

2
IA

Tècniques de programació en intel·ligència artificial

Carles Sierra García

Programa de Doctorat Intel·ligència Artificial

Departament de Llenguatges i Sistemes Informàtics

Temes d'Intel·ligència Artificial

/sie

IA

Temes d'Intel·ligència Artificial

Temes d'Intel·ligència Artificial 2



Tècniques de programació en intel·ligència artificial

Carles Sierra García

Institut d'Investigació en Intel·ligència Artificial del CSIC



Responsable de la col·lecció: Carles Riba Romeva
Coordinadora de l'àrea: Núria Castell Ariño

Aquesta publicació s'acull a la política de normalització lingüística i ha comptat amb un ajut del Departament de Cultura i de la Direcció General d'Universitats, de la Generalitat de Catalunya.

Aquesta publicació ha comptat amb l'ajut del Programa de Doctorat Intel·ligència Artificial del Departament de Llenguatges i Sistemes Informàtics de la UPC.

En col·laboració amb el Servei de Llengües i Terminologia de la UPC

Primera edició: desembre de 1994

© Carles Sierra García, 1994
© Edicions UPC, 1994
Edicions de la Universitat Politècnica de Catalunya, SL
C. Jordi Girona Salgado, 31, 08034 Barcelona

Producció: Servei de Publicacions de la UPC
i CPDA (Centre de Publicacions d'Abast)
Av. Diagonal, 647, ETSEIB, 08028 Barcelona

Dipòsit legal: B-11.402-94
ISBN 84-7653-397-7

Al Carles i l'Enric

Presentació

Benvolgut lector, la primera pregunta que potser et faràs és: ¿per què un llibre de tècniques de programació en intel·ligència artificial? És potser la programació en intel·ligència artificial quelcom diferent de la programació normal? La programació en intel·ligència artificial (IA) presenta unes característiques particulars que si bé no són exclusives d'aquesta àrea de la informàtica, sí que tenen un caràcter més marcat. La primera és que els programes no es valoren només per la seva qualitat computacional, sinó per la seva comprensibilitat i facilitat de manipulació. L'escriptura de programes s'entén com un mitjà formal de transmetre i manipular idees entre les persones; així un programa més llegible, més clar, més elegant és sovint preferit a un programa eficient. Aquesta, diguem-ne, filosofia de la programació, a més de les característiques dels problemes que la intel·ligència artificial ataca, ha fet que els llenguatges simbòlics hagin estat majoritaris entre els seus investigadors i enginyers. Així, llenguatges com el LISP o el Prolog sobreviuen en gran mesura gràcies al seu ús en intel·ligència artificial; amb el LISP arrelat fortament en els laboratoris americans, el Prolog al Japó i ambdós distribuïts força uniformement a Europa. La segona característica particular és la metodologia de programació. Després de la crisi del *software* de finals de la dècada dels seixanta la programació va concebre's com un procés en dues etapes, una primera d'especificació del problema que s'ha de resoldre i una segona d'implementació, sempre amb l'objectiu de poder *provar* que la implementació satisfieia allò que l'especificació indicava. Aquest camí ha estat llarg i ha donat lloc a llenguatges cada cop més avançats en aquesta línia. Els llenguatges de la intel·ligència artificial han semblat refractaris a aquesta evolució i sovint això ha estat motiu de crítiques, que suggerien que la programació en intel·ligència artificial estava mancada de metodologia. La realitat ha estat un altra ben diferent. La programació en intel·ligència artificial és una programació que podríem anomenar *especulativa*; en aquesta programació l'especificació i la implementació es duen a terme de forma paral·lela. Els problemes que cal estudiar són sovint mal definits i poc estructurats, la qual cosa provoca que la programació esdevingui una tasca d'enginyeria experimental, on la comprensió del problema que s'ha de resoldre i la implementació de la solució són difícilment destriables. L'última característica que destacaria de la programació en intel·ligència artificial és que sovint es defineixen llenguatges específics per a grups de problemes semblants. És una mica conseqüència de la primera de les característiques esmentades. Es vol apujar el nivell expressiu dels llenguatges perquè la lectura i manipulació experimental posterior dels programes sigui més fàcil.

Moltes vegades en els congressos de l'àrea es fan presentacions de llenguatges en lloc de presentació de solucions concretes a problemes, i això és així perquè en gran mesura trobar el llenguatge adient als problemes implica trobar-los una solució. Aquesta última característica és una de les raons de l'èxit que llenguatges com el LISP han tingut en l'àrea, a causa de la seva facilitat d'extensió i de la facilitat de programació d'intèrprets i metaintèrprets. Criteris com ara l'eficiència no apareixen en la llista que he fet, i és que encara que per a molts sigui un anatema, l'eficiència, malgrat no ser menystinguda, juga un paper de segon nivell respecte als criteris anteriors. Apareixen abans els llenguatges i els models d'una forma temptativa que les seves implementacions eficients. Que pensi el lector en l'aparició dels sistemes de producció (1972) i el disseny de l'algorisme *rete* (1982) o bé el disseny de llenguatges orientats a objectes com a extensions de LISP, Smalltalk (1969), Flavors (1979), New Flavors (1981) i CommonLoops (1982), que introduïren la majoria de les idees que actualment trobem implementades de forma eficient en llenguatges com el C++ (1986).

