicció del sistema. Malgrat la seva importància, no pot ésser estimat més que experimentalment, com qualsevol altre biaix d'un algorisme d'aprenentatge.
 - (e) El biaix conservador del sistema és útil per a la diagnòsi mèdica, però en cas de prescindir del biaix conservador, BOLERO millora substancialment la correctesa dels plans. Sense el biaix conservador, i tenint en compte les accions que condueixen

a resultats (i.e. accions que no fracassen), amb pocs casos en memòria, BOLERO arriba a un 100% en el grau d'assoliment en la predicció.

- (f) El nombre de fets definits en el domini juga un paper important. Reduir el conjunt de fets, però, presenta un problema d'adquisició de coneixements. En aquest sentit, el mecanisme d'aprenentatge de la importància estratègica dels fets pot utilitzar-se per reduir el conjunt de fets. És a dir, es poden anar eliminant progressivament aquells fets, la importància estratègica dels quals disminueixi d'un cert valor (llindar).

Els resultats obtinguts, tant en identificació com en predicció, ens garanteixen un bon funcionament del sistema a partir d'un nombre poc elevat de casos: 79. El conjunt de 79 casos són molt pocs si el comparem amb altres algorismes d'aprenentatge que, en utilitzar aplicacions de laboratori - *toy problems* -, treballen amb un nombre de l'ordre de centenars de casos.

BOLERO necessita pocs casos, 8 dels 79, per obtenir una correctesa i una precisió dels plans prou satisfactòria. I a mesura que augmenta el nombre de casos en memòria, els graus d'assoliment, d'enfocament i d'ordenació milloren. Només el temps de resposta empitjora en funció del nombre de casos. Però el temps de resposta del sistema es podria millorar, si fos necessari, mitjançant el desenvolupament d'un mecanisme que redueixi el nombre d'indexacions. Aquest mecanisme pot ser un mètode que valori la conveniència d'incorporar nous casos (secció 5.3.3), un mètode de comptactació (secció 5.3.4) o un mètode que redueixi el nombre de fets irrelevants (secció 5.4.4).

L'avaluació de BOLERO és prou exhaustiva però no pas completa. Hi ha moltes característiques de BOLERO que s'escapen a l'avaluació, com són l'ús de diferents tipus de coneixements implícits en la implementació. Aquests coneixements fan referència als coneixements de domini utilitzat en el mètode de generalització i el mètode de propagació de l'activació, les heurístiques de selecció de nodes en el mecanisme de recuperació i el criteri d'ordenació en l'avaluació, entre d'altres. Aquests coneixements, que faciliten l'aprenentatge i la generació de plans, no han estat adquirits automàticament, sinó que han estat proporcionats pel mestre o per l'enginyer de coneixements. Això representa també un biaix del mètode d'aprenentatge.

Capítol 6

Treballs relacionats

En aquest capítol es tracten els sistemes que constitueixen un precedent d'algun dels mètodes descrits en BOLERO. Com s'ha posat de manifest en el capítol 1, malgrat que hi ha planificadors reactius per una banda i sistemes d'aprenentatge de plans per una altra, hi ha pocs sistemes que combinin ambdues tècniques com es fa en BOLERO. En la bibliografia, però, existeixen sistemes l'estudi dels quals ha contribuït positivament a la construcció de BOLERO i que hem recollit en aquest capítol en quatre grups. En primer lloc, en la secció 6.1, revisem altres sistemes basats en casos amb mètodes similars als que usa BOLERO. Després (secció 6.2) revisem els planificadors basats en casos coneguts amb anterioritat (CHEF [68]) o paral·lelament (SMART [162]) al desenvolupament de BOLERO. A continuació (secció 6.3) es compara la planificació reactiva del sistema BOLERO-SBC amb la d'altres sistemes també anomenats reactius. Finalment (secció 6.4), visitem altres sistemes que presenten una tasca en comú amb BOLERO: l'adquisició de coneixements estratègics.

En cada secció s'ha aprofundit en l'estudi d'uns sistemes en particular que es caracteritzen per estar ben difosos en la literatura i per haver aportat innovacions importants en la planificació reactiva, en el raonament basat en casos o en l'adquisició de coneixements de planificació en general. Així, al llarg del capítol s'analitzen els sistemes CHEF [68], SMART [162], Theo-Agent [19], ASK [63], PRODIGY/EBL [106], SAGE [93], PROTOS [11] i CASEY [88]. Tots els sistemes han estat comparats amb BOLERO observant els punts en comú i detectant els avantatges i els punts febles de BOLERO envers els sistemes descrits.

6.1 Raonament basat en casos

El raonament basat en casos és una tècnica molt utilitzada per al desenvolupament de sistemes basats en coneixements pels avantatges que presenta (veure secció 1.4). Cada sistema utilitza una representació dels casos particular en funció de la tasca per a la qual es desenvolupa (i.e. aprenentatge de coneixements de domini, de plans, etc.). I cada sistema utilitza un mètode particular per a la recuperació, adaptació i aprenentatge de casos.

Respecte a l'emmagatzemament dels casos en memòria, molts dels sistemes basats en casos, a més a més de representar generalitzacions dels casos en memòria, representen individus, i.e. casos concrets, com per exemple a JULIA [82]. A BOLERO només s'han representat generalitzacions, i de forma particular s'ha explicitat les solucions dels casos en els nodes terminals. Però a BOLERO no s'emmagatzemen casos individuals. Respecte al mètode de generalització, cada sistema basat en casos presenta un procediment de particular.

Així, per exemple, és interessant remarcar que REFINER [141] generalitza els fets partint d'una xarxa semàntica com ho fa BOLERO (i.e., els fets *taronja* i *poma* es generalitzen en el fet *fruita*). A més a més, REFINER presenta un algorisme particular per inferir rangs numèrics dels fets. Per exemple, amb els fets (*edat*, 78) i (*edat* 90) es pot inferir el fet (*edat*, [78-90]). Aquest procés de generalització de fets numèrics és interessant, però no transportable a BOLERO; l'opció de BOLERO ha estat la de tractar als fets numèrics com a enumerats en assignar-los, mitjançant un procés de normalització, intervals simbòlics (vegeu secció 4.1.1).

La recuperació de casos en memòria basada en el mecanisme de propagació de l'activació emprat a BOLERO és també una tècnica utilitzada habitualment en els sistemes basats en coneixements (per exemple, ACT [8]). Els mètodes de propagació d'activació són un model cognitiu, i.e. modelitza la manera com, aparentment, les persones utilitzem el nostre mecanisme d'inferència [62]. Els mecanismes de propagació d'activació assumeixen que l'estructura de la xarxa on es produeix l'activació correspon a l'estructura de la similitud semàntica entre els fets. Fets similars estan situats estructuralment a prop. A més a més, un mètode de propagació d'activació pressuposa que existeixen heurístiques d'avaluació per determinar si els conceptes són comparables semànticament.

En un mecanisme de propagació de l'activació es produeix una atenuació de l'activació a mesura que l'activació s'allunya del node originari. A BOLERO no s'ha desenvolupat cap mecanisme d'atenuació, però sí que es té pensat incorporar-lo en un futur. La idea consisteix en el fet que a mesura que s'activen fets en la xarxa de coneixements de domini, l'activació disminueix i repercuteixi en el grau d'activació dels nodes que es recuperen de memòria.

Es interessant distingir la propagació de l'activació del mètode de propagació de marcadors [71]. La propagació de marcadors consisteix, bàsicament, a propagar una marca (informació) en una xarxa des dels fets coneguts del problema, de manera que les marques romanen en cada node que travessa la marca. Quan dues marques es troben en un node, es diu que s'ha trobat un camí, pla o subtasca. El mètode de propagació de marcadors és un mètode desenvolupat amb propòsit computacional, però no un model cognitiu, i ha estat aplicat per ajudar a la presa de decisions en la planificació.

Tant la propagació de marcadors com la propagació de l'activació són mecanismes que permeten el paral·lisme [72]. Els processos que marquen o activen els nodes poden ser locals, i.e. realitzats per un processador simple localitzat al node i fent ús del coneixement del node i la connexió amb els seus veïns. Paral·litzar aquest mètode presenta problemes d'efectes globals que s'estan investigant. Mentrestant, els sistemes PARADYME [83], CYRUS [85] i TA [171] fan ús de tècniques concurrents per recuperar els casos de memòria, i en un futur esperem incorporar tècniques concurrents en BOLERO per tal de millorar el temps de resposta del sistema.

En la propagació de l'activació és interessant distingir una activació indiscriminada d'una activació controlada. En BOLERO s'ha dut a terme el que hem anomenat una propagació de l'activació restringida, en la qual només es permeten unes direccions determinades entre totes les possibles en la xarxa de coneixements (vegeu secció 4.4.1). A CYRUS [85] la propagació és encara més restrictiva: només s'activen els nodes i se segueixen les connexions de característiques explícitament representades en el cas o problema actual.

A més a més del mecanisme de propagació de l'activació, en el mètode de recuperació de casos de la memòria és fonamental la funció de similitud emprada, sobre les quals, paradoxalment, hi ha pocs estudis formals. Bonissone [20] proposa l'ús de valors lingüístics associats a valors semàntics per indexar i recuperar casos de memòria. Així, en lloc de dir que

una característica indexa un cas amb una força de .45 (i.e. una importància de .45), es pot dir que una característica indexa *fortament* o *molt-fortament* un pla. També en lloc de dir que el cas de memòria té un grau de *matching* de .9 amb el problema actual, es pot dir que el cas de memòria té un *matching-quasi-complet* o un *complet-matching*. En aquest sentit, en el sistema HYPO [10] s'ha desenvolupat un mecanisme per comparar casos, no partint de valors numèrics, sinó mitjançant un procés d'assignació de crèdits entre casos. A BOLERO s'han utilitzat valors numèrics per comparar casos puix que la utilització d'etiquetes o un procés d'assignació de crèdits requereix un esforç addicional, tant en l'adquisició de coneixements (i.e. establir l'etiqueta de cada índex o definir les regles que determinen l'assignació de crèdits) com en la seva manipulació.

Malgrat que les funcions de similitud són definides *ad hoc* en cada sistema, és interessant observar que, com s'ha fet a BOLERO (vegeu secció 4.4.1), molts autors utilitzen la importància (rellevància) dels fets per avaluar la semblança entre casos. A més a més de la proposta de Bonissone, a Bloom [4] s'utilitza la distància euclidiana de les característiques dels casos multiplicada per la seva importància; a CABOT [24] s'utilitza una distància euclidiana ponderada per la importància dels fets en la determinació de la similitud entre casos; i Cain [23] utilitza una mesura de similitud basada en la importància de la relació entre els fets i els casos i la importància dels fets en si mateixos. Sembla, doncs, que la utilització de la importància dels fets en la recuperació dels casos en memòria és una pràctica comuna en els sistemes basats en casos.

De tots els casos recuperats en memòria, se selecciona només un cas per adequar la solució al problema en curs. És molt comú que, com s'ha fet en BOLERO, aquesta elecció es basi en un conjunt d'heurístiques. Per exemple, a PARADYME [84, 83] s'utilitzen heurístiques basades en l'especificitat dels fets, en la seva freqüència i d'altres. La utilització d'heurístiques també s'estén a l'adaptació de les solucions dels casos recuperats per resoldre el cas o problema en curs; com per exemple es fa a CASEY [88] (vegeu la secció 6.1.2)

Finalment, en els sistemes basats en casos, l'aprenentatge per experiència consolida tot el procés de raonament. Hi ha sistemes basats en casos que aprenen a partir del reeiximent i n'hi ha que aprenen a partir del fracàs (la majoria). Entre els sistemes que aprenen a partir del fracàs cal destacar aquells que, com BOLERO, reorganitzen la memòria per tal d'evitar repetir el fracàs de nou, d'aquells que aprenen els fracassos per tal de reconèixer-los i evitar-los en el futur; com és el cas de MEDIATOR [81].

En l'aprenentatge per l'experiència, tots els sistemes presenten el mateix problema que BOLERO, que consisteix a determinar quins casos cal emmagatzemar. Lenhert [97], per exemple, analitza el nombre de casos en memòria i observa que l'augment de casos perjudica el rendiment del sistema en lloc de millorar-lo. A JULIA [82] es fa explícit el problema de determinar quins casos incorporar, però no es proposen solucions. A BOLERO el problema consisteix a aprendre a partir de casos amb plans finals, però molt similars en memòria (vegeu secció 5.3.3). La utilitat de l'aprenentatge ha estat més estudiada en altres sistemes d'aprenentatge automàtic que en els sistemes basats en casos, com es mostra en la secció 6.4.2.

Remarquem també que BOLERO distingeix una etapa d'aprenentatge diferent a l'aprenentatge per experiència, que és l'aprenentatge per observació. Com ja ha estat dit quan s'ha descrit el mètode d'aprenentatge per observació, els sistemes basats en casos o bé aprenen des de zero per l'experiència pròpia, o no deixen ben clar com s'obtenen els precedents que permeten configurar una memòria de casos i, per tant, resoldre problemes nous. A BOLERO l'obtenció de precedents es fa explícita a partir del mestre i mitjançant l'aprenentatge per observació, o eventualment BOLERO pot aprendre des de zero per l'experiència pròpia.

Entre tots els sistemes basats en casos hem destacat PROTOS [11] i CASEY [88] com a antecedents importants de BOLERO. Tots dos s'analitzen a continuació.

6.1.1 PROTOS

Descripció i funcionament.

PROTOS [11, 13, 125] és un aprenent que mitjançant l'establiment d'un diàleg amb el mestre aprèn coneixements del domini. En particular, PROTOS aprèn una definició extensional de conceptes polimòrfics (conceptes difícils de descriure per un conjunt enumerat de característiques) a partir de casos proveïts per un mestre. Això vol dir que de resultes de l'aprenentatge s'obté una definició implícita del concepte (categoria) en un conjunt d'exemplars (un exemplar és un cas retintut en memòria).

Un cas està constituït per un identificador, un conjunt de característiques i l'identificador de la categoria (concepte) a la qual pertany. PROTOS organitza els casos en memòria en una xarxa. En cada node de la xarxa hi ha una categoria, un exemplar o una característica. Els índexs es construeixen a partir d'explicacions proporcionades per un mestre i que justifiquen per què una característica pertany a un exemplar o categoria. Aprènent explicacions, PROTOS aprèn coneixements de domini (figura 6.1).

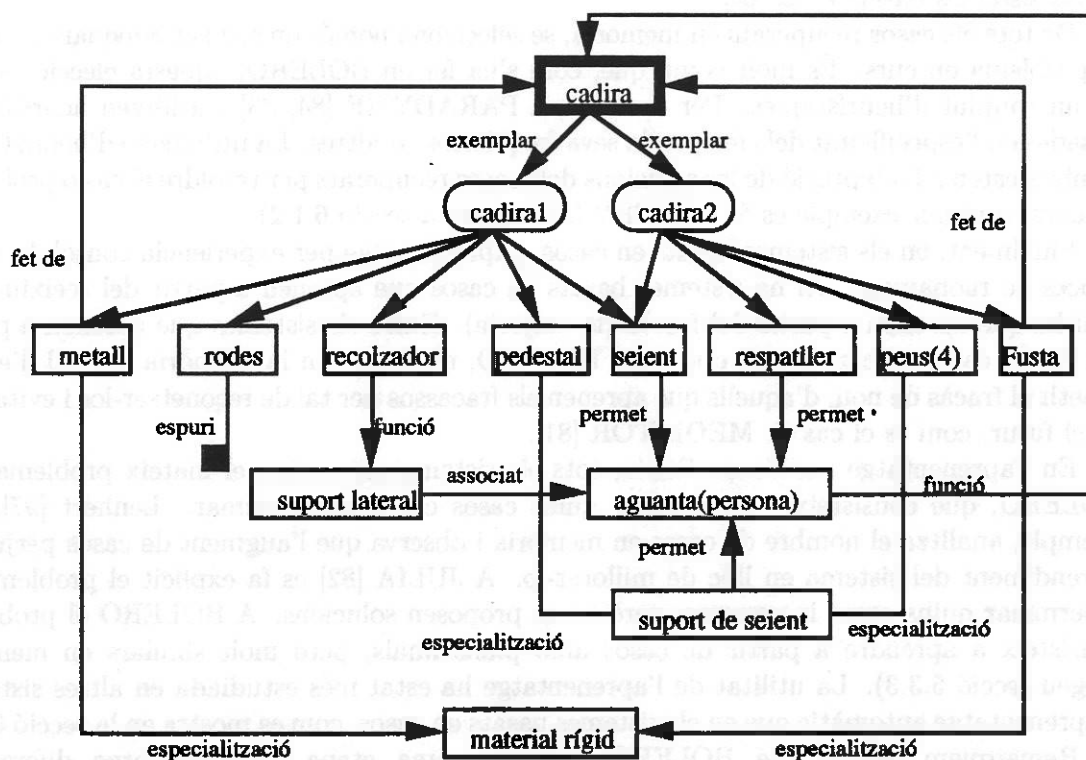


Figura 6.1: Coneixements de domini a PROTOS. Els arcs sense marcar denoten característiques (de [13]).

La memòria de PROTOS és, doncs, una xarxa de categories on es defineixen quatre mecanismes d'indexació (figura 6.2):

- *reminders*: són índexs que relacionen característiques amb exemplars o categories a les quals pertany la característica (per exemple en la figura 6.2 seient és un *reminder* de cadira).
- *restriccions (censors)*: són índexs entre característiques i categories en les quals la característica no pertany (exclusió mútua entre una característica i una categoria).
- *prototipicalitat*: freqüència amb què un exemplar apareix com a membre d'una categoria.
- *diferències*: són índexs entre exemplars d'una mateixa categoria (intracategoria) o entre categories (intercategories) i relacionen les característiques diferents entre exemplars o entre categories.

Cadascun dels índexs té una importància que reflecteix la certesa amb què s'estableix el lligam entre una característica i una categoria o exemplar.

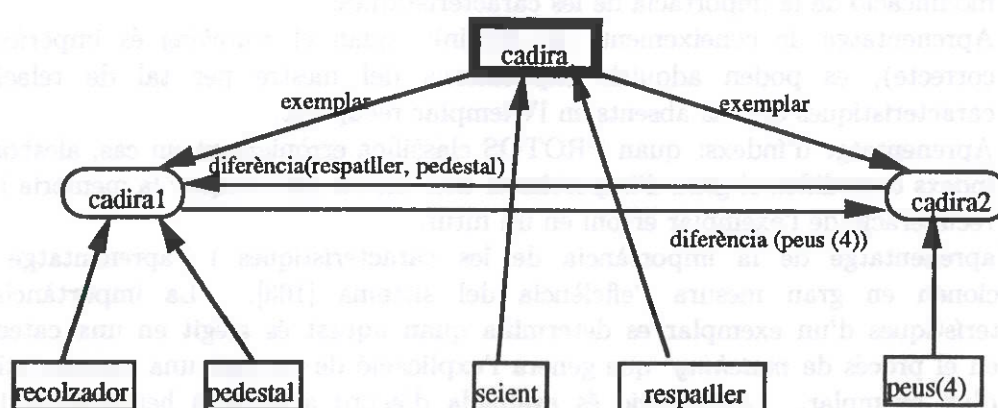


Figura 6.2: Índexs per a la recuperació d'exemplars (de [13]).

