
Sistemes basats en el coneixement

PID_00267990

Vicenç Torra i Reventós

Revisió a càrrec de
Jasmina Casals Terré

Temps mínim de dedicació recomanat: 5 hores



Vicenç Torra i Reventós

Jasmina Casals Terré

La revisió d'aquest recurs d'aprenentatge UOC ha estat coordinada pel professor: Carles Ventura Royo (2019)

Tercera edició: setembre 2019
© Vicenç Torra i Reventós
Tots els drets reservats
© d'aquesta edició, FUOC, 2019
Av. Tibidabo, 39-43, 08035 Barcelona
Realització editorial: FUOC

Cap part d'aquesta publicació, incloent-hi el disseny general i la coberta, no pot ser copiada, reproduïda, emmagatzemada o transmesa de cap manera ni per cap mitjà, tant si és elèctric com químic, mecànic, òptic, de gravació, de fotocòpia o per altres mètodes, sense l'autorització prèvia per escrit dels titulars dels drets.

Índex

Introducció	5
Objectius	6
1. Introducció als sistemes basats en el coneixement	7
1.1. Apunts sobre la construcció d'un sistema basat en el coneixement	8
1.1.1. La construcció del model	9
1.1.2. Validació i verificació	13
2. La representació del coneixement	14
2.1. Nivells d'un formalisme de representació del coneixement	16
2.2. Aspecte formal i aspecte inferencial	18
2.3. Tipus de coneixement	19
2.4. L'aspecte inferencial i la cerca	20
2.5. Altres aspectes sobre els formalismes de representació del coneixement	21
3. Sistemes basats en regles	25
3.1. Aspecte inferencial en un sistema basat en regles	26
3.2. Anàlisi dels sistemes basats en regles	30
4. Sistemes amb representació estructurada	32
4.1. Aspecte formal	33
4.2. Aspecte inferencial	37
4.3. Anàlisi dels sistemes de marcs	45
5. Sistema de raonament basat en casos	47
5.1. Aspecte inferencial	48
6. Sistemes de raonament basat en models	51
6.1. Aspecte formal	52
6.2. Aspecte inferencial	53
Activitats	55
Glossari	59
Bibliografia	60

Introducció

Heuristic Search Hypothesis. The solutions to problems are represented as symbol structures. A physical symbol system exercises its intelligence in problem solving by search – that is, by generating and progressively modifying symbol structures until it produces a solution structure.

A. Newell; H.A. Simon (1976). "Computer Science as Empirical Inquiry: Symbols and Search". *Communications of the ACM* (vol. 3, núm. 19, pàg. 113-126).

L'èxit dels sistemes basats en el coneixement i, en particular, dels sistemes basats en regles els anys vuitanta va ser una de les claus del *boom* de la intel·ligència artificial. Diverses aplicacions van mostrar la utilitat d'aquests sistemes quan es tractava de problemes amb un domini fitat.

En aquest mòdul presentarem els elements principals per a entendre els sistemes basats en el coneixement. El mòdul està dividit en sis apartats.

El primer apartat constitueix una introducció als sistemes basats en el coneixement. Es veuen els seus components principals i les fases més importants que s'han de tenir en compte en el seu desenvolupament.

A continuació es passa a un segon apartat en el qual es planteja la representació del coneixement. Comencem definint què és el coneixement, per passar després a definir-ne els conceptes més importants. També es descriu aquí la qüestió de la cerca, i el seu paper en els sistemes de representació del coneixement. Aquesta qüestió apareixerà més endavant en parlar de formes concretes de representació. L'apartat acaba amb alguns dels problemes relatius a la representació.

L'apartat tercer està dedicat als sistemes en què el coneixement està representat mitjançant regles. Es descriu el funcionament d'aquest tipus de sistema.

L'apartat quart introdueix els mecanismes de representació estructurada centrant-nos en els sistemes de marcs. Veurem aquí les característiques dels sistemes que fan servir aquesta representació.