Tanmateix, aquest llibre no pretén seguir un esquema de manual d'intel·ligència artificial, estructurat habitualment al voltant de la representació dels coneixements, sinó que vol ser un llibre on es puguin trobar els mecanismes de disseny de llenguatges i d'implementació que aquests manuals no contenen. El que he intentat fer és una recopilació d'aquestes tècniques utilitzant, de vegades, exemples de la literatura i, d'altres, exemples de pròpia creació amb l'objectiu que puguin servir a aquells que s'enfronten amb la tasca de fer projectes d'intel·ligència artificial que requereixin el disseny de llenguatges o models de càlcul propis, diferents dels que es puguin trobar en llenguatges existents, o bé que requereixin una combinació de tècniques pròpia, adequada als tipus de problemes que s'han de resoldre.

El primer dels capítols fa un repàs a algunes característiques del llenguatge LISP que són de gran utilitat de cara a entendre el que vindrà a continuació. No és un capítol per aprendre el llenguatge, que llibres per a això n'hi ha molts i molts bons, sinó alguns elements de programació d'ordre superior i definició de macros que el lector poc habituat a treballar amb el LISP pot tenir oblidats o rovellats. Els capítols següents estan estructurats com a grups de tècniques en els quatre paradigmes: funcional, declaratiu, imperatiu i reflexiu, de forma que el lector pugui trobar ràpidament la descripció d'allò que l'interessa. Finalment, la bibliografia recull un grup de llibres i articles on es poden trobar les tècniques descrites aquí de forma més extensa i detallada.

Carles Sierra

Blanes, 1994

INDEX

1 Elements del CommonLisp	11
1.1 Funcions de segon ordre	11
1.2 Macros	13
1.3 Model d'entorns	15
2 Tècniques de programació funcional	19
2.1 Currificació	19
2.2 Continuacions	21
2.3 Corrents	31
2.4 Avaluació mandrosa	38
2.5 Enregistrament	43
2.6 Representació de l'estat	45
2.7 Corrents i intel·ligència artificial	46
3 Tècniques de programació imperativa	49
3.1 Modularitat i estat local	49
3.2 Programació de restriccions	52
3.3 Encapsulament i pas de missatges	62
3.4 Delegació	68
3.5 Herència	69
4 Tècniques de programació declarativa	71
4.1 Metainterpretació en programació lògica	71
4.2 Avaluació parcial	74
5 Tècniques de programació reflexiva	79
5.1 El control	79
5.2 Sistemes reflexius	81
5.3 Reflexió funcional	84
5.4 Reflexió lògica	90
Bibliografia	101

Carles Sierra García

Doctor en Informàtica per la UPC.

Actualment treballa a l'Institut d'Investigació en Intel·ligència Artificial com a col·laborador científic del Consell Superior d'Investigacions Científiques (CSIC).

Els seus principals temes d'interès són la reflexió computacional i els sistemes basats en coneixements.

Aquest llibre pretén ser una eina de treball on es puguin trobar els mecanismes de disseny de llenguatges i d'implementació que habitualment no apareixen en els manuals d'intel·ligència artificial. Les tècniques que s'hi presenten es troben repartides en tot un seguit de llibres comentats a l'apartat de la bibliografia i que de vegades poden resultar d'accés difícil als estudiants de la llicenciatura o dels cursos de doctorat.

L'objectiu d'aquest treball de recopilació és ajudar a aquells que s'enfrontin a la tasca de fer projectes d'intel·ligència artificial que requereixin el disseny de llenguatges o models de càlcul propis, diferents dels que es puguin trobar en els llenguatges ja existents, o bé que requereixin una combinació de tècniques pròpies, adequades als tipus de problemes que s'han de resoldre.

El Departament de Llenguatges i Sistemes Informàtics (LSI) de la UPC és l'encarregat de l'ensenyament dels cursos de software que s'ofereixen en diversos centres d'aquesta Universitat, dels quals els de més tradició són la Facultat d'Informàtica de Barcelona (FIB) i l'Escola Tècnica Superior d'Enginyers Industrials de Barcelona (ETSEIB). Aquest Departament està dividit en cinc seccions que corresponen a les diferents àrees de recerca: Informàtica gràfica, Informàtica teòrica, Intel·ligència artificial, Programació i sistemes d'informació. LSI té al seu càrrec dos Programes de Doctorat: Intel·ligència Artificial i Software.

AIF