El procés de raonament a PROTOS consisteix a interpretar un nou cas i classificar-lo. La classificació consta de dues etapes: la formació d'hipòtesis i la confirmació d'hipòtesis. En la formació d'hipòtesis es recuperen categories de la memòria mitjançant els *reminders* i els *censors*. Entre les categories recuperades, PROTOS selecciona només un subconjunt dels exemplars de cada categoria tenint en compte la prototipicalitat dels exemplars. Els exemplars recuperats s'ordenen d'acord amb el grau d'importància dels *reminders* dels exemplars. La confirmació d'hipòtesis es basa en un procés de *pattern matching* basat en coneixements entre el cas i els exemplars recuperats. Aquest procés consisteix a elaborar una explicació de per què el cas correspon a l'exemplar recuperat, E_r , usant la tècnica d'aprenentatge basat en explicacions (EBL, vegeu secció 1.3). En aquest procés de *matching* s'utilitzen heurístiques per avaluar la qualitat de les explicacions i es basa en la importància de les característiques (que PROTOS obté de l'expert). Si en l'elaboració de les explicacions alguna de les característiques $F = \{f_1, \dots, f_n\}$ del cas no correspon a l'exemplar E_r (i.e. *matching* imperfecte), PROTOS pot utilitzar índexs *diferència* per cercar exemplars propers a E_r que tinguin alguna d'aquestes característiques f_i (l'estratègia de cerca d'exemplars en memòria és, doncs, una forma de *hill climbing*).

Aprenentatge en PROTOS

PROTOS aprèn a partir del fracàs, bé perquè no hi ha cap exemplar similar al cas actual en memòria, bé perquè classifica erròniament el cas. Un fracàs proporciona quatre possibilitats

d'aprenentatge:

1. Aprenentatge d'exemplars: quan un cas no pot ser classificat perquè no existeix cap exemplar similar en memòria, aleshores es reté el cas en memòria i esdevé un nou exemplar. Un cas es considera similar a un exemplar en memòria si té totes les característiques de l'exemplar excepte aquelles que es consideren de poca importància o d'importància irrellevant. En qualsevol circumstància, PROTOS pregunta al mestre sobre la conveniència de la creació d'un nou exemplar, i és el mestre qui finalment decideix retenir o no un cas en memòria.
2. Generalitzacions: quan un cas no es reté en memòria perquè és molt similar a un exemplar, aleshores es generalitza l'exemplar. La generalització es realitza bé per l'extensió dels coneixements del domini a les característiques del nou cas, bé per la modificació de la importància de les característiques.
3. Aprenentatge de coneixements de domini: quan el *matching* és imperfecte (però correcte), es poden adquirir explicacions del mestre per tal de relacionar les característiques del cas absents en l'exemplar recuperat.
4. Aprenentatge d'índexs: quan PROTOS classifica erròniament un cas, aleshores aprèn índexs o modifica el grau d'importància dels índexs per adequar la memòria i evitar la recuperació de l'exemplar erroni en un futur.

L'aprenentatge de la importància de les característiques i l'aprenentatge d'índexs condicionen en gran mesura l'eficiència del sistema [103]. La importància de les característiques d'un exemplar es determina quan aquest és afegit en una categoria i es basa en el procés de *matching* que genera l'explicació de per què una característica forma part d'un exemplar. L'explicació és avaluada d'acord amb unes heurístiques i, llavors, PROTOS assigna a la característica un valor d'importància d'entre 0 i 1 (normalment els valors inicials són baixos). Quan PROTOS fracassa en la classificació d'un cas, la importància de les característiques de l'exemplar que han dirigit la classificació són revisades. De fet, la modificació de les característiques es realitza característica a característica i sota l'assessorament del mestre. El mestre proporciona la importància de les característiques qualificant-les amb els valors següents: *necessària*, *alta*, *moderada*, *baixa* i *espúria*. PROTOS transforma aquests valors en una certesa numèrica. Quan una característica és reconeguda en un índex de tipus *diferència*, és a dir, com a element discriminant entre dos exemplars, aleshores la importància de la característica s'incrementa (i.e. es considera més rellevant).

L'aprenentatge dels diferents tipus d'índexs (*reminders*, *censors*, *prototypicalitat* i *diferències*) i els corresponents graus d'importància té lloc en diferents circumstàncies. Els *reminders* s'aprenen analíticament dels casos d'entrenament i es modifiquen quan causen una classificació incorrecta. La utilització d'un conjunt d'heurístiques guien l'aprenentatge de *reminders*. Quan un *reminder* recorda un exemplar equivocadament, aleshores la importància del *reminder* es decrementa. Quan un *reminder* indexa adequadament un exemplar, aleshores la importància del *reminder* s'incrementa. Si el valor de la importància d'un *reminder* es decrementa molt sovint, el *reminder* pot arribar-se a esborrar. Els *censors* s'aprenen per indicació del mestre. La importància dels *censors* oscil·la entre *dèbil*, *moderat*, *fort*, *absolut*. Si existeix un *sensor* amb importància *absolut*, pot provocar l'eliminació dels *reminders* implicats amb el *sensor*. La prototypicalitat d'un exemplar està relacionada amb la semblança de l'exemplar a altres membres de la categoria a la qual pertany. Cada cop que un exemplar és utilitzat per a la classificació d'un cas, es modifica la prototypicalitat de l'exemplar envers la categoria que representa. Els índexs *diferència* s'aprenen com a conseqüència d'una classificació incorrecta en el procés de resolució d'un cas (índex intercategoria) o com a

conseqüència d'un *matching* imperfecte (índex intracategoria). En ambdós casos s'aprenen per l'establiment d'un diàleg amb l'expert.

Discussió

El sistema PROTOs ha estat aplicat en audiologia per a l'aprenentatge de trastorns a l'oïda. PROTOs ha estat avaluat experimentalment amb 200 casos d'entrenament i 26 casos de test. Tots els casos de test han estat resolts correctament (i.e. el diagnòstic obtingut ha estat acceptat pel mestre). El nombre d'exemplars recuperats de memòria augmenta amb el nombre de casos d'entrenament. Això és conseqüència, segons apunta Bareiss, del fet que els darrers 50 casos d'entrenament són casos atípics. La qual cosa vol dir que PROTOs és susceptible a l'ordre dels casos en l'aprenentatge. De fet, a PROTOs no queda ben definit el mètode d'avaluació experimental, i sembla que només s'ha relitzat una única prova. En BOLERO s'han realitzat 10 proves diferents, per la qual cosa podem garantir que els resultats obtinguts són independents de l'ordre dels casos.

PROTOs aprèn del fracàs, mentre que BOLERO aprèn tant del reeiximent com del fracàs dels plans que genera. En l'aprenentatge de nous casos, PROTOs només reté els exemplars que són realment diferents als que hi ha en memòria. Altrament, generalitza els exemplars. BOLERO mai reté un exemplar, sinó generalitzacions d'exemplars.

Les idees essencials sobre la modificació de la confiança dels índexs han motivat el mètode de l'aprenentatge de la importància dels índexs en BOLERO. Però mentre que PROTOs requereix de l'expert per a la modificació del grau d'importància de les característiques i/o dels índexs, a BOLERO augmenta i disminueix la importància estratègica dels índexs de forma automàtica. En general, PROTOs depèn fortament en el diàleg amb el mestre. El mestre és qui decideix si una classificació és correcta o no, proporciona explicacions per a característiques absents en exemplars, assessora l'aprenentatge d'índexs i l'aprenentatge de la importància de les característiques, etc. BOLERO, en aquest sentit, no vincula el mestre més que en determinar la validesa d'una solució (i.e. els patrons d'avaluació) i en proporcionar els casos d'entrenament.

Referent a la importància de les característiques, és interessant remarcar que en l'anàlisi realitzada per Mallory [103] sobre la correctesa de PROTOs, destaca la contribució negativa de la importància de les característiques. La importància de les característiques és més aviat disminuïda que augmentada. Generalment s'assigna un valor baix com a importància d'una característica i, donat que el mecanisme de PROTOs és eficient (PROTOs fracassa poques vegades la classificació d'un cas en una categoria), aleshores existeixen poques possibilitats de modificar la importància de les característiques (i.e. augmentar-la).

6.1.2 CASEY

Descripció i funcionament

CASEY [88, 87, 89] és un sistema basat en casos que ha estat integrat amb un sistema de raonament causal¹. El raonament basat en casos proporciona eficiència, mentre que el raonament causal proporciona robustesa (degradació gràcil de les respostes del sistema,

¹Un model causal codifica l'estructura i el comportament d'un sistema fent explícites les relacions (causals, associatives, d'especialització, etc.) entre els components del sistema [73]. En un model causal no apareixen coneixements de control, sinó que els coneixements es representen típicament com a xarxes de descripcions estructurals [144].

flexibilitat davant de situacions no anticipades, facilitat d'explicació, facilitat de manteniment, etc. [144, 118]).

El component causal del sistema integrat és el sistema Heart Failure [88]. Heart Failure és un sistema pensat per a la diagnosi de malalties coronàries. Heart Failure té un model de coneixements basat en fets (*findings*, dades observables i valors d'entrada) i estats. Els fets i els estats estan associats, i entre els estats s'estableixen relacions causals. Heart Failure resol un problema mitjançant la construcció d'una explicació (com per exemple la de la figura 6.1.2) en què un conjunt d'estats justifica la presència dels fets "anormals" en un cas. Heart Failure construeix l'explicació per un recorregut exhaustiu de tots els estats. A partir de l'explicació, Heart Failure troba una solució (i.e. diagnòstic).

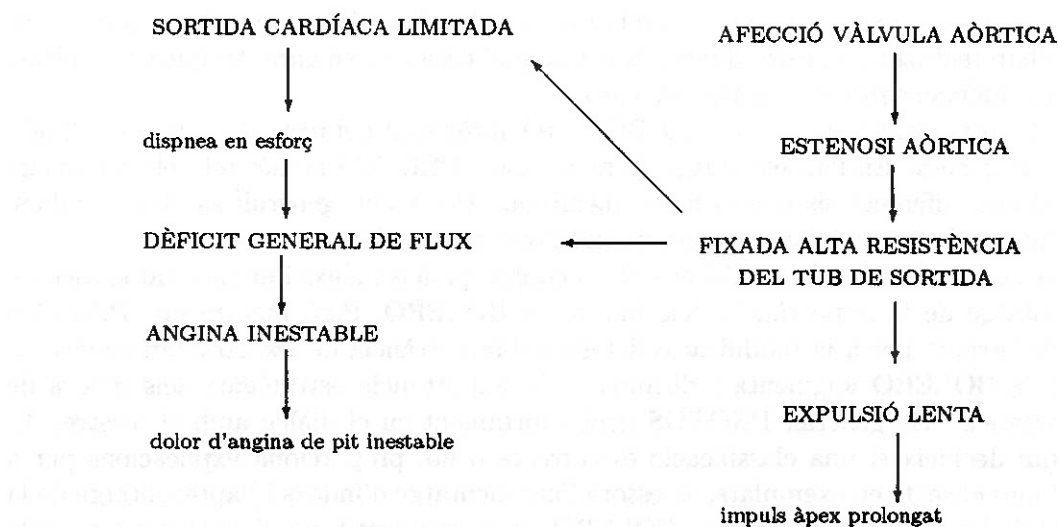


Figura 6.3: Explicació causal generada per Heart Failure per diagnosticar un pacient (de [88]). En majúscules es representen els estats; i en minúscules els fets. Els estats justifiquen la presència dels fets en el pacient.

CASEY, per una altra banda, resol problemes mitjançant l'ús d'experiències anteriors. Només quan les experiències anteriors no es poden utilitzar en la resolució d'un problema, CASEY fa ús de Heart Failure per cercar una solució. Els casos a CASEY estan representats pels fets, les solucions i els estats relacionats causalment amb els fets anormals (i.e. els estats que justifiquen la presència d'anomalies). Els casos en memòria estan indexats pels fets i les solucions. Cada fet té associada una importància que indica el nombre de vegades que apareix el fet donant suport a un estat determinat.

Els passos que realitza un sistema basat en casos, és a dir, la recuperació d'un cas en memòria, l'adaptació, l'avaluació de les solucions i l'emmagatzament, es particularitzen a CASEY per ser mètodes basats en coneixements causals. La funció de similitud que emprava per recuperar casos de memòria es basa en el nombre de característiques relacionades causalment en el cas de memòria (C_m), la importància de les característiques relacionades, les característiques anormals que presenta el cas actual (C), i el nombre total de característiques que C_m i C tenen en comú.

Un cop es recuperen casos de memòria, CASEY avalua les diferències entre un cas de

memòria C_m i el cas actual C en un procés que s'anomena justificació i que es basa en un conjunt de principis d'evidència. En la justificació es distingeixen diferències insignificants i diferències significants. Una diferència insignificant és aquella que no afecta l'explicació d'estats que apareixen a C_m . Per exemple, suposem que tenim un nou cas C amb les dades: {*dispnea-en-esforç*, *dolor-angina-de-pit-inestable*, *pols-amb-ritme-lent*} que suporten l'estat EXPULSIÓ-LENTA. Aquest cas es diferencia amb el cas C representat en la figura 6.1.2 pels fets *impuls-àpex-prolongat* i *pols-amb-ritme-lent*. Les dades dels casos C_m i C són diferents però el cas C , d'acord amb els coneixements causals, també suporta l'estat EXPULSIÓ-LENTA amb el fet *pols-amb-ritme-lent*. Per tant, malgrat que els fets no són els mateixos, els estats suportats per C_m continuen essent vàlids a C . Una diferència significativa ocorre quan no existeix cap fet de C que suporti un estat de C_m .

Algunes de les diferències són reparables. L'adaptació de les solucions consisteix en aplicar a les discrepàncies un conjunt d'estratègies de reparació. Per exemple, la regla *elimina estat*, elimina un estat de C_m no suportat per cap característica de C . Altres regles inclouen estat, relacions, etc. entre els elements de C fins a obtenir una explicació que justifica els fets anormals del cas amb uns determinats estats.

Un cop s'ha resolt un nou cas, aquest s'incorpora en memòria. No existeix una etapa diferenciada d'avaluació de la solució, ja que Koton confia en la justificació elaborada per a les diferències entre els casos per decidir la validesa de la solució, i.e. el procés de justificació assegura que els coneixements causals donen suport a la solució adaptada.

Aprentatge en CASEY

L'aprenentatge consisteix en la modificació de la importància dels fets i en la construcció de generalitzacions. La importància d'un fet es veu augmentada cada cop que el fet apareix en un cas nou (per tant, la importància de les característiques mesura la tipicitat de les característiques). La incorporació d'un nou cas en memòria pot resultar en la generalització de més d'un cas. La generalització permet agrupar característiques comunes a diferents casos. Les generalitzacions fan possible que CASEY generi explicacions parcials quan la informació d'un problema és incompleta². CASEY utilitza totes les característiques del cas per realitzar generalitzacions, raó per la qual pot considerar característiques no reflectides en el model causal. Això significa que CASEY esten el model causal a altres característiques que puguin aparèixer en casos futurs.

Discussió

CASEY ha estat avaluat experimentalment en la diagnosi de malalties coronàries. Per avaluar la correctesa de CASEY s'han comparat les solució dels casos amb les solucions que hauria proveït Heart Failure per una cerca exhaustiva de tots els estats possibles. Els casos disponibles han estat 45 per al diagnòstic de 15 malalties diferents, un nombre molt inferior al que hem disposat a BOLERO. El percentatge d'èxit a CASEY és del 86%, és a dir, el 86% dels casos obtenen la mateixa solució que a Heart Failure a excepció d'algunes característiques que no es poden justificar. L'eficiència de CASEY ha estat avaluada per la comparació entre el nombre d'estats generats en CASEY i el nombre d'estats generats per Heart Failure. CASEY sempre genera menys estats, raó per la qual resulta més eficient. L'aportació important de CASEY és, doncs, que ha demostrat l'eficiència del sistema basat en

²Aquesta particularitat de CASEY, però, no està ben explicada en la literatura.

casos envers al raonament causal: quan més casos s'han processat, menys esforç és necessari per resoldre nous casos, ja que és més probable trobar un cas molt similar en memòria i no és necessari recórrer a Heart Failure (i.e. a la cerca exhaustiva de solucions). Cal observar, però, que no es coneix en detall el procés de l'avaluació experimental realitzat en CASEY (com per exemple l'ordre dels exemples, els casos, etc.).

Una característica preocupant de l'aprenentatge de la importància de les característiques és que, com l'autor de CASEY apunta, pot provocar biaixos. Per exemple, si sempre es processa un mateix tipus de cas, les característiques presents en els casos adquiriran una importància *injustificadament* rellevant. El motiu d'aquest desajustament és que la importància dels índexs a CASEY es basa en la tipicalitat dels casos. Per tant, aquest problema no es dona amb el mètode d'aprenentatge de la importància estratègica dels índexs a BOLERO (vegeu secció 4.6.3), perquè si un cas té èxit no es modifiquen els índexs. Només es modifiquen quan BOLERO fracassa en la construcció de plans, ja que només en aquestes circumstàncies es coneix que la indexació no és adequada i cal refinar-la.

6.2 Planificació basada en casos

La diversificació dels sistemes basats en casos respecte als mètodes de recuperació, adaptació, etc. és també vàlida per als planificadors basats en casos. Cal destacar, però, que els planificadors basats en casos presenten una característica que BOLERO no posseeix i que és la capacitat de reparar plans que fracassen. BOLERO aprèn a partir de plans imperfectes (tant com de plans reeixits), però si un pla és imperfecte BOLERO no genera un pla alternatiu. BOLERO pregunta el pla correcte al mestre. Tanmateix, hi ha planificadors, com ara NEGOTIATOR [85] o CHEF [68], que són capaços de, un cop fracassat el pla, reparar-lo i generar un pla nou mitjançant les adaptacions adequades que evitin el fracàs, o bé recuperant un pla alternatiu de memòria.

Existeixen pocs planificadors basats en casos, entre els quals hem destacat en aquesta secció CHEF [68] i SMART [162]. Cal observar que ni CHEF ni SMART són planificadors reactius: cap d'ells alterna la construcció de plans amb la seva execució. A més a més, els dominis en els quals s'han desenvolupat són dominis artificials, mentre que BOLERO ha estat aplicat i avaluat en una aplicació real.

6.2.1 CHEF

Descripció i funcionament

CHEF [68] és el primer sistema on s'integra el raonament basat en casos, i la planificació resultant la planificació basada en casos (*case-based planning*). CHEF està dissenyat per a l'aprenentatge de receptes de cuina xinesa: a partir dels ingredients i del tipus de plat que es vol cuinar CHEF es capaç de contruir una recepta (pla) basant-se en receptes anteriors.

Com a sistema basat en casos, el més important de CHEF és la seva peculiar organització en la memòria d'experiències (casos) anteriors. CHEF organitza els casos en tres organitzacions dinàmiques de memòria (organitzacions actualitzables amb la presència de noves experiències, i.e. casos nous) i una organització estàtica (no modificada per noves experiències). Entre les dinàmiques hi ha una memòria de plans, una memòria d'errors (fracassos) i una memòria de modificacions. La memòria estàtica està constituïda per un conjunt d'estratègies de reparació de plans.