El mòdul segueix amb dos apartats dedicats al raonament basat en casos i al raonament basat en models. Es descriu el funcionament d'aquests sistemes, i es comparen les seves característiques.

Objectius

Aquest mòdul didàctic vol assolir els objectius següents:

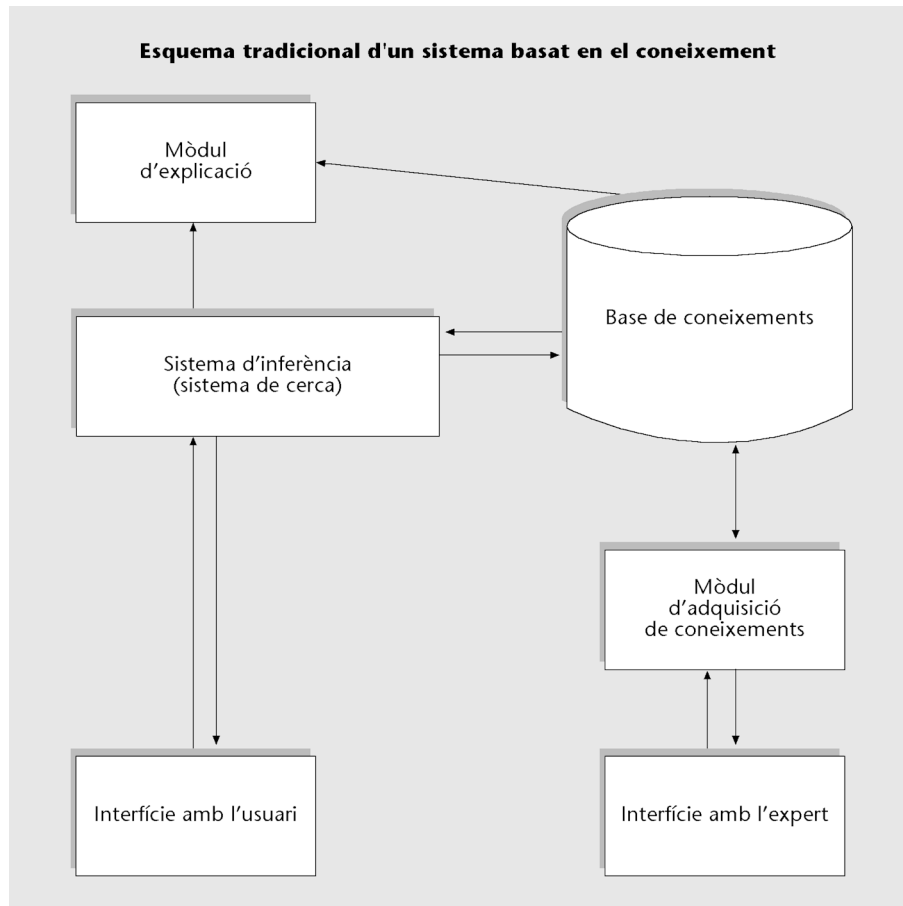
- 1.** Obtenir una visió general del que són els sistemes basats en el coneixement.
- 2.** Veure els sistemes basats en el coneixement com a eines per a resoldre problemes concrets.
- 3.** Estudiar diferents eines de representació del coneixement.
- 4.** Conèixer els pros i contres dels mètodes de representació.
- 5.** Veure el paper que té la cerca en els sistemes basats en el coneixement i en els mecanismes de representació del coneixement.

1. Introducció als sistemes basats en el coneixement

Com el seu nom indica, els sistemes basats en el coneixement o SBC (*knowledge-based systems* o *knowledge systems*) resolen els problemes utilitzant de manera intensiva coneixement del camp d'aplicació.

Per exemple, en un sistema de diagnosi mèdica hi haurà informació sobre les relacions entre els símptomes i les malalties. Aquesta informació és la que guiarà la cerca quan es plantegi un problema concret. Per exemple, quina és la malaltia que té un determinat pacient.

Figura 1



Per tal de tenir en compte tant el coneixement com els mecanismes de cerca un sistema basat en el coneixement es veu tradicionalment d'acord amb l'arquitectura de la figura 1. Aquesta figura subratlla els principals mòduls que l'integren:

a) **Base de coneixements:** correspon al magatzem de coneixement del sistema. S'hi guarda tot allò que és necessari per a guiar el procés dirigit a trobar les solucions.

b) **Subsistema d'inferència:** és la part que raona sobre les solucions dels problemes. El raonament s'activa a partir, per exemple, de la pregunta d'un usuari i tot el procés de cerca està dirigit per la base de coneixements a respondre aquesta pregunta. A aquest subsistema se l'anomena a vegades *motor d'inferència*.

c) **Interfície amb l'usuari:** és la que permet que els usuaris finals puguin interactuar amb el sistema.

A més, a vegades hi ha mòduls d'explicació i d'adquisició del coneixement. El primer permet de construir una descripció de com es construeix la solució. Per a fer-la es basen amb el que fa el sistema d'inferència i amb el coneixement de la base de coneixements. El mòdul d'adquisició del coneixement ajuda en tot el procés de construcció de la base de coneixements.

Tot i que la descomposició en mòduls presentada permet de subratllar els diferents elements d'un sistema d'aquestes característiques, les implementacions reals d'aquests sistemes no mantenen sempre aquesta estructura. De tota manera, en parlar de les formes concretes de representació ens apareixerà un aspecte formal i un aspecte inferencial que més o menys corresponen a la base de coneixements i al sistema d'inferència. També es veurà que els dos aspectes estan molt lligats i que sovint no són independents i, per tant, no poden formar mòduls separats.

En aquest mòdul es descriuran els elements bàsics dels sistemes basats en el coneixement. Començarem amb la descripció del procés de construcció d'aquests sistemes. Un cop vist el procés, veurem diferents mecanismes de representació del coneixement i descriurem el funcionament d'alguns sistemes.

1.1. Apunts sobre la construcció d'un sistema basat en el coneixement

La primera qüestió que cal tractar abans de començar la construcció d'un sistema basat en el coneixement és decidir si realment necessitem aquest sistema. A causa que la construcció d'un sistema d'aquestes característiques requereix una bona inversió de temps i recursos és convenient avaluar si aquesta és l'opció més bona.

Algunes de les qüestions que cal tenir en compte a l'hora de prendre la decisió són aquestes:

L'enginyeria del coneixement

L'àrea de la intel·ligència artificial que estudia el procés de construcció de sistemes basats en el coneixement és l'anomenada *enginyeria del coneixement*, de manera anàloga al fet que l'enginyeria del programari estudia la construcció de programari.

Lectures complementàries

Per a més detalls sobre l'enginyeria del coneixement podeu consultar les obres següents:

G. Guida; G. Tasso (1994). *Design and Development of Knowledge-Based Systems*. John Wiley and Sons.

M. Stefik (1995). *Introduction to Knowledge Systems*. Morgan Kauffman.

- Els problemes a resoldre no es poden resoldre amb les tècniques de programació habitual.
- El domini de l'aplicació disposa d'un coneixement ben estructurat, i hi ha experts que el coneixen.
- En els entorns en què es necessita el coneixement, no sempre es disposa de l'experiència humana.
- El cost del desenvolupament del sistema basat en el coneixement s'ha de poder justificar en termes de la utilitat posterior del sistema.

1.1.1. La construcció del model

Un cop s'ha decidit que la construcció és convenient passarem a definir quins són els problemes que volem resoldre i a definir la base de coneixements (BC)¹ convenientment. La base de coneixements haurà de contenir tot el coneixement que el sistema necessita per a resoldre els problemes previstos. En general, hi ha dues maneres d'aconseguir aquest coneixement: per mitjà d'un expert (aquest és el cas que s'ha il·lustrat en la figura 1) o mitjançant tècniques d'aprenentatge.