Les diferents memòries comparteixen estructures de dades com són els objectes, els estats, els objectius, els operadors (que en CHEF s'anomenen passos, *steps*) i els plans. Els objectes representen coses del món com ara els ingredients i els utensilis de cuina. Els objectes tenen diferents estats o característiques: textura, gust, etc. Els objectius refereixen l'estat dels objectes que es vol aconseguir, com per exemple un pollastre cruixent o un *soufflé* esponjós. Els passos són un conjunt de pre-condicions més un conjunt d'accions. Cada pla (figura 6.4) és un conjunt de passos, més un conjunt de condicions que els fan possible (ingredients). Existeixen diferents tipus d'enllaços entre les diferents estructures, com per exemple: entre passos i els objectes que modifica, entre estats i característiques desitjables dels objectes, etc.

En la memòria de plans s'organitzen els plans d'acord amb el tipus que tenen associat i els objectius que solucionen. És a dir, cada pla ha de ser d'un tipus predeterminat, com per exemple un pla de fritures, *soufflés*, etc. El tipus de pla serveix per indexar els plans en la memòria conjuntament amb el objectius que satisfà el pla. Per manipular la memòria de plans, CHEF disposa d'uns coneixements generals (*role specifications*) que determinen els tipus d'objectius que poden afegir-se en un tipus de pla i els resultats que s'esperen de l'execució del pla.

```
(def: rec broccoli-with-tofu
  (ingredients
    ingr1 (tofu lb .5)
    ingr2 (soy-sauce tablespoon 2)
    ingr3 (rice-wine spoon 1)
    ingr4 (corn-starch tablespoon .5)
    ingr5 (sugar spoon 1)
    ingr6 (broccoli lb 1)
    ingr7 (r-pepper piece 6))
  (actions
    act1 (chop object (ingr1) size (chunk))
    act2 (marinade object (result act1)
      in (& (ingr2) (ingr3) (ingr4) (ingr5))
      time (20))
    act3 (chop object (ingr6) size (chunk))
    act4 (stir-fry object (&result act2) (ingr7)) time (1))
    act5 (add object (result act3) to (result act4))
    act6 (stir-fry object (result act5) time (2)))
  (style style-stir-fry))
```

Figura 6.4: Pla (recepta) per cuinar bròquil amb costelles (*tofu*). (Fig. 3.1, [68] pag. 73).

En la memòria d'errors s'organitzen els errors, una explicació causal de l'error i característiques predictives de l'error (característiques que es trobaven en memòria quan l'error es va produir). Un error és un objectiu no satisfet en un pla o un estat no desitjat que ha sorgit de l'execució d'un pla. Els errors es reconeixen per l'activació de les característiques que els prediuen i que activen a la vegada els plans.

La memòria de modificacions la constitueixen una taula de regles i un conjunt de crítics. La taula de regles està indexada pels tipus de plans i objectius, i en cada cella se situen el conjunt de passos necessaris per afegir un objectiu a un pla. Els crítics són regles d'aplicació específica sobre objectes (figura 6.5).

La memòria de reparacions organitza estratègies de reparació sota descripcions dels diferents problemes que reparen. L'organització d'aquesta memòria és realment complexa, i donat que el propòsit d'aquest treball no és explicar la particular representació de coneixements a CHEF, ens valdrà saber que a partir d'una explicació causal d'un fracàs (error) és possible recuperar una estratègia que repara el pla en curs.

CHEF recupera un cas de la memòria de plans a partir del tipus de plan que es pretén construir (per exemple una **fritura**) i els ingredients (per exemple **vedella** i **patates**). A partir dels ingredients i del tipus CHEF dedueix els objectius que s'han de satisfer (com per exemple obtenir unes **patates cruixents**). Amb els objectius i el tipus de pla CHEF accedeix a la memòria de plans i recupera un pla del mateix tipus on tots els objectius són satisfets. A continuació CHEF adapta el pla recuperat al problema actual a partir de la memòria de modificacions. Així, si hi manca algun ingredient s'accedeix a la memòria de modificacions per incloure el tractament oportú en el pla recuperat. Finalment, CHEF emmagatzema el nou pla en la memòria de plans. Un cop s'ha obtingut un nou pla, el pla és executat per un simulador que utilitza un conjunt de regles d'inferència contruïdes per a tal propòsit. En cas que aparegui algun error no esperat, es considera que el pla és erroni, i.e. fracassa.

Aprentatge de plans

El fracàs dels plans dona a CHEF l'oportunitat d'aprendre. CHEF és capaç, a partir de les mateixes regles que el simulador utilitza, de generar una explicació causal del mal funcionament del pla. Un cop coneguda la causa del fracàs del pla, CHEF utilitza la memòria de reparacions per recuperar una estratègia de reparació que repari el pla. Aquest nou pla s'emmagatzema en la memòria de plans. L'explicació causal del fracàs també s'utilitza per actualitzar la memòria d'errors i així evitar que es torni a repetir l'error en una situació similar. D'aquesta manera, quan CHEF resol un nou problema, accedeix a la memòria d'errors i esbrina si existeix la possibilitat que es produeixi algun error ocorregut amb una experiència anterior amb les mateixes dades (i.e. ingredients i tipus de pla). Si és així, es genera un objectiu nou (predicció) que s'afegeix a la llista d'objectius del problema a resoldre. Resolent tots els objectius s'està, a la mateixa vegada, evitant el fracàs del pla.

Cal remarcar que tots els plans que s'emmagatzemen a CHEF tenen els índexs generalitzats. Això vol dir que qualsevol dels ingredients de les receptes es generalitzen abans de ser emmagatzemats. Per exemple, una recepta amb bou i bròquil, estarà emmagatzemada sota els índexs **carn** i **verdures**. En la recuperació d'una recepta, també s'accedeix a la memòria de plans mitjançant un procés d'abstracció dels ingredients. Per exemple, si l'ingredient és **col-i-flor** s'accedeix a la memòria per l'índex **verdura**, i a partir d'aquest es dedueixen les característiques desitjades (textura, etc.) o els objectius.

```
(add:crit shrimp binds (shrimp *new-item*)
  steps ((before (cook-step object *new-item*) do (shell object *new-item*)))))
```

Figura 6.5: Regla d'aplicació específica sobre l'ingredient gamba (*shrimp*).
(Fig. 5.3, [68, pag. 146])

Discussió

CHEF és un sistema de planificació basat en casos on la principal idea és la reusabilitat de plans executats en experiències anterior més que la construcció de nous plans. L'aprenentatge en CHEF es realitza quan un pla resulta erroni en la seva execució. Els errors s'organitzen en la memòria per tal d'evitar-los en el futur. CHEF, però, no aprèn a partir de plans satisfactoris (positius) com ho fa BOLERO. L'aportació principal de CHEF ha estat la d'integrar per primera vegada la planificació i el raonament basat en casos. Tanmateix, la representació dels coneixements a CHEF és molt complexa, com ho demostren les seves tres memòries dinàmiques, més una estàtica, més altres tipus de coneixements (crítics, regles de simulació, etc.) que intervenen en el procés de resolució d'un nou problema. A més a més, el domini d'aplicació de CHEF és un domini particular i seria interessant estudiar com es comporta el sistema en altres dominis. És important destacar que no hi ha hagut una metodologia d'avaluació del comportament del sistema. No s'han proporcionat corbes d'aprenentatge ni del cost de la resolució de problemes. Tota la funcionalitat de CHEF està basada en exemples particulars.

El simulador incorporat en CHEF per tal d'avaluar els plans que construeix ha estat eliminat en els sistemes successors TRUCKER i RUNNER [67]. En aquests sistemes l'execució dels plans es monitoritza, i és permesa la suspensió i la continuació d'objectius quan les circumstàncies ho requereixen. Cal no confondre TRUCKER i RUNNER amb els planificadors reactius, sinó que es tracta de planificadors oportunistes. La monitorització de l'execució no està vinculada a l'adequació de la planificació als canvis que se succeeixen a l'entorn, sinó que s'utilitza per dur a terme aquelles parts de plans que no han pogut ser satisfetes en el moment de la seva construcció i que s'ajornen fins que les condicions que permeten aplicar-les s'acompleixen.

6.2.2 SMART

Descripció i funcionament

Manuela Veloso ha desenvolupat un sistema basat en l'experiència, SMART [162], per fer més eficient la planificació en PRODIGY [32] (*speed-up learning*). PRODIGY és una arquitectura de propòsit general on s'han desenvolupat principalment aplicacions de planificació o cerca. La representació dels coneixements en PRODIGY està constituïda per operadors, regles d'inferència i regles de control (figura 6.6). Els operadors i les regles d'inferència estan constituïts per una pre-condició i una llista d'efectes. Les pre-condicions són les condicions d'aplicabilitat de l'operador o la regla. Els efectes dels operadors canvien l'estat intern del sistema i el del món. Els efectes de les regles d'inferència canvien només l'estat intern del sistema (de fet, una regla d'inferència només expressa les relacions entre els objectes del domini). Les regles de control estan constituïdes per una condició i una acció (selecció, filtrat o preferència) sobre els operadors o regles d'inferència.

SMART interacciona amb NoLimit [161], el planificador no-lineal de PRODIGY, per tal de resoldre problemes. NoLimit resol un problema per l'aplicació d'operadors, regles d'inferència i regles de control mitjançant l'anàlisi dels medis en relació als objectius (*means-ends analysis*) i l'encadenament endarrera (*backward chaining*) de les regles o operadors. Donat un estat inicial E i els objectius, NoLimit aplica operadors o regles a E transformant-lo en un nou estat E' que s'acosta més a un estat final on tots els objectius són satisfets. La resolució d'un problema és, doncs, la seqüència d'operadors necessaris per arribar de l'estat inicial a

```

(a) (POLISH
      (params (obj time prev-time))
      (preconds
        (and (is-object obj)
              (or (clampable obj POLISHER)
                  (shape obj RECTANGULAR))
              (last-scheduled obj prev-time)
              (later time prev-time)
              (idle (scheduled obj POLISHER time))))
      (effects
        ((del (surface-condition obj old-cond))
          (add (surface-condition obj POLISHED))
          (del (last-scheduled obj prev-time))
          (add (last-scheduled obj time))
          (add (scheduled obj POLISHER time))))))

(b) (IS-CLAMPABLE
      (params (obj machine))
      (preconds
        (and (has-clamp machine)
              (temperature obj COLD)))
      (effects
        ((add (clampable obj machine)))))

(c) If (and (CURRENT-NODE node)
            (CURRENT-GOAL node (SHAPE obj shape))
            (CANDIDATE-OPERATOR node ROLL)
            (CANDIDATE-OPERATOR node LATHE))
    Then (PREFER OPERATOR ROLL to LATHE)

```

Figura 6.6: Representació dels coneixements a PRODIGY: (a) L'operador POLISH; (b) La regla d'inferència IS-CLAMPABLE; i (c) Una regla de control. (de [106]).

l'estat final. En la resolució d'un problema es produeixen decisions sobre l'aplicabilitat dels operadors i regles, i NoLimit desenvolupa un arbre on es reflecteixen totes les decisions que es prenen. Per començar a resoldre un problema es defineix un estat inicial i els objectius que es volen assolir, que constitueixen el node arrel de l'arbre de cerca. Cada decisió que es realitza durant la resolució d'un problema dóna lloc a un nou node. És a dir, per a cada operador, regla d'inferència o regla de control que NoLimit aplica o cada vegada que NoLimit es troba una instància de les pre-condicions dels operadors (o regla), es crea un nou node. Sempre que s'arribi a un estat en què no es pugui continuar la resolució d'un problema (*impass*), NoLimit pot recuperar un estat anterior gràcies a un sistema de manteniment del raonament (*reason maintenance system*) incorporat a l'arquitectura de PRODIGY.

NoLimit, com a planificador no-lineal (vegeu secció 1.1.1), treballa amb un conjunt d'objectius en un mateix cicle. A més a més, NoLimit segueix una estratègia de compromís casual (*casual commitment strategy*) que tracta de prendre decisions (quin objectiu expandir, quin operador elegir, quines assignacions realitzar, etc.) mitjançant l'ús de regles de control, mitjançant la intervenció de l'usuari o mitjançant la interacció amb SMART (figura 6.7). NoLimit pot interaccionar amb SMART per obtenir anàlegs que li permeten prendre decisions seguint les decisions preses en l'anàleg, i així evitar repetir els errors que es puguin haver realitzat en experiències prèvies. Si no és possible trobar un anàleg, aleshores NoLimit resol un problema utilitzant la resta dels seus coneixements.

NoLimit proporciona una traça que conté l'arbre de cerca desenvolupat per construir un pla. SMART utilitza les traces per generar casos. En els casos SMART representa el camí de la solució de l'arbre de decisió amb anotacions dels punts de decisió (i.e. els motius i context

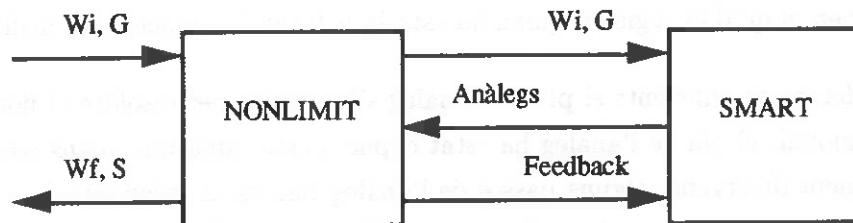


Figura 6.7: Interacció entre el resolador de problemes per analogia NoLimit i el gestor de memòria SMART (de [163]). W_i és l'estat inicial, G l'objectiu, W_f l'estat final, S la solució, Anàlegs els casos recuperats de memòria i Feedback la solució més informació sobre la rellevància dels anàlegs.

que han donat lloc a les decisions, així com les alternatives inexplorades o abandonades). Per exemple, l'aplicació d'un operador O es registra com un node en el qual s'emmagatzema informació sobre els operadors alternatius, els objectius que s'estan estudiant i els arguments que justifiquen per què s'aplica precisament l'operador O . SMART organitza els casos en memòria indexats per l'estat inicial (constituït per un conjunt de fets) i els objectius.

Donat un nou problema P , es recupera de memòria un cas anàleg fent ús de les dades de l'estat inicial i l'objectiu. Per recuperar un anàleg, SMART utilitza una mesura de similitud basada en "empremtes" (*foot-print*) i substitucions. Una empremta d'un objectiu la constitueixen els fets de l'estat inicial que són rellevants en la validació d'un objectiu. El grau de semblança entre el problema actual P i un cas en memòria C , es determina pel nombre de fets que comparteixen les empremtes dels objectius comuns a P i C . Per comparar fets d'empremtes diferents s'utilitzen substitucions. Donats dos conjunts de fets F_1 i F_2 (i.e. empremtes), es troba una substitució σ que s'aplica a F_2 , i es compta el nombre de fets idèntics entre F_1 i F'_2 , éssent F'_2 el resultat d'aplicar σ a F_2 [163]. Per exemple, si $F_1 = \{(dintre\ clau12\ habitació1), (dintre\ caixa1\ habitació1)\}$, $F_2 = \{(dintre\ clau13\ habitació4), (dintre\ clau14\ habitació2), (dintre\ caixa53\ habitació4)\}$, i $\sigma = \{clau12/clau13, habitació1/habitació4, caixa1/caixa53\}$, el grau de semblança és 2.

NoLimit utilitza l'anàleg per derivar analògicament (*derivational analogy* [30, 28]) una solució. La derivació analògica significa construir la solució d'un cas fent ús dels passos de decisió que han tingut lloc en un cas anàleg (experiència previa). És a dir, en lloc de generar un arbre de cerca per al cas actual, es torna a generar el camí solució del cas anàleg. En aquesta reconstrucció de la solució es verifica que les condicions que han donat pas a decisions (punts de decisió) en el cas anàleg es compleixen en el cas actual. Si totes les justificacions explícites en l'anàleg es satisfan, el pla de l'anàleg es copia directament per resoldre el nou cas. Si existeix alguna condició d'algun operador que no es compleix, aleshores existeixen tres possibilitats: replanificar l'operador que fracassa, reestablir les condicions que manquen com a objectius anteriors o experimentar per tal de justificar les condicions que no se satisfan (amb el mòdul d'aprenentatge per experimentació de PRODIGY [31]).

Aprenentatge per experiència

L'aprenentatge per experiència a SMART es dona per la reorganització de la memòria amb la inclusió d'un nou cas resolt per NoLimit (interaccionant amb SMART). NoLimit proporciona

un *feedback* en el qual es registra quina ha estat la utilitat de l'anàleg. La utilitat d'un anàleg pot ser:

1. completament suficient: el pla de l'anàleg s'ha copiat per resoldre el nou problema,
2. extensional: el pla de l'anàleg ha estat copiat i està amb uns quans passos més,
3. localment divergent: alguns passos de l'anàleg han estat invalidats, i
4. globalment divergent: el resolador de problemes ha hagut de fer marxa enrera amb alguns dels passos copiats de l'anàleg.

La utilitat és una informació usada per incloure un nou cas en memòria i ajustar els índexs de l'anàleg. Aquesta part de l'aprenentatge, però, està encara en fase de desenvolupament en SMART.

Discussió

De la mateixa manera que passa amb el SBC^B de l'arquitectura BOLERO-SBC, NoLimit és capaç de resoldre problemes. SMART proporciona eficiència a NoLimit, com BOLERO a SBC^B . Però SMART no pot aprendre d'un mestre, com ho fa BOLERO. El que fa SMART quan no té cap cas en memòria és deixar que NoLimit resolgui casos utilitzant altres estratègies com l'aplicació de regles de control o el diàleg amb l'usuari. Els casos resolts per NoLimit sense l'ajuda d'SMART poden incorporar-se a SMART, constituint el conjunt de precedents a partir dels quals resoldre nous casos. La diferència fonamental entre els casos proporcionats pel mestre a BOLERO, i els precedents d'SMART obtinguts per la resolució d'un problema en NoLimit, rau en el fet que els casos de NoLimit inclouen informació sobre els operadors disponibles en cada pas de decisió, quin operador s'ha escollit envers un conjunt i altre tipus d'informació que un resolador de problemes pot proporcionar, però que és difícil d'adquirir d'un expert humà.

SMART ha estat utilitzat en diferents aplicacions de planificació, com el món dels blocs. SMART és, doncs, un planificador independent del domini com, hipotèticament, ho és BOLERO. En el domini dels blocs, l'avaluació dels plans resulta més senzilla que en la diagnòsi mèdica: es coneix els efectes dels operadors (i.e. accions) abans de la seva execució i, per tant, es pot simular l'execució. Com a conseqüència es coneix la solució moltes vegades amb anticipació a l'aplicació del pla. En la diagnòsi mèdica això no és possible, i a més a més moltes vegades no es coneix la solució exacta d'un cas.

Els resultats d'SMART indiquen que la utilització d'experiències anteriors permeten estalviar a NoLimit fins a 20 vegades l'esforç de cerca. Respecte del problema de l'escalament, SMART funciona amb un nombre elevat de casos, però l'escalament no fa referència només a la quantitat de problemes sinó a les característiques de la informació que es tracta en les aplicacions reals (informació incompleta i incerta, no sistematitzada, etc.; vegeu secció 1.1.2).