⁽¹⁾Abreugem *base de coneixements* amb la sigla BC.

En el primer cas, durant el procés anomenat *adquisició del coneixement* un expert juntament amb l'esmentat enginyer del coneixement formulen el coneixement.

El procés d'adquisició de coneixement consisteix en la construcció d'un model del camp on s'ha d'aplicar el sistema (el domini d'aplicació) sobre la base de l'experiència de l'expert.

L'expert és el qui sap del camp. Sap quins problemes es plantegen, i les tècniques apropiades per a resoldre'ls. Sap com avaluar les dades i com tractar casos mal definits o amb incertesa. L'enginyer del coneixement, en canvi, és l'expert en les tècniques d'intel·ligència artificial i de representació del coneixement. Les tasques principals d'aquest darrer són seleccionar les eines adequades per a la representació, ajudar a l'expert a formular el coneixement necessari i implementar-lo en una base de coneixements de manera eficient.

Per a la construcció del model es poden tenir en compte les tres funcions que ha de satisfer:

1) Un model és una abstracció que es fa servir per a uns objectius concrets (s'utilitza en un marc concret per a fer una determinada tasca). Quan sabem l'objectiu del model, podem fer una aproximació del fenomen que volem mo-

delitzar, aconseguint només la informació que és rellevant per a aconseguir l'acompliment de la tasca. D'aquesta manera, la complexitat del problema es redueix.

2) Un model representa una manera estructurada de comprendre les entitats i els processos que permeten construir la solució d'una tasca en el món real.

3) Un model ha de permetre comprendre més bé el procés i també predir.

El procés de construcció d'un sistema basat en el coneixement i, en particular, del model no és una tasca fàcil. D'una banda, perquè la comunicació entre l'expert i l'enginyer del coneixement és difícil a causa de les diferències de formació. De l'altra, el model ha d'explicitar (i fer-ho de manera consistent) tot allò necessari per a resoldre el problema. També hi ha les dificultats inherents en els formalismes de representació (per exemple, dificultats per a expressar coneixement incert, relacions espacials, el problema de representar el canvi, etc.). Per a facilitar aquest procés s'han desenvolupat algunes metodologies. Algunes d'aquestes metodologies són, per exemple, les anomenades KADS, Components of Expertise i Generic Tasks.

Representació del canvi: el problema del marc

El problema del marc (*frame problem*) apareix quan intentem representar els resultats d'accions i esdeveniments. El problema correspon a la necessitat aparent de representar una gran quantitat de fets obvis sobre coses que no canvien quan es fa una acció. Quan un sistema utilitza aquests fets per a fer un raonament, serà poc eficient.

Aprenentatge

Hem comentat abans dues alternatives per a la construcció de la base de coneixements. La primera era que un expert i un enginyer del coneixement construïssin un model que permetés de resoldre els problemes. L'altra alternativa és la de fer servir tècniques d'aprenentatge. Ara passem a comentar aquesta segona possibilitat.

Les tècniques d'aprenentatge permeten construir models d'un domini a partir d'un conjunt d'exemples.

Per exemple, els sistemes d'aprenentatge inductiu formulen hipòtesis que representen el conjunt d'exemples que se li passen. Aquestes formulacions poden ser usades com una part del coneixement d'una base de coneixements.