L'aprenentatge per experiència a SMART està poc desenvolupat. Existeixen indicis que permeten pensar que, després de la resolució d'un problema, es pot avaluar la utilitat del cas recuperat en memòria, en la qual basar l'aprenentatge. L'aprenentatge per experiència de SMART consistiria, doncs, a modificar la importància de les característiques definides en les empremtes. Les empremtes s'utilitzen per indexar els casos i s'hi poden distingir característiques més rellevants que altres. Modificar la importància de les característiques de les empremtes en funció de la utilitat del cas recuperat equival, doncs, a alterar la indexació. Res de tot això està implementat actualment a SMART.

6.3 Planificació reactiva

Existeixen pocs planificadors reactius donat que la planificació reactiva és una tècnica relativament recent (vegeu secció 1.1.3). Existeixen diferents aproximacions a un comportament reactiu: la utilització de regles de tipus estímul-resposta de Pengi [3]; la construcció de plans universals (i.e. plans que no es comprometen en cap seqüència d'accions però reaccionen a l'entorn [139]); la planificació no-lineal reactiva (IPEM [7]); l'especificació de situacions i la reacció adequada en procediments declaratius (PRS [57]); l'ús de produccions de control especials (RoboSoar [90]), i la definició de diferents nivells i comportaments reactius, uns més elaborats que altres [75]. Entre totes aquestes aproximacions cal distingir RoboSoar [90] i Theo-Agent [19], que estan definits en arquitectures cognitives [160], i.e. arquitectures que proporcionen marcs generals per al desenvolupament de sistemes basats en coneixements. Però mentre RoboSoar ha resultat de l'adequació d'un planificador clàssic per interaccionar amb l'entorn extern, Theo-Agent ha estat desenvolupat des d'un principi per reaccionar a canvis, a l'entorn. Per aquesta raó, entre els dos sistemes hem considerat més convenient analitzar Theo-Agent en aquesta secció.

6.3.1 Theo-Agent

Descripció i funcionament

Theo-Agent [19, 107, 108] és un planificador desenvolupat en Theo, una arquitectura de propòsit general per a l'aprenentatge [110]. La reacció de Theo-Agent es basa en el temps de resposta de Theo-Agent depèn del temps que triga a executar-se un cicle de control. El cicle de control de Theo-Agent consta de tres passos: (1) recollida de dades dels sensors (inputs), (2) selecció d'una acció per ser executada i (3) execució de l'acció escollida. És interessant remarcar que perquè la quantitat d'inputs que rep Theo-Agent sigui manipulable es distingeixen els sensor *eager* dels sensors *lazy*. Només els inputs dels sensors *eager* es verifiquen en cada cicle, i els inputs dels sensors *lazy* només intervenen en la planificació quan resulten rellevants³. La selecció d'un acció es basa en l'aplicació d'una regla d'estímul-resposta (vegeu secció 1.1.3) o en la construcció d'un pla. En el darrer cas els objectius de Theo-Agent són utilitzats explícitament per elaborar un pla, i la primera acció del pla és l'elegida per ser executada immediatament.

Aprenentatge de regles estímul-resposta

La construcció de plans és una ocasió que aprofita Theo-Agent per a aprendre noves regles d'estímul-resposta mitjançant un mecanisme d'aprenentatge basat en explicacions (EBL, vegeu secció 1.3.3). Theo-Agent, basant-se en la situació del món (conjunt d'inputs) en la qual s'ha engegat la planificació i la primera acció seleccionada (i.e. la primera acció del pla), genera una explicació que justifica la selecció de l'acció. Per exemple, per recollir una tassa i dipositar-la a la paperera, Theo-Agent ha desenvolupat un arbre de cerca, com el que mostra la figura 6.8, a partir del qual genera una explicació.

L'explicació és utilitzada per operacionalitzar una regla "objectiu" (*target*) que explicita quan una acció és aplicable, i que resulta en una nova acció estímul-resposta. Per exemple la figura 6.3.1 mostra la regla estímul-reposta apresada per a l'acció seleccionada en la figura 6.8. Les regles apresades poden ser aplicades en la resolució de nous problemes.

³Un dels problemes de Theo-Agent és, precisament, controlar automàticament quan un sensor és rellevant.

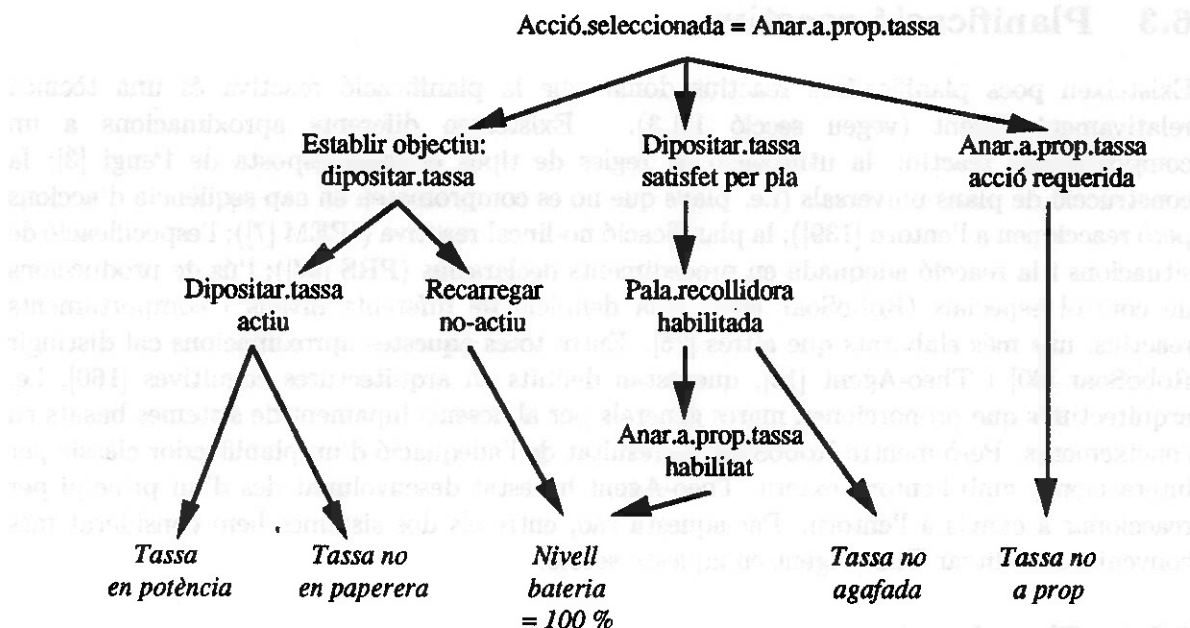


Figura 6.8: Arbre de decisió per seleccionar una acció (de [107]).

REGLA2:

- Si (c1) una tassa és en el camp visual
 (c2) la tassa no està a la paperera
 (c3) la bateria està al 100%
 (c4) la tassa no ha estat agafada
 (c5) la tassa no és propera

Llavors acció-seleccionada = anar.aprop.tassa

Figura 6.9: Regla de tipus estímul-resposta (adaptat de [107])

El mecanisme d'aprenentatge en Theo-Agent es coneix també com a compilació (*compilation*), en el qual es transformen descripcions d'inputs de les accions en regles de control efectives [135, 109].

Discussió

Theo-Agent és un planificador reactiu que, a més a més, aprèn de l'experiència. Però Theo-Agent no aprèn plans com ho fa BOLERO, sinó reaccions instantànies a situacions particulars a partir de plans. Theo-Agent esdevé més reactiu amb l'experiència, però, donat que en les regles d'estímul-resposta no es reflecteixen els efectes de les accions, Theo-Agent no esdevé necessàriament més correcte. A més a més, com ja ha estat introduït en el primer capítol (secció 6.3), els sistemes reactius basats en regles de tipus estímul resposta poden induir cicles en la resolució de problemes.

Un altre problema important en Theo-Agent és determinar el cost de l'aprenentatge i de la utilitat de les regles apreses. Una nova regla pot tenir una expressió força complexa que sigui costosa d'avaluar. A més a més, Theo-Agent presenta problemes d'escalament. Theo-Agent

ha estat avaluat en un domini on s'han generat 10 regles d'estímul-resposta [108]. Però es desconeix el funcionament del sistema en entorns més complexos. Cal, doncs, desenvolupar com indica el seu autor a [108], tant mesures d'utilitat com mecanismes d'indexació de les regles apreses. BOLERO, en aquest sentit, no té problemes d'escalament, com s'ha demostrat amb la seva avaluació experimental, però no disposa de cap mecanisme per determinar la utilitat de la incorporació de nous casos. Els sistemes d'aprenentatge presenten, en general, el problema d'equilibrar els coneixements (plans, etc.) apresos amb la càrrega que representa el seu tractament en el sistema. En la secció 6.4.2 s'explica un sistema, PRODIGY/EBL [106], que per primera vegada utilitza mesures d'utilitat de les regles que aprèn.

6.4 Adquisició de plans

Com hem apuntat a la secció 1.3, l'adquisició de coneixements representa el coll d'ampolla en la construcció de sistemes basats en coneixements. El desenvolupament de mecanismes complexos de raonament que necessiten gran quantitat de coneixements (de domini, estratègics, etc.) requereix un esforç en aquesta àrea de la Intel·ligència Artificial. En aquest sentit, és important desenvolupar mecanismes d'aprenentatge automàtic ja que disminueixen l'esforç tant de l'expert com de l'enginyer de coneixements. Els mètodes d'aprenentatge automàtic, però, requereixen a la mateixa vegada una determinada quantitat de coneixements. Per exemple, BOLERO necessita d'una xarxa de coneixements de domini per aprendre plans. Els mètodes d'aprenentatge basats en explicacions pressuposen que la teoria del domini que utilitzen és completa i correcta. Però, com l'adquireixen? El sistema OCCAM [119], per exemple, és un sistema que integra un mètode d'EBL amb altres mètodes empírics per tal d'adquirir els coneixements necessaris per dur a terme l'aprenentatge basat en explicacions. Els coneixements que necessiten els mètodes d'aprenentatge automàtic han de tenir-se en compte en el desenvolupament de SBC, perquè tenen també un cost en l'adquisició de coneixements.

A més a més del coneixement que requereixen els mètodes d'aprenentatge automàtic és important conèixer com s'avalua el coneixement que s'aprèn. L'avaluació és l'única garantia que el comportament del sistema esdevindrà correcte. A BOLERO hem definit un patró d'avaluació basat en l'expert-mestre i cinc experts més (veure secció 2.4). ODYSSEUS [169] utilitza diferents patrons d'avaluació:

1. externs

- (a) la solució del problema
- (b) els passos que condueixen a la solució del problema
- (c) el *feedback* de l'entorn
- (d) l'oracle

2. interns

- (a) eficiència
- (b) correctesa

Els mètodes basats en explicacions, en assumir que la teoria del domini que utilitzen és correcta i completa, no tenen problemes d'avaluació: totes les instàncies positives poden demostrar-se com a instàncies d'un concepte de la teoria. El que ocorre realment amb els mètodes d'EBL, és que més que aprendre conceptes nous, s'està reescriuint la teoria del domini

```

(a) Rule PROPOSE-TRIGGERING-GENERAL-QUESTIONS
    IF (empty (differential))
      (in-class ?action general-questions)
      (not-empty (pottentially triggered ?action))
    THEN (propose ?actions do-history-and-exam)

(b) Rule FILTER-EXECUTED-ACTIONS
    IF (is (executed? ?action) yes)
    THEN (filter ?action)

(c) Rule SELECT-CHEAP-CONFIRMING-EVIDENCE
    IF (in gather-evidence-for-differential (current-goals))
      (in ?hypothesis (differential))
      (in ?action (pottentially-confirmed-by ?hypothesis))
    THEN (select ?action)

```

Figura 6.10: Exemples de regles d'estratègies en ASK. (de [63]).

de manera que resulti operacional. Tanmateix, l'avaluació dels coneixements apresos és, en general, un problema pendent en els mètodes d'aprenentatge automàtic la majoria dels quals opten per realitzar la validació de forma externa (i.e. consultar a un expert).

Entre els mètodes d'adquisició de plans, que hi ha relativament pocs comparats amb els mètodes d'adquisició de coneixements de domini, hem destacat en aquesta secció ASK [63], PRODIGY/EBL [106] i SAGE [93].

6.4.1 ASK

Descripció i funcionament

ASK (Acquiring Strategic Knowledge [63]) és una eina per a l'adquisició de coneixements estratègics per a aplicacions desenvolupades en l'arquitectura MU [43]. MU és una arquitectura pensada per a sistemes experts en medicina (diagnosis), on el coneixement és incert i incomplet.

Els coneixements estratègics en MU estan representats per regles d'estratègia on les premisses són conjuncions de característiques de control i les conclusions són accions. Les característiques de control són abstraccions de l'estat del sistema: funcions, atributs i relacions entre els objectes dels coneixements del domini. Es distingeixen tres tipus de regles d'estratègia: de focalització, de filtrat i de selecció. Les regles de focalització (figura 6.10a) examinen els objectius i l'estat actual del procés de resolució d'un problema i proposen accions per consolidar-los. Les regles de filtrat (figura 6.10b) expressen les restriccions de l'aplicabilitat de les accions, és a dir, eliminen les accions no-aplicables en un moment donat. Les regles de selecció (figura 6.10c) seleccionen un subconjunt d'accions per ser executades.

Les regles d'estratègia s'activen tan bon punt com alguna característica de control canvia. Primer es tracten les regles de focalització, que proporcionen un conjunt d'accions possibles; en segon lloc les regles de filtrat eliminen d'aquest conjunt les regles no aplicables, i finalment les regles de selecció proposen un conjunt d'accions, ordenades per preferència. Es continua la resolució del problema amb el conjunt d'accions seleccionat.

Adquisició de coneixements estratègics

L'adquisició de coneixements es basa a consultar a l'expert cada canvi d'estratègia que es produeix durant la resolució d'un problema. Això és, les regles d'estratègia proporcionen un conjunt d'accions amb les quals seguir la resolució d'un problema d'acord amb el darrer canvi d'una característica de control. Aleshores ASK mostra el conjunt d'accions seleccionat a l'expert, qui decideix si les accions són correctes o no. Si el conjunt d'accions seleccionat és correcte, es continua la resolució del problema. En cas contrari, i.e. les accions no són acceptades per l'expert, ASK inicia un diàleg amb l'expert per tal d'adquirir els coneixements necessaris per produir el conjunt d'accions adients a l'estat actual del sistema. El procés d'adquisició de coneixements també pot iniciar-se per la impossibilitat de suggerir més accions, no havent-se consolidat encara el objectius.

El primer pas en el procés d'adquisició de coneixements estratègics consisteix a demanar a l'expert el conjunt d'accions que haurien de ser escollides (exemples positius) i el conjunt d'accions que no s'haurien d'haver seleccionat (exemples negatius). Després, amb tots dos conjunt d'accions, ASK executa un procés d'assignació de crèdit per tal de determinar quines regles d'estratègia han causat la selecció incorrecta d'accions, i quins són els objectes dels coneixements del domini rellevants al procés de decisió (característiques de control). A continuació, ASK continua el diàleg amb l'expert per tal de verificar si amb les característiques de control o regles seleccionades és capaç de justificar els exemples positius i distingir-los dels negatius. Quan ASK no pot fer aquesta distinció, demana a l'expert nous atributs de control tals que siguin capaços d'establir aquesta distinció. A partir de les justificacions, el sistema genera una nova regla d'estratègia construint les regles tan generals com sigui possible. Finalment, s'executen de nou les regles d'estratègia sense que l'estat del sistema hagi estat modificat, i es mostra a l'usuari el nou conjunt d'accions seleccionades. Si l'expert accepta el nou conjunt d'accions, ASK torna el control al mecanisme de resolució de problemes. Altrament, s'inicia de nou el cicle d'adquisició de coneixements.

Discussió

ASK és una eina híbrida d'adquisició de coneixements, ja que és una eina d'elicitació de coneixements, que requereix la intervenció d'un expert (i/o enginyer de coneixements) i en la qual els mètodes d'aprenentatge automàtic també hi són presents. Concretament:

- l'enginyer de coneixements col·labora en la definició de característiques de control
- com a eina d'elicitació de coneixements, ASK utilitza un model de coneixements estratègics (regles d'estratègia) per a l'adquisició de coneixements estratègics en un domini específic.
- en l'adquisició de coneixements s'utilitzen mètodes d'aprenentatge a partir d'exemples positius i negatius per aprendre noves regles.

La utilitat d'ASK s'ha demostrat en l'adquisició de coneixements estratègics per a una aplicació al diagnostic de dolors de pit (MUM [63]). Els coneixements estratègics de MUM van ser implementats primerament per un enginyer dels coneixements en LISP, i amb ASK s'han tornat a readquirir les estratègies per avaluar el mètode d'elicitació de coneixements.

En l'avaluació s'ha fet palès que ASK no resulta molt eficient si s'han d'adquirir coneixements estratègics des de zero (sense un conjunt d'estratègies inicials). És a dir, ASK és bo per adaptar coneixements estratègics generals a un domini concret. Per aquesta raó, cal fornir ASK amb un conjunt de regles d'estratègia per defecte, de caràcter molt general

per a la diagnosi mèdica (com per exemple: “quan s’ha de cerca una hipòtesi, preguntar pels símptomes”, “executar cada acció només una vegada”, “concentrar-se en dades que seleccionin hipòtesis”, etc.). Aquestes estratègies bàsiques han de ser proporcionades per l’enginyer de coneixements. BOLERO no requereix coneixements previs sobre plans, i.e. BOLERO aprèn des de zero. A més a més, BOLERO aprèn tant de l’èxit com del fracàs dels plans que genera, mentre que ASK només aprèn del fracàs.

Els coneixements d’estratègies que s’obtenen mitjançant ASK en l’arquitectura MU permet mostrar a qualsevol SBC desenvolupat en MU un comportament reactiu. És a dir, tan bon punt es disposa de nova informació (característica de control), l’estat actual del sistema pot activar regles d’estratègia que suggereixin accions a executar. Les accions sempre estan suportades per les característiques de control i no per l’efecte d’accions prèvies.

6.4.2 PRODIGY/EBL

Descripció i funcionament

Steven Minton ha desenvolupat un sistema d’aprenentatge basat en explicacions (PRODIGY/EBL) per adquirir coneixements de control en l’arquitectura PRODIGY [32], ja introduïda en la secció 6.2.2. La importància de PRODIGY/EBL no radica tant en l’aplicació d’una tècnica de EBL, sinó en les mesures que defineix per tal de garantir que les regles apreses són realment efectives i milloren la resolució de problemes del sistema (*speed-up learning*). El problema de la utilitat del coneixement en els mètodes d’aprenentatge és tant o més greu que el problema dels sorolls [33], i PRODIGY/EBL és el primer sistema que incorpora un tractament a aquest problema.

Dins de l’arquitectura PRODIGY cal detallar, a més del que ja ha estat dit en la secció 6.2.2, el formalisme de representació de les regles de control que determinen l’aprenentatge en PRODIGY/EBL. Les regles de control estan constituïdes per una condició i una acció sobre els operadors o regles d’inferència. Les condicions es formen a partir de metaconceptes com CURRENT-GOAL, CURRENT-OPERATOR, etc. Les accions poden ser de tres tipus: de selecció, de filtrat o de preferència d’un operador o regla d’inferència envers un conjunt d’operadors o regles d’inferència. Existeixen diferents tipus de regles de control: de selecció, de filtrat i de preferència. En el procés de resolució d’un problema les regles de control de tipus de selecció escullen un conjunt de candidats (nodes, objectius, etc) sobre els quals continuar la resolució del problema. Les regles de control de tipus filtrat eliminen candidats, i finalment les regles de preferència escullen un candidat per continuar la validació dels objectius. En PRODIGY/EBL s’utilitzen les traces que genera el resolador general de problemes de PRODIGY (RGP) per aprendre noves regles de control. RGP es diferencia de NoLimit (presentat en la secció 6.2.2) en el fet que és un planificador lineal i no interacciona amb SMART en la resolució de problemes⁴.

Aprenentatge de regles de control

El mètode d’aprenentatge que utilitza Minton és una versió de l’EBL que anomena Especialització Basada en Explicacions (*Explanation Based Specialization*, EBS). Recordem que l’EBL consisteix a provar que un exemple és una instància d’un concepte mitjançant l’elaboració d’una explicació. L’explicació permet formular una descripció operacionalitzable

⁴La implementació de NoLimit ha estat posterior a PRODIGY/EBL.

del concepte. La particularitat de l'EBS és que es parteix d'un conjunt d'esquemes que proporcionen la base per generar explicacions.

El primer pas en l'aprenentatge de regles de control de PRODIGY/EBL consisteix a seleccionar un exemple de concepte per aprendre d'entre els quatre possibles "conceptes objectius" (*target concepts*): SUCCEEDS (reeiximent), FAILS (fracàs), SOLE-ALTERNATIVE (alternativa única) o GOAL-INTERFERENCE (interferència d'objectius). Els exemples es reconeixen dins de la traça que el RGP genera mitjançant informació continguda en els conceptes objectius (*example recognizers*, figura 6.11) i un conjunt d'heurístiques que indiquen l'interès dels exemples. Després, per a cada exemple s'obté una explicació que justifica per què és un exemple del concepte objectiu. Per construir aquesta explicació PRODIGY/EBL té una teoria del domini i de l'arquitectura basada en axiomes que anomena esquemes de demostració (figura 6.12). Cada concepte objectiu té un esquema associat i, a més a més, PRODIGY/EBL disposa d'altres esquemes de demostració que fan referència al domini particular en què es du a terme la resolució de problemes. Una explicació no és més que el desenvolupament d'un esquema de demostració del concepte objectiu fent ús d'un exemple del concepte i dels esquemes de demostració del domini. L'explicació resultant s'intenta simplificar i combinar amb explicacions obtingudes en altres exemples en un procés que s'anomena de compressió, de manera que l'explicació resultant sigui com més senzilla possible. A continuació l'explicació compresa es transforma en una regla de control: les explicacions procedents del concepte objectiu SUCCEEDS o GOAL-INTERFERENCE es transformen en regles de preferència, les de l'objectiu FAILS en regles de filtrat, i les de SOLE-ALTERNATIVE en regles de selecció. Finalment, PRODIGY/EBL avalua la utilitat de la regla apresada.

La mesura d'utilitat es basa en el cost acumulatiu de validar la satisfacció de la premissa de la regla de control versus els beneficis que produeix la regla en la cerca de solucions. Quan PRODIGY/EBL aprèn una regla nova fa una estimació del seu cost a partir de l'exemple, i a continuació es valida l'estimació de la utilització de la regla en la resolució de problemes nous durant un període de temps. L'avaluació del cost de les regles apresades és, doncs, empíric. Només les regles que tenen una utilitat alta s'emmagatzemen.

PRODIGY/EBL aprèn tant de les solucions, com dels fracassos i de la interacció d'objectius. Amb les mesures d'utilitat s'ha detectat, però, que realment l'aprenentatge menys útil és el que s'esdevé a partir del reeiximent dels plans.

```

Target concept: (OPERATOR-FAILS op goal node)
Example Recognizer: (PICK-OUT-OP-FAILURE node)
Example Selection Heuristics: (...)
Rule Construction Template: If (and (CURRENT-NODE node)
                                (CURRENT-GOAL node goal)
                                (CANDIDATE-OPERATOR op node )
                                (OPERATOR-FAILS op goal node))
                            THEN (REJECT OPERATOR op)

```

Figura 6.11: Representació d'un concepte objectiu a aprendre a PRODIGY (de [106]).

- (a): An operator succeeds if it directly solves the goal
 (OPERATOR-SUCCEEDS *op goal node*)
 if (AND (MATCHES-EFFECT *goal op*)
 (APPLICABLE *op node*))
- (b): POLISH is applicable if its preconditions are satisfied.
 (APPLICABLE *op node*)
 if (AND (MATCHES-OP *op (POLISH obj time)*)
 (KNOWN *node (AND (IS-OBJECT obj)*
 (OR (SHAPE *obj RECTANGULAR*)
 (CLAMPABLE *obj POLISHER*))
 (AVAILABLE *obj time*)
 (IDLE POLISHER *time*)))

Figura 6.12: Representació d'esquemes de demostració a PRODIGY (de [106]). (a) Esquema de demostració del concepte objectiu OPERATOR-SUCCEEDS, que depèn de l'arquitectura de PRODIGY. (b) Esquema de demostració de l'aplicabilitat de l'operador POLISH d'acord amb la teoria del domini.

Discussió

L'aportació més significativa de PRODIGY/EBL ha estat l'aplicació de l'aprenentatge basat en explicacions com a tècnica per a l'optimització del procés de resolució de problemes. L'aprenentatge de regles de control no proporciona per si mateix un millorament en la resolució de problemes, com ho demostren les anàlisis experimentals fetes per Minton amb l'EBS, sinó que és l'adquisició de regles que realment són útils les que proporcionen la millora. La utilitat de les regles apreses s'ha de tenir molt present perquè la incorporació d'un nombre considerable de regles que tenen un cost d'instanciació de la premissa molt alt, que s'apliquen poc o que produeixen un benefici molt baix perjudiquen l'eficiència en la resolució de problemes, més que afavorir-la.

A BOLERO no s'ha desenvolupat cap mecanisme per avaluar la conveniència d'incorporar un cas en memòria, però seria molt interessat construir-ne un per tal de millorar el temps de resposta del sistema. Com ja s'ha indicat en la secció 5.3.3, la quantitat de casos incorporats en memòria determina el nombre d'indexacions, i en conseqüència el temps necessari per resoldre problemes. La utilitat d'incorporar un cas *C* es pot avaluar analitzat:

- si existeix un pla anàleg en memòria que només es diferenciï del pla de *C* en l'ordre de les accions, i
- si aquesta diferència en l'ordre és rellevant.

Només si no existeix un pla anàleg o l'ordre de les accions és rellevant, el cas s'incorpora. Aquest mecanisme d'avaluar la utilitat dels casos és un dels treballs a desenvolupar en el futur (secció 7).

PRODIGY/EBL és sensitiu a l'ordre i al tipus dels exemples d'aprenentatge. En aquest sentit, Minton indica que el més indicatiu a PRODIGY/EBL és proporcionar inicialment exemples senzills i anar-los complicant incrementalment. A BOLERO hem observat també una dependència de l'ordre dels exemples en els resultats de les accions (veure secció 5.4.4), i si bé es coneix que hi ha casos més complexos que altres, no s'ha fet aquesta distinció en l'avaluació experimental del sistema.

6.4.3 SAGE

Descripció i funcionament

SAGE (*Strategy Acquisition Governed by Experimentation* [93, 92]) és un sistema que aprèn heurístiques específiques a un domini a partir de mètodes dèbils de cerca de solucions (*depth-first search*, *mean-ends analysis*). Les heurístiques actuen com a estratègia de control de cerca en l'espai d'estats, d'una manera semblant al sistema LEX [111]⁵. Les estratègies apreses permeten focalitzar la cerca i eviten la marxa enrera (*backtracking*) durant la resolució de problemes.

SAGE té els components següents: un component de cerca de solucions o resolador de problemes, un component d'assignació de crèdits, i un component de modificació del comportament del sistema. El resolador de problemes proporciona experiències a SAGE a partir de les quals aprendre estratègies. El component d'assignació de crèdits, i el component de modificació del comportament són els responsables de l'aprenentatge d'estratègies.

SAGE representa un problema mitjançant un estat inicial, un estat objectiu i un espai d'estats determinants per un conjunt d'operadors. Els operadors són utilitzats per resoldre problemes mitjançant una cerca en profunditat (*depth-first search*) o mitjançant l'anàlisi dels medis en relació als objectius (*means-ends analysis*). El resolador de problemes té també accés a les regles d'estratègia que proposen els operadors adients a les condicions de l'estat actual. Quan el resolador de problemes aplica un operador, genera una traça en la memòria on queda registrat l'operador i els arguments amb què ha estat instanciat (per exemple: (*moviment-1 era moviment disc-1 de estaca-A a estaca-C*)).

TOH

Si hi ha un *disc* en *estaca-actual* en *estat-actual*,
 i hi ha algun *altra-estaca* diferent d'*estaca-actual*,
 i en *estat-actual* no existeix un *altre-disc* en *estaca-actual* més petit que *disc*,
 i en *estat-actual* no existeix un *tercer-disc* en *altra-estaca* més petit que *disc*,

Llavors considerar moviment de *disc* d'*estaca-actual* a *altra-estaca*.

Figura 6.13: Estratègia de caràcter general per resoldre el problema de les torres de Hanoi (de [92]).

Aprenentatge d'estratègies a SAGE

Un cop SAGE ha resolt un problema mitjançant els mètodes dèbils de cerca, s'activa el mode d'aprenentatge. En primer lloc el component d'assignament de crèdits distingeix els comportaments desitjables del sistema dels indesitjables i determina els elements del sistema responsables del comportament. A continuació el component de modificació de SAGE modifica regles existents o aprèn noves regles d'estratègia. Així, inicialment SAGE només disposa d'una regla d'estratègia (o un grup petit de regles) de caràcter general, com la que mostra la figura 6.13, i finalment SAGE disposa de regles d'estratègia molt específiques (figura 6.15).

⁵Langley anomena de diferents maneres aquestes heurístiques: regles de propostes (*rule proposes*), produccions, etc. Per no confondre les heurístiques del control de l'espai de cerca amb les heurístiques que fa servir SAGE en l'algorisme d'aprenentatge, hem convingut d'anomenar les primeres estratègies o regles d'estratègies.

L'assignació de crèdits es basa en el coneixement de la solució correcta (disponible, car s'ha resolt el problema). Qualsevol estat que s'allunyi de la solució es marca com a estat indesitjable; en cas contrari, l'estat es marca com a estat desitjable. Es defineixen diferents errors en el curs de la resolució d'un problema: moviments circulars, camins de solució més llargs, punts morts (*dead-ends*), impossibilitat de continuar i estats il·legals. A més, existeixen diferents heurístiques que s'utilitzen per identificar els operadors responsables d'un error que ha dut el sistema a un estat indesitjable. Per exemple, la figura 6.14 mostra una heurística per detectar operadors que condueixen a punts morts. Les traces dels operadors que condueixen des d'un estat correcte s a un de no desitjable $\bar{s}$ esdevenen instàncies negatives de la regla d'estratègia R resultat de la qual s'ha aplicat l'operador. Les traces dels operadors que des de l'estat s no han conduït a $\bar{s}$ són instàncies positives d'una altra regla d'estratègia R' .

El component de modificació de SAGE analitza R i R' . Si $R = R'$ (i.e. els exemples positius i negatius són instàncies de la mateixa regla), aleshores s'engega un mecanisme de discriminació amb la instància positiva i la instància negativa a partir del qual es crea una regla d'estratègia nova. Si les regles són diferents (i.e. $R \neq R'$), aleshores es disminuirà la importància de R i s'augmentarà la importància de R' .

DEAD-END

Si	<i>estat-actual</i> és l'estat actual, i <i>moviment</i> condueix de <i>estat-ant</i> a <i>estat-actual</i> , i no existeix cap moviment possible des de <i>estat-actual</i> , que no hagi estat ja realitzat,
Llavors	fer esdevenir <i>estat-ant</i> l'estat actual i marcar <i>moviment</i> com a indesitjable.

Figura 6.14: Heurística per a detectar punts morts en la resolució de problemes (de [92]).

La generació d'una nova estratègia es basa, doncs, en un procés de discriminació que consisteix a diferenciar el context (estat de la memòria i els valors de les variables) en què s'ha donat una instanciació incorrecta de la regla d'estratègia (context de rebuig) i el context en què s'ha instanciat correctament (context de selecció). A partir de les diferències entre ambdós contextos es construeix una variant de la regla d'estratègia R . Els elements que apareixen en el context de selecció i no apareixen en el context de rebuig són utilitzats per construir condicions addicionals que, afegides a la regla, resulten en una nova variació de R . Per contra, els elements que apareixen en el context de rebuig i que no apareixen en el context de selecció intervien com a condicions negades en la nova variació de la regla. Per exemple, la figura 6.15 és una variació de la regla d'estratègia que es mostra a la figura 6.13. Es poden construir variacions a una regla per conjuncions de condicions i disjuncions. La nova regla té assignat un grau d'importància (*strength*) inferior al que té la regla de la qual es deriva (i.e. R). D'aquesta manera primer s'aplica la regla més general R i després la derivada R_d , si no és que la importància augmenta per processos d'aprenentatge posteriors. Si la importància augmenta, vol dir que la regla és útil. Només les regles que tenen assignat un grau d'importància superior a un llindar són utilitzades en el procés de resolució de problemes.

Després d'aprendre les noves regles, SAGE executa de nou el problema. En la nova execució es podran detectar nous errors. I així successivament fins que SAGE no detecta cap error en la resolució d'un problema.

TOH-1

Si hi ha un *disc* en *estaca-actual* en *estat-actual*,
 i hi ha algun *altra-estaca* diferent d'*estaca-actual*,
 i en *estat-actual* no existeix un *altre-disc* en *estaca-actual* més petit que *disc*,
 i en *estat-actual* no existeix un *tercer-disc* en *altra-estaca* més petit que *disc*,
 i no és el cas que
 moviment-ant condueix de *estat-ant* a *estat-actual*,
 i *moviment-ant* era un moviment de *disc* d'*altra-estaca* a *estaca-actual*
 Llavors considerar moviment de *disc* d'*estaca-actual* a *altra-estaca*.

Figura 6.15: Estratègia per resoldre el problema de les torres de Hanoi, on s'ha après a evitar repetir un estat ja visitat (de [92]).

Discussió

SAGE ha estat aplicat en diversos dominis, on ha demostrat la seva generalitat. El concepte que té l'autor sobre l'escalament (*scaling up*) del seu mètode és, però, discutible. En el domini de les torres de Hanoi, SAGE és capaç d'utilitzar les heurístiques apreses en un problema de 3 discs, per resoldre el problema de 4 discs. Encara que les aplicacions de laboratori permeten avaluar els mètodes d'aprenentatge d'una manera exhaustiva i diferent a la que permeten les aplicacions reals ([45]), és molt arriscat deduir que a partir del bon funcionament de SAGE en aplicacions de laboratori el sistema no presenta problemes d'escalament. Precisament, el problema de l'escalament resideix en la transmissió d'un mètode d'aprenentatge vàlid en aplicacions de laboratori a aplicacions del món real. Si en un problema d'ordenació de 4 blocs per mides SAGE aprèn 67 regles, de les quals només una esdevé útil, el nombre de regles que pugui aprendre en una aplicació del món real pot esdevenir força important. A BOLERO s'ha afrontat la construcció d'una aplicació real des d'un inici, evitant el problema de l'escalament.

És interessant observar que la comparació entre contextos (context de selecció i context de rebuig) que es du a terme en el procés de discriminació de SAGE és en essència la comparació que té lloc a BOLERO per descobrir les característiques diferenciadores d'un cas recuperat erròniament de memòria i el cas correcte, en l'aprenentatge de la importància estratègica dels índexs. Un punt feble que presenta SAGE respecte de BOLERO és la impossibilitat d'incloure condicions sobre l'estat objectiu en les regles d'estratègia. L'inclusió de l'estat objectiu en les regles permet aprendre sobre les condicions de finalització d'un problema. Aprendre sobre les condicions de terminació és important en la planificació com s'ha demostrat en BOLERO. Per acabar, és interessant remarcar que SAGE aprèn d'errors, mentre que BOLERO aprèn tant del reeiximent com del fracàs dels plans que genera.

6.5 Conclusions

En aquest capítol hem visitat diferents sistemes basats en casos, planificadors basats en casos, sistemes amb comportaments reactius, i sistemes que faciliten l'adquisició de plans⁶. En cada secció s'han valorat els avantatges, similituds i problemes que presenta BOLERO envers

⁶La comparació de BOLERO amb altres treballs podria haver estat més extensa si es considera sistemes de control, adaptatius i d'altres, però creiem que queden massa lluny de l'estructura (arquitectura) i de la finalitat de BOLERO.

altres sistemes, però no s'ha realitzat una comparació per determinar quin sistema és millor a partir dels resultats individuals de cadascun dels sistemes. La raó és que cada sistema ha estat avaluat diferentment: cada sistema s'ha aplicat a dominis diferents, s'ha avaluat amb diferents mesures (temps, utilitat, cost de memòria), etc. Per dur a terme una avaluació objectiva s'hauria d'aplicar els diferents mètodes sobre els mateixos dominis i prendre les mateixes mesures⁷.

No obstant això, la anàlisi dels treballs relacionats amb BOLERO ens permet formular les conclusions següents:

1. BOLERO, com a sistema basat en casos, presenta semblances i divergències entre els seus predecessors en referència als mètodes de recuperació, adaptació i aprenentatge.
2. La funció de similitud definida en BOLERO sembla estar en concordància amb les funcions d'altres autors en què la importància estratègica dels fets juga un paper fonamental.
3. El problema d'avaluar la utilitat d'incorporar un cas en memòria és un problema compartit entre tots els sistemes basats en casos, incloent BOLERO.
4. No hi ha cap planificador basat en casos provat experimentalment en una aplicació real com és el cas de BOLERO.
5. No hi ha sistemes que combinin la planificació reactiva i l'aprenentatge de plans, a excepció de Theo-Agent [19]. Theo-Agent, però, en aprendre regles del tipus estímulo-resposta presenta problemes en el seu funcionament a llarg termini.
6. Hi ha pocs sistemes que, com BOLERO, combinin l'aprenentatge a partir del reeiximent i del fracàs dels plans. La majoria dels sistemes recollits en aquest capítol aprenen només a partir del fracàs.
7. Els coneixements que fan possible l'aprenentatge d'estratègies són un cost a calcular tant en el desenvolupament de BOLERO com en el d'altres mètodes d'adquisició de coneixements.

Davant d'aquestes conclusions, podem dir que BOLERO té aportacions originals com són la combinació de la planificació reactiva i l'aprenentatge tant per observació com per l'experiència pròpia.

⁷La necessitat de formular una metodologia d'avaluació s'ha fet palesa en els darrers anys per tal d'avaluar els diferents mètodes d'aprenentatge automàtic ([77, 94, 95, 44, 45]), i en els sistemes basats en coneixements en general ([100, 104]), i avançar en la línia de recerca adequada.