Exemple d'un sistema d'aprenentatge

Suposem que disposem de cinc registres en un fitxer descrits per quatre variables (v_1 , v_2 , v_3 i v_4), i necessitem expressar la quarta variable en termes de les tres primeres. Aleshores un sistema d'aprenentatge inductiu ens podrà oferir diverses hipòtesis sobre la relació (diferents expressions que ens relacionen les variables d'acord amb els exemples). Per exemple, si tenim els exemples:

Taula 1

v_1	v_2	v_3	v_4
1	1	1	1
1	1	0	1

v1	v2	v3	v4
1	0	1	1
0	0	0	0
0	1	0	0

El sistema podria plantejar que $v4$ és equivalent a $v1$, o que $v4$ és equivalent a $v1 \wedge (v2 \vee v3)$. Aleshores, si ens trobem amb un nou registre amb els valors corresponents a les variables $v1$, $v2$ i $v3$ però sense el de la variable $v4$, podem conjecturar el valor de $v4$ amb alguna de les nostres hipòtesis.

Els sistemes d'aprenentatge poden ser aplicats a la construcció d'un sistema basat en el coneixement. En aquest cas, en lloc de necessitar que un expert ens construeixi un model (ens defineixi una relació entre les variables) podem construir aquest model a partir d'uns exemples.

Per exemple, considereu el cas d'un sistema per a la diagnosi d'un circuit digital que donades tres entrades binàries $v1$, $v2$ i $v3$, ens calcula una sortida també binària $v4$. Aleshores, per a construir el sistema de diagnosi podem demanar a l'expert que ens doni la descripció del circuit (com s'expressa $v4$ en termes de $v1$, $v2$ i $v3$). Tanmateix, si això no és possible, podem fer unes proves inicials amb un circuit que funcioni correctament. Provem què és el que dona per a unes quantes combinacions de valors de $v1$, $v2$ i $v3$. A partir dels resultats construïm un model (com hem fet abans) i fem servir aquest model en el sistema de diagnosi.

Construcció d'un sistema basat en el coneixement

Volem construir un sistema basat en el coneixement per a controlar la temperatura d'un calorímetre. Considerarem la temperatura a la qual volem l'aparell i la temperatura a la qual està en aquest moment. Amb aquesta informació, el nostre sistema ha de decidir si vol escalfar o refredar i quant perquè la temperatura tendeixi cap a la temperatura desitjada. Volem aconseguir arribar a la temperatura desitjada al màxim d'aviat i que hi continuï (que no hi hagi oscil·lacions).

D'acord amb el que hem vist, una manera de fer això és que un expert ens digui què és el que hem de fer en cada situació. L'altra alternativa que podem considerar és utilitzar un mètode d'aprenentatge. En aquest cas, el que fem és demanar a l'expert que ens ensenyi què fa en unes quantes situacions. Li plantejem uns quants objectius: posar el calorímetre a la temperatura A, ara posar-lo a la temperatura B. En aquest cas, si connectem un ordinador al sistema i enregistrem les dades del sistema i del que aplica l'expert, podem aplicar un mètode d'aprenentatge que ens determini què hem de fer en cada situació.

Hem de subratllar que encara que els mètodes d'aprenentatge construeixen models que podem fer servir en sistemes basats en el coneixement, aquests models tenen limitacions. Els sistemes construïts hauran de tenir en compte aquestes limitacions. Un aspecte important a tenir en compte és que els exemples que fem servir en l'aprenentatge són un subconjunt de les situacions en què el sistema es trobarà quan operi realment. Això farà que si el model seleccionat no és del tot adequat, el sistema basat en el coneixement no funcionarà correctament en totes les situacions.

Limitacions dels models

En l'exemple de les quatre variables booleanes, hem considerat dues equivalències per a la variable $v4$. Si el sistema basat en el coneixement pren com a model que la variable $v4$ és equivalent a $v1$, quan el circuit digital dona com a resultat $v4 = 0$ a l'entrada $v1 = 1$, $v2 = 0$ i $v3 = 0$, tindrem que assenyala un error del comportament del sistema. En aquest cas, si haguéssim pres la hipòtesi que la variable $v4$ és igual a $v1 \wedge (v2 \vee v3)$, llavors el model ens diria que el sistema funciona correctament.