Capítol 7

Contribucions i treball futur

En aquest capítol es realitza una anàlisi del treball realitzat en aquesta tesi per tal d'explicitar les contribucions principals del treball i els nous problemes que han sortit a la llum arran d'aquest treball i que cal resoldre en un futur. En la secció 7.1 s'avalua la recerca. Després, en la secció 7.2 es refereixen les contribucions principals del treball. Finalment, en la secció 7.3 es formulen noves línies de recerca.

7.1 Avaluació de la recerca

L'anàlisi del treball de recerca es pot realitzar seguint les anotacions suggerides per Cohen i Howe [44], en què distingeixen cinc etapes per avaluar un projecte de recerca: 1) justificació i viabilitat del sistema, 2) avaluació del mètode, 3) implementació, 4) experimentació i 5) avaluació dels resultats. A continuació es presenten cada etapa aplicada al desenvolupament de BOLERO.

Etapa 1: Justificació i viabilitat del sistema

Després d'un estudi preliminar de la planificació, els mètodes d'adquisició de coneixements, i els sistemes basats en coneixements en general, s'ha plantejat el desenvolupament d'un nou planificador, BOLERO, amb les característiques següents:

- planificació reactiva
- adquisició de coneixements estratègics mitjançant mètodes d'aprenentatge automàtic
- capaç de millorar l'eficiència d'un sistema basat en coneixements
- utilitzable en aplicacions reals

La planificació reactiva està dirigida a la construcció de plans i pensada per alternar-se amb l'execució dels plans, de manera que es reacciona als canvis de l'entorn. La planificació reactiva és necessària per manipular aplicacions reals on es presenten problemes d'incompletitud i incertesa en les dades.

Utilitzar tècniques d'aprenentatge automàtic facilita l'adquisició de coneixements. L'experiència de l'IIIA en el desenvolupament de sistemes basats en regles ens ha fet donar especial importància al tema de l'adquisició de coneixements. L'adquisició de coneixements ocupa una part molt important en el desenvolupament de sistemes donada la dificultat de l'expert a expressar els coneixements en formalismes específics, com ho són les regles i metaregles, i la interacció que es produeix entre regles i metaregles.

Dissenyar un planificador que millora l'eficiència d'un SBC permet mantenir independents el desenvolupament de dues parts diferenciades: els coneixements de domini i els coneixements estratègics. Així, BOLERO pot ser aplicat en principi a d'altres sistemes existents que presenten un rendiment deficient i millorar-los (*speed-up learning*).

Disenyar un sistema tenint sempre present que serà utilitzat en aplicacions reals evita el problema de l'escalament (*scaling up*). A més a més, en l'IIIA existeixen diferents aplicacions reals amb les quals l'experimentació i, consegüentment, l'avaluació del sistema ha estat viable.

Etapa 2: Avaluació del mètode

El mètode escollit ha estat la planificació basada en casos (PBC). La PBC ja ha estat utilitzada en altres sistemes, dels quals s'han derivat parts del mètode. Però la particularitat de BOLERO és la de ser un planificador reactiu, on la planificació i l'execució de plans s'alternen. A més a més, BOLERO no és un planificador aïllat, sinó que millora l'eficiència d'un SBC existent.

Utilitat. La PBC com a tècnica d'adquisició de coneixements facilita la interacció persona-màquina, ja que la comunicació es pot realitzar mitjançant casos concrets i no trossos parcials de coneixements, com ho són les regles. La PBC com a mecanisme de resolució de problemes permet aprendre de la pròpia experiència del sistema (així, un sistema no-expert o novell pot esdevenir expert). La PBC permet treballar en un domini de problemes de difícil estructuració i representació, característiques que presenten les aplicacions reals.

Àmbit d'aplicació. BOLERO pretén ser un planificador adaptable a qualsevol SBC sense coneixements de planificació o planificació deficient i millorar-ne l'eficiència. BOLERO pot ser també un instrument de comparació entre l'adquisició automàtica de coneixements i l'adquisició manual mitjançant un enginyer de coneixements. L'avantatge de BOLERO rau no només en la rapidesa i l'estalvi de l'esforç de l'enginyer de coneixements, sinó que evita la detecció i correcció d'interaccions entre diferents parts del coneixements (i.e. treball d'assaig i error).

Assumpcions. S'assumeix que es disposa d'un expert-mestre per aprendre plans per observació i un conjunt d'experts addicionals per configurar el patró d'avaluació. A més a més, el fet de treballar amb una aplicació real presenta una limitació, que és la disponibilitat de casos per a l'aprenentatge. La generació automàtica de casos no és una tasca fàcil quan es manipulen aplicacions reals, car hi ha dades contradictòries, valors excloents, etc. que fan impossible la generació automàtica. Per això se suposa que es disposa d'un conjunt significants de casos que permeten l'aprenentatge.

Funcionament. En el funcionament del sistema es distingeixen dues parts: l'adquisició de coneixements i la construcció de plans. En la primera (adquisició) el temps de resposta dels sistema no és massa rellevant, però sí en la segona. En la construcció de plans s'ha d'evitar incloure accions no necessàries (sobreres).

Limitacions. Per tal de poder definir un mètode d'avaluació de plans vàlid, es restringeix el tipus d'aplicacions a desenvolupar en BOLERO. BOLERO pot ser aplicat amb èxit en

aplicacions de diagnosi mèdica. En aquest tipus d'aplicació la informació disponible per a resoldre un problema és incerta i incompleta i es requereix d'una planificació reactiva.

Etapla 3: Implementació.

La implementació de BOLERO ha estat realitzada en una SUN, utilitzant Common Loops Object System (CLOS). El comportament del sistema pot ser avaluat, car els problemes que soluciona esdevenen casos. Els casos es poden presentar a un expert, el qual els pot jutjar.

Etapla 4: Experimentació.

S'ha realitzat una avaluació experimental del sistema. Els experiments s'han dissenyat seguint l'esquema conegut pel nom de *leave-one-out*. Els experiments s'han realitzat amb casos reals, casos amb dades incompletes i incertes i que presenten sorolls en les dades (amb una distribució desconeguda). El patró d'avaluació ha estat multiexpert, encara que la presència d'un mestre ha esbiaixat els resultats. La independència del sistema respecte d'un domini ha estat provada quan s'ha utilitzat en dues aplicacions: RENOIR i PNEUMONIA (per bé que només la segona s'ha experimentat exhaustivament). Els experiments han estat realitzats en diferents configuracions del sistema per tal d'avaluar l'impacte dels diferents components, i s'han repetit diverses vegades en cada configuració de manera que els resultats no depenen ni de l'ordre ni de la complexitat dels casos.

Etapla 5: Avaluació dels resultats.

Els resultats ens demostren que el sistema:

- acara el problema de l'adquisició de coneixements estratègics satisfactòriament
- afronta el problema de la informació incompleta i incerta que presenten les aplicacions reals
- millora el procés de resolució d'un sistema expert basat en regles mitjançant la integració del sistema i el planificador en una arquitectura metanivell
- treballa amb aplicacions reals

L'avaluació dels resultats també fa palès que:

- l'avaluació multiexpert determina l'aprenentatge per experiència
- la utilització d'aplicacions mèdiques condiciona l'aparició d'un biaix conservador que limita les capacitats del sistema
- la importància de les característiques de les dades en l'eficiència del sistema: rellevància, sorolls, quantitat, etc.

7.2 Contribucions

De l'avaluació del treball de recerca es poden formular les contribucions principals següents:

- Un mètode per a l'adquisició automàtica de coneixements de planificació basat en casos on un mestre proporciona els exemples d'aprenentatge (casos). Aquest mètode ajuda a alleugerir el problema de l'adquisició de coneixements en la construcció d'un SBC. El mètode d'aprenentatge per observació permet desenvolupar de forma independent els dos nivells de coneixements que avui en dia presenta qualsevol SBC: coneixements estratègics i coneixements de domini.

- Un mètode d'aprenentatge per l'experiència pròpia on un planificador basat en casos aprèn del reeiximent i del fracàs dels plans que genera. Amb l'aprenentatge per experiència es pot evolucionar d'un comportament principiant, on només es disposa de coneixement teòric (de domini) per generar plans, a un sistema expert capaç de reusar plans d'experiències anteriors per solucionar nous problemes.
- Un planificador basat en casos on els plans es recuperen en funció de la informació disponible. Aquesta característica és molt important si tenim present que en el món real la informació coneguda sobre un problema és tot sovint incompleta i incerta.
- Una arquitectura metanivell on s'integren dos sistemes independents: un planificador basat en casos i un sistema basat en coneixements capaç de resoldre problemes per una cerca exhaustiva de totes les solucions. Com a conseqüència de la integració, es millora l'eficiència de la resolució de problemes del SBC. I atès que el SBC executa els plans del planificador basat en casos, el planificador és capaç d'avaluar els plans que genera i, consegüentment, aprendre de la seva experiència pròpia.
- Una arquitectura metanivell on la resolució de problemes es consolida mitjançant l'alternació de la planificació i l'execució de plans, amb capacitat de reacció davant dels canvis que es produeixen a l'entorn. La planificació reactiva és necessària en entorns reals, on la informació és incompleta i incerta. En entorns mèdics la incompletesa i incertesa de les dades es fa més palesa quan es necessita donar una solució urgent.
- Un mètode d'adquisició i generació de plans útils en aplicacions reals, on la qualitat i quantitat de la informació que es manipula fan de la planificació una tasca complexa. Per una banda, la quantitat de dades necessàries per dur a terme la resolució de problemes fan de l'adquisició de coneixements una tasca laboriosa. Per una altra banda, la qualitat de la informació requereix una planificació flexible en la qual la qualitat de la resposta del sistema no es vegi deteriorada per la manca d'informació.

7.3 Treballs futurs

El treball de recerca desenvolupat en aquesta tesi dona solució al problema de l'adquisició i generació de plans, però a la vegada l'avaluació experimental de BOLERO ha plantejat problemes nous. Així, per resoldre els nous problemes i, en conseqüència, millorar BOLERO, es plantegen els següents treballs futurs:

Definició d'una mesura d'utilitat dels casos. Durant tota l'avaluació de BOLERO s'ha posat de manifest la necessitat de definir una mesura d'utilitat dels casos, de manera que només s'incorporin en memòria casos que aportin un coneixement realment significatiu (i.e. que continguin plans significativament diferents).

Desenvolupament d'un mecanisme de compactació de nodes. Una manera de millorar el temps de resposta del sistema és reduir el nombre d'indexacions mitjançant la reducció del nombre de nodes. La reducció del nombre de nodes es pot realitzar mitjançant la identificació de cadenes de nodes amb només un únic descendent. La compactació de nodes requereix, però, estudis acurats que proposin mecanismes de descompactació necessaris per a la incorporació incremental de casos.

Estudi de mecanismes per distingir fets irrelevantes de fets rellevants. L'experiència de BOLERO ha mostrat que l'augment de fets perjudica tant en el temps de resposta del sistema (nombre d'indexacions) com amb la correctesa i precisió dels plans. Malgrat que la distinció entre fets rellevants i fets irrelevantes pot aconseguir-se, requereix un esforç molt alt per part de l'expert en l'etapa d'adquisició de coneixements previs a l'aprenentatge. Per tant, és convenient estudiar mecanismes que aprenguin la rellevància dels fets i eliminin progressivament de memòria fets irrelevantes.

Estudi i aplicació de noves mesures de similitud. Existeixen diferents mesures que poden ser aplicades en BOLERO a més a més de l'exposada en el capítol 4. Mesures basades en t-conormes, o mesures que s'avaluïn sobre valors lingüístics poden representar alternatives a la mesura de similitud definida en BOLERO.

Incorporació d'un mecanisme d'atenuació en la propagació d'activació. Una de les característiques de la propagació d'activació és la d'atenuar l'activació a mesura que s'allunya de l'origen. Això es pot incorporar a BOLERO, de manera que el fet origen influeixi amb més rellevància en les mesures de similitud que els fets activats a partir d'ell.

Incorporació de mètodes paral·lels. El mecanisme d'activació de la propagació és un mecanisme que permet el paral·lisme. El processament en paral·lel depèn en certa mesura de la màquina i llenguatge de programació sobre el qual s'ha desenvolupat BOLERO. Un estudi aprofundit sobre les possibilitats de desenvolupar l'activació de la propagació en paral·lel pot repercutir en una millora substancial en el temps de resposta del sistema respecte a la resolució de problemes.

Experimentar amb diferents funcions per corregir la importància estratègica dels fets. Hem utilitzat una funció de combinació d'evidències basada en t-normes i t-conormes per augmentar o disminuir la importància estratègica de generalitzacions. Les t-normes i t-conormes poden prendre moltes formes, i l'experimentació amb un conjunt d'elles pot proporcionar nous resultats en l'aprenentatge per experiència.



A més a més d'intentar resoldre els problemes plantejats, l'experiència en el desenvolupament de BOLERO ens permet també formular noves línies de recerca per seguir en el futur i desenvolupar sistemes millors. Entre aquestes línies de recerca destaquem les següents:

Desenvolupament de mètodes d'aprenentatge de fracassos. Com ja ha estat dit en el seu moment, és diferent aprendre a partir del fracàs (o plans imperfectes) que aprendre fracassos. L'aprenentatge de fracassos permet recordar experiències d'errors anteriors i reconèixer-les en problemes nous. L'aprenentatge de fracassos pot comportar l'estudi de mecanismes de reparacions de plans.

Estudi i desenvolupament d'aprenentatge no-supervisat. En el disseny de BOLERO es va plantejar el desenvolupament de mètodes d'aprenentatge no-supervisat. És a dir, es tractaria d'analitzar les traces de resolució de problemes d'un SBC i deduir plans que puguin

millorar l'eficiència mitjançant, per exemple, el descobriment d'objectius fracassat. Aquests mètodes, però, requereixen mecanismes d'avaluació estrictes i precisos que determinin la validesa dels coneixements que s'aprenen. Així, mentre que en robòtica és relativament senzill definir si un pla és millor que un altre sobre la base de criteris com *el pla més curt*, *la utilització reduïda de recursos*, etc., aquests criteris no són vàlids per a la diagnosi mèdica. L'estudi de mecanismes d'aprenentatge no-supervisat obra línies de recerca poc explorades i molt àmplies.

Estudi de l'aplicabilitat de BOLERO en entorns diferents de la diagnosi mèdica.

L'aplicació de BOLERO en dues aplicacions ens permet formular la hipòtesi d'independència de domini de BOLERO. Però existeixen un altre tipus d'aplicacions, com la trajectòria de robots, la terapèutica, on moltes de les assumpcions realitzades en BOLERO no són vàlides. És, doncs, necessari realitzar un estudi de l'extensibilitat de l'aproximació de BOLERO a d'altres tipus de dominis. En aquest sentit, també seria interessant estudiar BOLERO en aplicacions de laboratori ben conegudes i que permeten avaluar els sistema en condicions extremes.

Apèndix A

Símbols utilitzats en la representació de coneixements de BOLERO

A.1 Fets

F : conjunt de fets

$f_i = \langle id_i, v_i \rangle$: fet monovaluat

$f_i = \langle id_i, V_i \rangle$: fet multivaluat

id_i : identificador

v_i : valor; V_i : conjunt de valors

F_a : fets resultats d'accions; $F_a \subset F$

F_f : fets resultats d'accions finals; $F_f \subset F_a$

A.2 Situacions

s_j^i : situació; $s_j^i \in \mathcal{P}(F)$

$s_{n^i}^i \in C^i$: situació del darrer episodi del cas C^i

A.3 Accions

A :conjunt d'accions

a_i : acció

$result(a_i) \subset \mathcal{P}(F)$: resultat d'una acció

accions executades & accions completades

accions interrompudes & accions continuades

accions intermèdies & accions finals

$\hat{A}$:conjunt d'accions executades

A.4 Plans

Γ conjunt de plans; $\Gamma = \mathcal{P}(A)$

$p_j^i = [a_0^i, a_1^i, \dots, a_j^i]$: pla immediat

$\bar{p}_j^i$: pla executat

$\bar{p}_j^i$: pla executat/immediat

$\bar{p}_j^i \in C^i$: pla parcial del cas C^i

$\bar{p}_{n^i}^i \in C^i$: pla final del cas C^i

$\bar{p}^i \in N^i$: pla parcial del node N^i

$\bar{p}^i \in N_t^i$: pla final del node terminal N_t^i

A.5 Episodis

$\delta_j^i = \langle s_j^i, \bar{p}_j^i \rangle$: episodi

$\delta_{n^i}^i \in C^i$: darrer episodi del cas C^i

A.6 Traces

Δ : conjunt de traces

$T(\pi^i) = [\delta_0^i, \delta_1^i, \dots, \delta_{n^i}^i]$: traça del problema π^i

A.7 Casos

$C^i = R(T(\pi^i))$: cas

$S_{C^i} = \left(\bigcup_{a_k \in \bar{p}_{n^i}^i} \text{result}(a_k) \right) \cap F_f$: solució del domini del cas C^i

$\bar{p}_{n^i}^i \in C^i$: solució del cas C^i des del punt de vista de la planificació

A.8 Nodes

$N^i = \langle v^i, \bar{p}^i, \tau^i \rangle$: node

$N_t^i = \langle v^i, \bar{p}^i, \tau^i, S^i \rangle$: node terminal

v^i : situació generalitzada; $v^i = \{g_j^i\}$

$g_j^i = \langle id_j^i, v_j^i, w_j^i, \sigma_j^i \rangle$: generalització

id_j^i : identificador

v_j^i : valor

w_i : importància estratègica

σ_i : freqüència

τ^i : tipicalitat

S^i : conjunt de solucions; $S^i = \{S_{C^i}\} \subset \mathcal{P}(F_f)$

Apèndix B

T-normes i T-conormes

Una t-norma T és una funció d'agregació d'evidències conjuntiva ($T : [0, 1]^2 \rightarrow [0, 1]$). Qualsevol t-norma compleix les propietats següents[21]:

1. Element absorbent. $T(0, w) = 0$
2. Element neutre. $T(w, 1) = T(1, w) = w$
3. Monocitat. $T(w_1, w_2) \leq T(w_3, w_4)$ si $w_1 \leq w_3$ i $w_2 \leq w_4$
4. Commutativitat. $T(w_1, w_2) = T(w_2, w_1)$
5. Associativitat. $T(w_1, T(w_2, w_3)) = T(T(w_1, w_2), w_3)$

Una t-norma pot prendre, entre d'altres, qualsevol de les formes següents:

1. $T(w_1, w_2) = \begin{cases} \min(w_1, w_2) & \text{si } \max(w_1, w_2) = 1 \\ 0 & \text{altrament} \end{cases}$
2. $T(w_1, w_2) = \max(0, w_1 + w_2 - 1)$
3. $T(w_1, w_2) = \frac{w_1 + w_2}{2 - (w_1 + w_2 - w_1 w_2)}$
4. $T(w_1, w_2) = w_1 w_2$
5. $T(w_1, w_2) = \frac{w_1 w_2}{w_1 + w_2 - w_1 w_2}$
6. $T(w_1, w_2) = \min(w_1, w_2)$

Una t-conorma és una funció de combinació d'evidències disjuntiva que satisfà les propietats següents:

1. Element absorbent. $C(1, w) = 1$
2. Element neutre. $C(0, w) = C(w, 0) = w$
3. Monocitat. $C(w_1, w_2) \leq C(w_3, w_4)$ si $w_1 \leq w_3$ i $w_2 \leq w_4$
4. Commutativitat. $C(w_1, w_2) = C(w_2, w_1)$
5. Associativitat. $C(w_1, C(w_2, w_3)) = C(C(w_1, w_2), w_3)$

Mercès a la propietat associativa, una t-conorma, igual que una t-norma, es pot estendre a n-arguments resultant en una funció C_n , que es defineix recursivament com a [122]:

$$C_n(w_1, \dots, w_{n-1}, w_n) = C(C_{n-1}(w_1, \dots, w_{n-1}), w_n)$$

Per a cada t-norma T es defineix la t-conorma dual C (respecte a la negació $N(x) = 1-x$) de la forma següent:

$$C(w_1, w_2) = 1 - T(1 - w_1, 1 - w_2)$$

Així, les t-conormes duals a les anteriors sis t-normes són [21]:

1. $C(w_1, w_2) = \begin{cases} \max(w_1, w_2) & \text{si } \min(w_1, w_2) = 0 \\ 1 & \text{altrament} \end{cases}$
2. $C(w_1, w_2) = \min(1, w_1 + w_2)$
3. $C(w_1, w_2) = \frac{w_1 + w_2}{1 + w_1 w_2}$
4. $C(w_1, w_2) = w_1 + w_2 - w_1 w_2$
5. $C(w_1, w_2) = \frac{w_1 + w_2 - 2w_1 w_2}{1 - w_1 w_2}$
6. $C(w_1, w_2) = \max(w_1, w_2)$

Apèndix C

PNEUMON-IA: l'expert i el principiant

PNEUMON-IA [166] és un sistema basat en regles desenvolupat en l'entorn MILORD [142]. L'objectiu de PNEUMON-IA és assessorar sobre l'etiologia d'una pneumònia adquirida extrahospitalàriament a partir de dades clíniques, radiològiques i dades de laboratori obtingudes des de l'inici de la malaltia [167]. L'expert responsable del desenvolupament de PNEUMON-IA ha realitzat una validació de la base de coneixements a partir d'un conjunt de casos. La validació ha posat de manifest que el comportament de PNEUMON-IA és tan bo com la d'altres metges experts en la diagnosi de pneumònies. Per aquesta raó, i atès que els sistemes basats en regles desenvolupats en MILORD distingeixen explícitament els coneixements de domini dels coneixements estratègics, hem usat PNEUMON-IA amb dues finalitats diferents: com l'expert mestre de BOLERO i, mitjançant una petita alteració, com el sistema basat en coneixements que interacciona amb BOLERO (PNEUMON-IA*). A continuació s'explica com s'ha dut a terme aquest desdoblament de la utilització de PNEUMON-IA.

C.1 PNEUMON-IA: l'expert mestre

PNEUMON-IA, tal com el seu autor l'ha dissenyat, disposa d'un conjunt de 64 metaregles d'estratègia que permeten generar diferents plans d'actuació en la resolució de problemes. La resolució de problemes en PNEUMON-IA, i en general en qualsevol aplicació desenvolupada en l'entorn MILORD, presenta un comportament reactiu. Cada vegada que es coneix informació sobre l'estat actual en la resolució d'un problema, es poden aplicar metaregles que suggereixen noves estratègies, que finalment es constitueixin en plans d'actuació. De fet, el cicle de control de MILORD (vegeu figura C.1) es basa en quatre nivells: heurístic, hipotètic, estructural i associatiu. El nivell heurístic conté metaregles que generen plans d'actuació d'acord amb un conjunt d'estratègies proporcionades pel nivell hipotètic. Un pla d'actuació de MILORD és, bàsicament, una seqüència de mòduls. Un mòdul és un agrupament d'objectius, regles i metaregles. Les regles d'un mòdul serveixen per deduir els objectius definits en el mòdul. Les metaregles del mòdul activen i inhibeixen regles del mateix mòdul. L'avaluació d'un mòdul significa encadenar les seves regles (endavant o enrera) a fi de concloure els seus objectius. El nivell hipotètic disposa d'un conjunt de metaregles a partir de les quals genera estratègies que són utilitzades pel nivell heurístic. El nivell hipotètic també és el responsable d'executar el

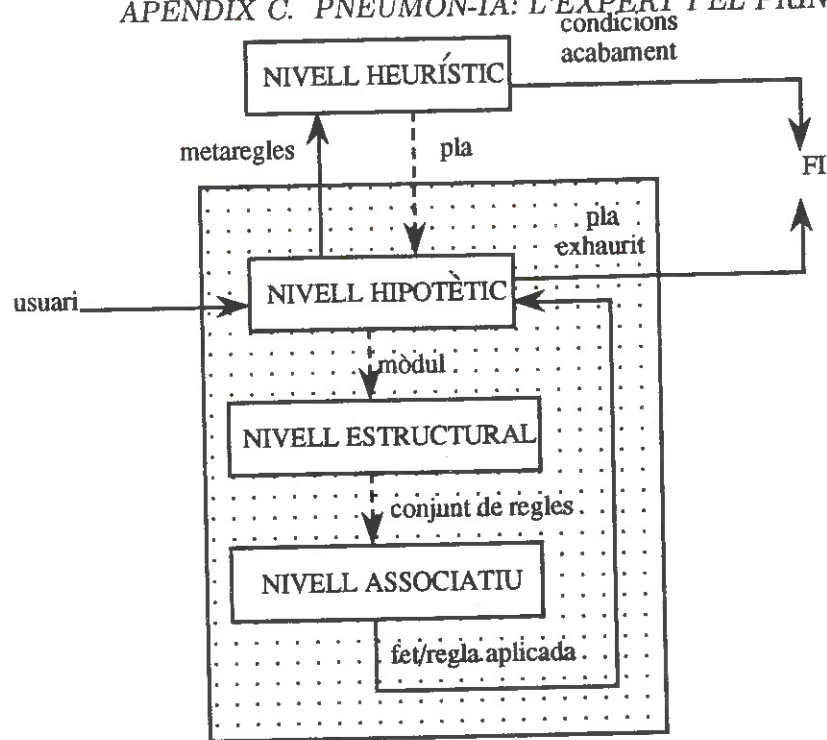


Figura C.1: Nivells de domini i de control de MILORD (de [142]). Les línies de guionets representen flux de dades, i les línies lisses flux de control.

pla d'actuació proporcionat pel nivell heurístic, per la qual cosa comunica el primer mòdul del pla al nivell estructural a fi de dur a terme la seva avaluació. Un cop el mòdul està avaluat, s'elimina del pla d'actuació i es procedeix a l'avaluació del mòdul següent. El nivell estructural disposa d'un conjunt de criteris de control (com ara encadenament enrera o endavant) a partir dels quals, i de les regles del mòdul que rep del nivell hipotètic, configura el conjunt conflicte. Finalment, el nivell associatiu disposa d'un conjunt de metaregles de preferència d'unes regles envers d'altres per seleccionar una sola regla del conjunt conflicte i, o bé l'aplica, o bé demana a l'usuari un fet necessari per a la seva aplicació.

D'aquesta manera, el control de MILORD s'inicia des del nivell hipotètic, on existeix una pla d'actuació per defecte. A continuació se succeeixen el nivell estructural i el nivell associatiu. Quan el cicle queda complet, el control retorna al nivell hipotètic. Només quan el nivell hipotètic suggereix noves estratègies per l'aplicació d'alguna de les metaregles definides en aquest nivell, el control passa al nivell heurístic. La fi de la resolució d'un problema es pot produir en dos nivells diferents:

1. en el nivell hipotètic, quan s'hagi exhaurit el pla d'actuació, o
2. en el nivell heurístic, quan alguna metaregla així ho indiqui.

En la resolució d'un problema a PNEUMON-IA, doncs, és possible distingir diferents episodis delimitats per cada canvi de pla d'actuació. Conseqüentment, "espia" la resolució de problemes de PNEUMON-IA és una manera d'obtenir casos d'entrenament per a BOLERO. Per aquesta raó s'ha procedit a la incorporació de les rutines necessàries en els punts estratègics del cicle de control de MILORD, per tal d'obtenir traces de la resolució de problemes de PNEUMON-IA que, finalment, esdevinguin casos. Cada mòdul de PNEUMON-IA que apareix en un cas s'ha entès com una acció possible de BOLERO. Els casos utilitzats en la validació de PNEUMON-IA s'han tornat a processar i s'han obtingut els casos que han

permès l'avaluació experimental de BOLERO.

El patró d'avaluació de cada cas, $PA = \langle PA_p, PA_d \rangle$, ha estat obtingut a partir dels plans dels casos processats per PNEUMON-IA i a partir de les opinions d'experts sobre cada cas que van ser recollides en la validació de PNEUMON-IA. En aquest sentit, haig d'expressar el meu agraïment al Dr. Verdaguer, que no ha posat mai cap impediment a fer ús tant del sistema PNEUMON-IA, com de la resta d'informació relacionada amb l'avaluació del seu sistema basat en regles.

C.2 PNEUMON-IA*: el principiant

Atès que a MILORD els coneixements estan estructurats en diferents nivells (els coneixements de domini i els coneixements estratègics són independents), resulta senzill eliminar una part dels coneixements de control tal i com es mostra en la figura C.2, per tal d'obtenir un sistema basat en coneixements capaç d'interaccionar amb BOLERO en l'arquitectura metanivell BOLERO-SBC.

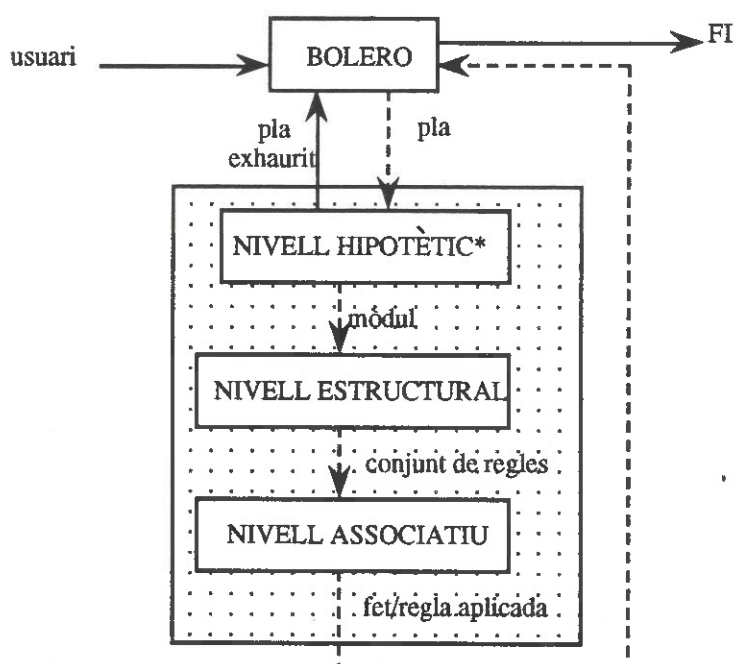


Figura C.2: BOLERO controla MILORD en l'arquitectura BOLERO-SBC. Les línies de guionets representen flux de dades, i les línies lisses flux de control. La capsa puntejada representa el que queda de MILORD després de les modificacions.

El cicle de control del sistema s'inicia ara a BOLERO. BOLERO és qui proposa un pla per defecte al nivell hipotètic de MILORD. Cada acció d'un pla de BOLERO correspon a un mòdul de MILORD, de manera que l'execució d'una acció a BOLERO és equivalent a l'avaluació d'un mòdul a MILORD. Ara no és possible aplicar metaregles en el nivell hipotètic, de manera que ara aquest nivell mai genera noves estratègies. Existeix un nivell hipotètic modificat, que

anomenem nivell hipotètic*, que només s'encarrega de fer executar el pla d'actuació. Quan el pla d'actuació s'exhaureix en el nivell hipotètic* de BOLERO-SBC, no es finalitza la resolució del problema, sinó que el nivell hipotètic* cedeix el control a BOLERO. El final del cicle de control a BOLERO-SBC ocorre quan en el nivell associatiu s'aplica una regla o s'obté un fet, i en aquest cas el control retorna a BOLERO. La decisió sobre la finalització de la resolució d'un problema és presa exclusivament per BOLERO.

Així, s'ha generat una nova versió de PNEUMON-IA, que hem anomenat PNEUMON-IA*. En la nova versió de PNEUMON-IA s'han eliminat totes les metaregles referents a estratègia. Els coneixements de PNEUMON-IA* estan constituïts només per mòduls. Cada mòdul conté objectius, regles i metaregles com a PNEUMON-IA. Les metaregles d'un mòdul, però, només afecten l'activació de les regles del mòdul, i no tenen un efecte en la generació d'estratègies, i.e. en el comportament del sistema. D'aquesta manera, PNEUMON-IA*, sense la col·laboració de BOLERO, es comporta com un principiant i només és capaç de resoldre un problema per la cerca exhaustiva de totes les solucions (i.e. executant tots els mòduls).



Vull agrair al desenvolupador de MILORD, Carles Sierra, la seva dedicació per poder dur a terme les modificacions oportunes, tant en les rutines que han permès l'obtenció de traces del procés de resolució de problemes en MILORD, com en la resta de modificacions realitzades en el control de MILORD.

Bibliografia

- [1] A. Aamodt. Knowledge-Intensive Case-Based Reasoning and Sustained Learning. In *Proc. ECAI*, 1990.
- [2] T.R. Addis, Y. Kodratoff, R.López de Màntaras, K. Morik, and E. Plaza. Four Stances on Knowledge Acquisition and Machine Learning. In Y. Kodratoff, editor, *Machine Learning-EWSL-91*, pages 514–533. Springer Verlag, March 1991. Lecture notes in Artificial Intelligence.
- [3] P.E. Agree and D. Chapman. Pengui: an Implementation of a Theory of Activity. In *Proc. AAAI*, volume 1, pages 268–272, Seattle, Washington, 1987. Distributed by Morgan Kaufmann Publishers, Inc.
- [4] D.W. Aha. Incremental Instance-Based Learning of Independent and Graded Concept Descriptions. In *Proc. IWML*, 1989.
- [5] J. Allen. Formals Models of Planning. In J. Allen and A. Tate, editors, *Readings in Planning*. Morgan Kaufmann Publishers, Inc., 1990.
- [6] R. Alterman and M. Wentworth. Determining the Important Features of a Case. In *Proc. CBR Workshop*. DARPA, May 1989.
- [7] J.A. Ambros-Ingerson and S. Steel. Integrating Planning, Execution and Monitoring. In *Proc. AAAI*, volume 1, pages 83–88, Saint-Paul, Minnesota, 1988. Distributed by Morgan Kaufmann Publishers, Inc.
- [8] J.R. Anderson, editor. *The Architecture of Cognition*. Harvard University Press, Massachusetts, 1983.
- [9] K.D. Ashley. Assessing Similarities Among Cases. A Position Paper. In *Proc. CBR Workshop*. DARPA, May 1989.
- [10] K.D. Ashley and E.L. Rissland. Waiting on Weighting: A Symbolic Least Commitment Approach. In *Proc. AAAI*, 1988.
- [11] R. Bareiss. *Exemplar-Based Knowledge Acquisition*. Academic Press, 1989.
- [12] R. Bareiss, editor. *Case-Based Reasoning Workshop*, Washington D.C., 1991. Defense Research Projects Agency (DARPA).
- [13] R.E. Bareiss and B.W. Porter. PROTOS: An Exemplar-based Learning Apprentice. In *Proc. Fourth IWML*, 1987.
- [14] R. Barletta. An Introduction to Case-Based Reasoning. *AI Expert*, August 1991.
- [15] R.K. Belew. Paradigmatic over-fitting. Personal communication.
- [16] M.A. Belmonte. *RENOIR: Un Sistema Experto para la Ayuda en el Diagnóstico de Colagenosis y Artropatías Inflamatorias*. PhD thesis, Universitat Autònoma de Barcelona, December 1990.
- [17] F. Bergadano, A. Giordana, and L. Saitta. Concept Acquisition in an Integrated EBL and SBL Environments. In *Proc. ECAI*, 1988.

- [18] F. Bergadano, Y. Kodratoff, and K. Morik. Machine Learning and Knowledge Acquisition. *AICOM*, 5(1), March 1992.
- [19] J. Blythe and T.M. Mitchell. On Becoming Reactive. In *Proc. IWML*, 1989.
- [20] P.P. Bonissone and S. Ayud. Similarity Measures for Case-Based Reasoning Systems. In *Proc. IPMU. International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 483–487. Universitat de les Illes Balears, 1992.
- [21] P.P. Bonissone and K.S. Decker. Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-off Precision and Complexity. In L.N. Kanal and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence*, pages 217–247. Elsevier Science Publishers B.V., North-Holland, 1986.
- [22] J. Breuker and B. Wielinga. Models of Expertise in Knowledge Acquisition. Technical report, University of Amsterdam, Department of Social Science Informatics, Amsterdam, August 1988. Memorandum 103 of the VF-project Acquisition of Expertise.
- [23] T. Cain, M.J. Pazzani, and G. Silverstein. Using Domain Knowledge to Influence Similarity Judgments. In *Proc. CBR Workshop*. DARPA, 1991.
- [24] J.P. Callan, T.E. Faucett, and E.L. Rissland. Adaptive Case-Based Reasoning. In *Proc. CBR Workshop*. DARPA, 1991.
- [25] L.J. Caplan and C. Schooler. Problem Solving by Reference to Rules or Previous Episodes: The Effects of Organized Training, Analogical Models, and Subsequent Complexity of Experience. *Memory & Cognition*, 18(2):215–227, 1990.
- [26] J. Carbonell. Scaling up Knowledge Based Systems via Machine Learning. Key-note presentation in the European Working Session on Learning, 1991.
- [27] J. Carbonell. Learning in Planning and Problem Solving. Summer School on Machine Learning, 1989.
- [28] J. Carbonell and M. Veloso. Integrating Derivational Analogy into a General Problem Solving Architecture. In *Proc. CBR Workshop*. DARPA, 1988.
- [29] J.G. Carbonell. Learning by Analogy: Formulating and Generalizing Plans from Past Experiences. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*. Tioga, 1983.
- [30] J.G. Carbonell. Derivational Analogy: A Theory of Reconstructive Problem Solving and Expertise Acquisition. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning. An Artificial Intelligence Approach*, volume II. Morgan Kaufmann Publishers, 1986.
- [31] J.G. Carbonell and Y. Gil. Learning by Experimentation. In *Proc. Fourth IWML*, 1987.
- [32] J.G. Carbonell, C.A. Knoblock, and S. Minton. PRODIGY: An Integrated Architecture for Planning and Learning. In Kurt VanLehn, editor, *Architectures for Intelligence. The Twenty-second Carnegie Mellon Symposium on Cognition*. Lawrence Erlbaum Associates, Publishers, 1991.
- [33] B. Carlson, J. Weinberg, and D. Fisher. Search Control, Utility, and Concept Induction. In Bruce W. Porter and Ray J. Mooney, editors, *Proc. Seventh ICML*, 1990.
- [34] J. Carreras i Martí, editor. *Diccionari de la Llengua Catalana*. Enciclopèdia Catalana, Barcelona, 1985.
- [35] R. Catrambone and H.J. Holyoak. Learning Subgoals and Methods for Solving Probability Problems. *Memory & Cognition*, 18(6):593–603, 1990.
- [36] D. Chapman. Planning for Conjunctive Goals. Technical Report 802, MIT, AI Lab, 1985.

- [37] D. Charney, L. Reder, and G.W. Kisbit. Goal Setting and Procedure Selection in Acquiring Computer Skills: A Comparison of Tutorials, Problem Solving and Learner Exploration. *Cognition and Instruction*, 7(4):323-342, 1990.
- [38] W. Clancey. Acquiring, Representing, and Evaluating a Competence Model of Diagnostic Strategy. In M.T.M. Chi, R. Glaser, and M.J. Farr, editors, *The Nature of Expertise*. Lawrence Erlbaum Associates, Publishers, 1988.
- [39] W.J. Clancey. The Advantages of Abstract Control Knowledge in Expert Systems. In *Proc. AAAI*, pages 74-78, Washington, D.C., 1983.
- [40] W.J. Clancey. The Epistemology of a Rule-based Expert System. A Framework for Explanation. *Artificial Intelligence*, 1983.
- [41] P.R. Cohen, D. Day, J. De Lisio, M. Greenberg, R. Kjeldsen, and D. Suthers. Management of Uncertainty in Medicine. *International Journal of Approximate Reasoning*, 1(1), 1987.
- [42] P.R. Cohen and E.A. Feigenbaum, editors. *Handbook of Artificial Intelligence*. Pitman, 1982.
- [43] P.R. Cohen, M. Greenberg, and J. Delisio. MU: A Development Environment for Prospective Reasoning Systems. In *Proc. AAAI*, 1987.
- [44] P.R. Cohen and A.E. Howe. How Evaluation Guides AI Research. *AI Magazine*, 9(4), 1988.
- [45] P.R. Cohen and A.E. Howe. Toward AI Research Methodology: Three Case Studies in Evaluation. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(3), 1989.
- [46] J.D. Craig. Blackboard System. *AI Review*, 2(2), 1988.
- [47] T.R. Davies. Determination, Uniformity and Relevance: Normative Criteria for Generalization and Reasoning by Analogy. In D.H. Helman, editor, *Analogical Reasoning. Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*, volume 197 of *Synthese Library*, pages 227-250. Kluwer Academic Publishers, 1988.
- [48] R. Davis and B. G. Buchanan. Meta-level Knowledge: Overview and Applications. In *Proc. IJCAI*, 1977.
- [49] T. Dean. Decision-Theoretic Control of Inference for Time-Critical Applications. *International Journal of Intelligent Systems*, 6(4):417-441, July 1991.
- [50] P. Desroches. Syclare: Système de Classification par Apprentissage et Reconnaissance des Formes. Manuel d'Utilisation. Technical Report 87/9, Centre d'Etudes Avancées de Blanes, Consejo Superior de Investigaciones Científicas (CEAB-CSIC), Blanes, Girona, Spain, 1987.
- [51] D. Dubois and J. L. Koning. Social Choice Axioms for Fuzzy Set Aggregation. *Fuzzy Sets and Systems*, 43(3), 1991.
- [52] R. Elio and P.B. Scharf. Modeling Novice-to-Expert Shifts in Problem-Solving Strategy and Knowledge Organization. *Cognitive Science*, 14:579-639, 1990.
- [53] O. Etzioni. Embedding decision-analytic control in a learning architecture. *Artificial Intelligence*, 49(1-3):129-159, 1991.
- [54] R.E. Fikes and N.J. Nilsson. STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving. In J. Allen and A. Tate, editors, *Readings in Planning*. Morgan Kaufmann Publishers, Inc., 1990.
- [55] J.H. Genenari, P. Langley, and D. Fisher. Models of Incremental Concept Formation. In J. Carbonell, editor, *Machine Learning. Paradigms and Methods*, pages 11-61. MIT/Elsevier, 1990.

- [56] D. Gentner. Structure Mapping: A Theoretical Framework for Analogy. In *Readings in Cognitive Science. A Perspective from Psychology and Artificial Intelligence*. Morgan Kaufmann Publishers, Inc., 1988.
- [57] M.P. Georgeff and A.L. Lansky. Reactive Reasoning and Planning. In J. Allen and A. Tate, editors, *Readings in Planning*, pages 729–734. Morgan Kaufmann Publishers, Inc., 1990.
- [58] P. Georgeff. Planning. In J. Allen and A. Tate, editors, *Readings in Planning*. Morgan Kaufmann Publishers, Inc., 1990.
- [59] W.B. Gevarter. The Nature and Evaluation of Commercial Expert System Building Tools. *Computer*, pages 24–41, May 1987.
- [60] F. Giunchiglia and A. Smaill. Reflection in Constructive and Non-Constructive Automated Reasoning. Technical Report DAI Research paper 375, Department of Artificial Intelligence, University of Edinburg, 1988.
- [61] L. Godo, R. López de Màntaras, C. Sierra, and A. Verdaguer. Managing Linguistically Expressed Uncertainty in MILORD: Application to Medical Diagnosis. In *Proc. Seventh International Symposium on Expert Systems and Applications*, pages 571–596, Avignon, France, 1987.
- [62] R.H. Granger, K.P. Eiselt, and J.K. Holbrook. Parsing with Parallelism: A Spreading Activation Model of Inference Processing Text Understanding. In J.L. Kolodner and C.K. Riesbeck, editors, *Experience, Memory and Reasoning*, chapter 14, pages 227–246. Lawrence Erlbaum Associates, 1986.
- [63] T.R. Gruber. *The Acquisition of Strategic Knowledge*, volume 4 of *Perspectives in Artificial Intelligence*. Academic Press Inc., 1989.
- [64] R.P. Hall. Computational Approaches to Analogical Reasoning: A Comparative Analysis. *Artificial Intelligence*, 39:39–120, September 1989.
- [65] K. Hammond, editor. *Case-Based Reasoning Workshop*, Pensacola Beach, Florida, 1989. Defense Research Projects Agency (DARPA).
- [66] K. Hammond, T. Converse, and M. Marks. Towards a Theory of Agency. In K.P. Sycara, editor, *Proc. Workshop on Innovative Approaches to Planning, Scheduling and Control*, San Diego, California, 1990. DARPA, Distributed by Morgan Kaufmann Publishers Inc.
- [67] K.J. Hammond. Case-based Planning. In *Proc. CBR Workshop*. DARPA, 1988.
- [68] K.J. Hammond. *Case-Based Planning. Viewing Planning as a Memory Task*, volume 1 of *Perspectives in Artificial Intelligence*. Academic Press, Inc., 1989.
- [69] P.J. Hayes. The Frame Problem and Related Problems in Artificial Intelligence. In J. Allen and A. Tate, editors, *Readings in Planning*. Morgan Kaufmann Publishers, Inc., 1990.
- [70] B. Hayes-Roth and M. Hewett. BB1: An Implementation of the Blackboard Control Architecture. In Robert Englemore and Tony Morgan, editors, *Blackboard Systems*. Addison Wesley, 1988.
- [71] J.A. Hendler. *Integrating Marker-passing and Problem-solving. A Spreading Activation Approach to Improved Choice in Planning*. Lawrence Erlbaum Associates, Publishers, 1988.
- [72] J.A. Hendler. The Design and Implementation of Marker-passing Systems. *Connection Science*, 1(1), 1989.
- [73] E. Hudlická. Construction and Use of Causal Model for Diagnosis. *International Journal of Intelligent Systems*, 3(3), 1988.
- [74] IJCAI (Tutorial MA2). *Case-Based Reasoning*, 1985.

- [75] L.P. Kaelbling. An Architecture for Intelligent Reactive Systems. In J. Allen and A. Tate, editors, *Readings in Planning*, pages 713–728. Morgan Kaufmann Publishers, Inc., 1990.
- [76] S. Kedar-Cabelli. Analogy - From a Unified Perspective. In D.H. Helman, editor, *Analogical Reasoning. Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*, volume 197 of *Synthese Library*, pages 65–103. Kluwer Academic Publishers, 1988.
- [77] D. Kibler and P. Langley. Machine Learning as an Experimental Science. In D. Sleeman, editor, *Proc. EWSL*, pages 81–92, Glasgow, UK, October 1988. Turing Institute, Pitman.
- [78] Y. Kodratoff. *Introduction to Machine Learning*. Morgan Kaufmann, 1988.
- [79] Y. Kodratoff and R.S. Michalski, editors. *Machine Learning. An Artificial Intelligence Approach*, volume III. Morgan Kaufmann Publishers, 1990.
- [80] J. Kolodner, editor. *Case-Based Reasoning Workshop*, Clearwater Beach, Florida, 1988. Defense Research Projects Agency (DARPA).
- [81] J.L. Kolodner. A Process Model of Case-based Reasoning in Problem Solving. In *Proc. IJCAI*, 1985.
- [82] J.L. Kolodner. Extending Problem Solver Capabilities Through Case-based Inference. In *Proc. Fourth IWML*, 1987.
- [83] J.L. Kolodner. Retrieving Events from a Case Memory: A Parallel Implementation. In *Proc. CBR Workshop*. DARPA, 1988.
- [84] J.L. Kolodner. Judging Which is the Best Case for a Case-Based Reasoner. In *Proc. CBR Workshop*. DARPA, 1989.
- [85] J.L. Kolodner and R.L. Simpson Jr. Problem Solving and Dynamic Memory. In J. L. Kolodner and C.K. Riesbeck, editors, *Experience, Memory and Reasoning*, chapter 6. Lawrence Erlbaum Associates, Publishers, Hillsdale, N.J., 1986.
- [86] Koton and Chase. Knowledge Representation in a Case-based System: Defaults and Exceptions. In *Proc. KR*, 1989.
- [87] P. Koton. A Medical Reasoning Program that Improves with Experience. In *Proc. SCAMC (IEEE)*, 1988.
- [88] P. Koton. Reasoning About Evidence in Causal Explanation. In *Proc. AAAI*, 1988.
- [89] P. Koton. Using a Case Memory to Integrate Case-based and Causal Reasoning. In *Proc. Tenth Conference of the Cognitive Science Society*, 1988.
- [90] J.E. Laird and P.S. Rosenbloom. Integrating Execution, Planning, and Learning in Soar for External Environments. In *Proc. AAAI*, pages 1022–1029, 1990.
- [91] J.E. Laird, P.S. Rosenbloom, and A. Newell. Chunking in SOAR: The Anatomy of a General Learning Mechanism. *Machine Learning*, 1(1):11–46, 1986.
- [92] P. Langley. Learning to Search: From Weak Methods to Domain-Specific Heuristics. *Cognitive Science*, 9:217–260, 1985.
- [93] P. Langley. Strategy Acquisition Governed by Experimentation. In L. Steels and J.A. Campbell, editors, *Progress in Artificial Intelligence*. Ellis Horwood, 1985.
- [94] P. Langley. Machine Learning as an Experimental Science. *Machine Learning*, 3(1):5–8, 1988.
- [95] P. Langley and M. Drummond. Toward an Experimental Science of Planning. In K.P. Sycara, editor, *Proc. Workshop on Innovative Approaches to Planning, Scheduling and Control*, San Diego, California, 1990. DARPA, Distributed by Morgan Kaufmann Publishers Inc.

- [96] N. Lavrač. *Principles of Knowledge Acquisition in Expert Systems*. PhD thesis, University In Maribor, Faculty of Technical Sciences, Ljubljana, 1990.
- [97] W.G. Lenhart. Case-based Reasoning as a Paradigm for Heuristic Search. Technical Report 107, Department of Computer and Information Science, Univ. Massachussets, october 1987.
- [98] V.R. Lesser and L.D. Erman. A Retrospective View of the HERSAY-II Architecture. In Robert Englemore and Tony Morgan, editors, *Blackboard Systems*. Addison Wesley, 1988.
- [99] B. López. CONKRET: A Control Knowledge Refinement Tool. In M. Ayel and J.P. Laurent, editors, *Validation, Verification and Test of Knowledge-Based Systems*, chapter 13, pages 191–203. John Wiley & Sons, 1991.
- [100] B. López, P. Meseguer, and E. Plaza. Knowledge Based Systems Validation: A State of The Art. *AICOM*, 3(2):58–72, June 1990.
- [101] R. López de Màntaras. A Distance-Based Attribute Selection Measure for Decision Tree Induction. *Machine Learning*, 6(1):81–92, 1991.
- [102] P. Maes. Issues in Computational Reflection. In P. Maes and D. Nardi, editors, *Meta-level Architectures and Reflection*, pages 21–35. Elsevier Science Pub. B. V., 1988.
- [103] R.S. Mallory. Sources of Classification Accuracy in PROTOS. Technical Report AI89-118, AI Lab, The University of Texas at Austin, December 1989.
- [104] P. Meseguer. *Validation of Multi-Level Rule-Based Expert Systems*. PhD thesis, Universitat Politècnica de Catalunya, Departamanet de Llenguatges i Sistemes Informàtics, April 1992.
- [105] R.S. Michalski. A Theory and Methodology of Inductive Learning. In R. Michalski, J. Carbonell, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*. Tioga, 1983.
- [106] S. Minton. *Learning Effective Search Control Knowledge: An Explanation-Based Approach*. PhD thesis, Computer Science Department, Carnegie Mellon University, 1988. Technical Report CMU-CS-88-133.
- [107] T. Mitchell. Explanation-Based Learning. Summer School on Machine Learning, 1989.
- [108] T.M. Mitchell. Becoming Increasingly Reactive. In *Proc. AAAI*, pages 1051–1058, 1990.
- [109] T.M. Mitchell. Plan-Then-Compile Architectures. *SIGART Bulletin*, 2(4):136–139, August 1991.
- [110] T.M. Mitchell, J. Allen, P. Chalosair, I. Cheng, O. Etzioni, and M. Ringuette. Theo: A Framework for Self-improving Systems. In K. VanLehn, editor, *Architectures for Intelligence*. Erlbaum, Hillsdale, NJ, 1990.
- [111] T.M. Mitchell and R. Banerji. Learning by Experimentation: Acquiring and Refining Problem-solving Heuristics. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*. Tioga, 1983.
- [112] T.M. Mitchell, R.M. Keller, and S.T. Kedar-Cabelli. Explanation-Based Generalization: A Unifying View. *Machine Learning*, 1(1), 1986.
- [113] T.M. Mitchell, S. Mahadevan, and L.I. Steinberg. LEAP: A Learning Apprentice for VLSI Design. In *Proc. IJCAI*, 1985.
- [114] A. Newell and H.A. Simon. *Human Problem Solving*. Prentice-Hall, Inc., 1972.
- [115] H.P. Nii. Blackboard Systems: Blackboard Application Systems, Blackboard Systems from a Knowledge Engineering Perspective. *The AI Magazine*, pages 82–106, August 1986.

- [116] H.P. Nii. Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. *The AI Magazine*, 7(2):38–53, 1986.
- [117] D.C. Park, D.A. Smith, and J.C. Cavanaugh. Metamemories of memory researchers. *Memory & Instruction*, 18(3):321–327, 1990.
- [118] R.S. Patil. Review of Causal Reasoning in Medical Diagnosis. In *Tenth Annual Symposium on Computer Applications in Medical Care (IEEE-SCAMC)*, 1986.
- [119] M.J. Pazzani. Integrating Explanation-based and Empirical Learning Methods in OCCAM. In *Proc. EWSL*, 1988.
- [120] M.J. Pazzani and W. Sarrett. A Framework for Average Case Analysis of Conjunctive Learning Algorithms. *Machine Learning*, 9:349–372, 1992.
- [121] J. Pepper and G. Kahn. Knowledge Craft: An Environment for Rapid Prototyping of Expert Systems. In *Proc. SME Conference on Artificial Intelligence for the Automotive Industry*, Detroit, Michigan, March 1986.
- [122] N. Piera, Ph. Desroches, and J. Aguilar-Martin. LAMDA: An Incremental Conceptual Clustering Method. Technical Report 89420, Laboratoire d'Automatique et d'Analyse des Systèmes (LAAS), 7, Avenue du Colonel Roche, 31077 Toulouse Cedex., France, 1989.
- [123] E. Plaza. *EAR: Sistema d'Ajut a l'Adquisició i Estructuració de Coneixements*. PhD thesis, UPC, FIB, May 1987.
- [124] B.W. Porter, R. Bareiss, and R.C. Holte. Concept Learning and Heuristic Classification in Weak-Theory Domains. *Artificial Intelligence*, 45(1–2):229–263, September 1990.
- [125] B.W. Porter and R.E. Bareiss. PROTOS: An Experiment in Knowledge Acquisition for Heuristic Classification Tasks. Technical Report AI TR86-35, The University of Texas at Austin, Artificial Intelligence Laboratory, 1986.
- [126] L. Portinale. Generalization Handling in a Dynamic Case Memory. Personal communication.
- [127] J.R. Quinlan. Learning Efficient Classification Procedures and Their Application to Chess End Games. In R. Michalski, J. Carbonell, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*. Tioga, 1983.
- [128] J.R. Quinlan. Induction of Decision Trees. *Machine Learning*, 1(1), 1986.
- [129] J.R. Quinlan. The Effect of Noise on Concept Learning. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning. An Artificial Intelligence Approach*, volume II, chapter 6. Morgan Kaufmann Publishers, 1986.
- [130] P.S. Rosenbloom and A. Newell. The Chunking of Goal Hierarchies: A Generalized Model of Practice. In R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning. An Artificial Intelligence Approach*, volume II. Morgan Kaufmann Publishers, 1986.
- [131] B.H. Ross, S.J. Perkins, and P.L. Tenpenny. Reminding-based Category Learning. *Cognitive Psychology*, 22:460–492, 1990.
- [132] Russell and Wefald. Principles of Metareasoning. In *Proc. KR*, 1989.
- [133] S. Russell. Analogy by Similarity. In D.H. Helman, editor, *Analogical Reasoning. Perspectives of Artificial Intelligence, Cognitive Science, and Philosophy*, volume 197 of *Synthese Library*, pages 251–269. Kluwer Academic Publishers, 1988.
- [134] S. Russell and E. Wefald. *Do the Right Thing. Studies in Limited Rationality*. The MIT Press, Cambridge, Massachusetts, 1991.

- [151] J. Wolstenholfe, R. W. Smith, and J. W. Smith, "Learning and Reasoning: A Missing Link in the Chain of Artificial Intelligence," in *Artificial Intelligence: The State of the Art*, J. W. Smith, ed., North-Holland, Amsterdam, 1980, pp. 1-10.
- [152] G. Torres, "Planificación para la Producción de Problemas," *Mundo Electrónico*, 1980.
- [153] University of Aberdeen, *Proceedings of the MLRT Workshop on Knowledge Compilation, and Speedup Learning*, Aberdeen, Scotland, July 1980.
- [154] P.E. Utgoff, *Machine Learning of Inductive Bias*, Kluwer, 1986.
- [155] P.E. Utgoff, "Study of Bias for Inductive Concept Learning," in R.S. Sutton, J.G. Carbonell, and T.M. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, volume II, chapter 5, Morgan Kaufmann Publishers, 1986.
- [156] W. van de Velde, *Learning from Experience*, PhD thesis, Vrije Universiteit Brussel, Artificial Intelligence Laboratory, January 1988, Technical Report 88-1.
- [157] W. van de Velde, "U/L or Taming the Multiplexer," in Katharina Morik, editor, *Proc. ECAI-88*, Morgan Kaufmann, 1988.
- [158] P. Van Haeften, *Meta-level Inference Systems*, Pitman, London, 1981.