Aquí hem parlat de l'aprenentatge sense entrar en detalls dels mètodes existents. De fet, atès que de formalismes de representació del coneixement n'hi ha molts, s'han desenvolupat diversos algorismes d'aprenentatge per a generar hipòtesis en els diferents formalismes. A més, per a un mateix formalisme hi ha algorismes que difereixen en les propietats del model que construeixen. En el mòdul 5 s'introdueixen diverses tècniques d'aprenentatge automàtic que són aplicables en la construcció d'un model basat en coneixement.

Compartir el coneixement

Hem considerat aquí el problema de la definició dels sistemes basats en el coneixement suposant que el coneixement que necessita el sistema és subministrat per un expert o après a partir d'exemples. Hem suposat que per a cada sistema partim de zero amb independència que hi hagin altres aplicacions en el mateix domini.

Aquesta manera, que és freqüent, de definir els sistemes basats en el coneixement, provoca duplicació de la feina i que augmentin els costos de construcció a mesura que els sistemes són més grans.

Amb el nom de compartir el coneixement² s'engloba tots els aspectes relatius al fet de compartir el coneixement entre diferents sistemes de manera que no sigui necessari recodificar-lo quan es comença a construir un nou sistema.

⁽²⁾En anglès, *knowledge sharing*.

Dins aquest camp s'han estudiat els impediments existents per a compartir el coneixement, i es consideren diversos camps d'acció:

1) Mecanismes per a traduir el coneixement formulat mitjançant diferents formalismes de representació: A causa que hi ha moltes famílies de llenguatges de representació, i hi ha molts llenguatges en cada família és impracticable construir traductors per a tots els parells de llenguatges. El nombre de traductors és quadràtic en relació amb el nombre de llenguatges. L'alternativa més adequada és definir un llenguatge intermedi (una *interlingua*) de manera que la representació del coneixement comú (o del coneixement que es comuniquen) es faci per mitjà d'aquest llenguatge. D'aquesta manera el nombre de traductors és lineal en relació amb el nombre de llenguatges.

2) **Especificació de sistemes de representació del coneixement:** Dins una mateixa família de llenguatges hi ha diferències en la semàntica. Aquestes diferències són degudes principalment a la necessitat de resoldre els conflictes entre el poder d'expressió, la completesa en la inferència i els recursos disponibles. Actualment s'estudien definicions d'especificacions comunes a tots els llenguatges d'una mateixa família.

El tema de compartir coneixement no és sols important quan es volen construir diversos agents autònoms (independents) en un mateix domini per no haver de recodificar el coneixement sinó també quan es construeixen sistemes distribuïts que treballaran de manera coordinada i simultàniament. Això és, en el cas de construir sistemes multiagents.

1.1.2. Validació i verificació

Un aspecte lligat als sistemes basats en el coneixement i al procés de construcció és la validació i verificació. Això, com en el cas de l'enginyeria del programari, és comprovar que el programa satisfà les necessitats de l'usuari. Tanmateix, les tècniques que es fan servir en aquella àrea no són útils aquí perquè els dos entorns no són equivalents. Una de les diferències és que els procediments d'inferència d'un sistema basat en el coneixement són simples, però el comportament real depèn del coneixement amagat en la base de coneixement. Això fa que sigui difícil de separar el programari i el model. Una altra de les diferències és que els requeriments d'usuari estan mal definits (l'especificació és incompleta).

Per això, ha calgut desenvolupat eines pròpies. Aquestes eines les podem dividir en dos grups: les que analitzen l'estructura interna del sistema i les que analitzen la seva conducta externa. Entre les primeres hi ha les que es fan servir per a comprovar que la base de coneixements no és inconsistent i que la representació del coneixement no permet que el procés deductiu generi cicles. Entre les segones hi ha la d'analitzar la conducta del sistema mitjançant, per exemple, tècniques de test. Això és, experimentar amb un conjunt de casos quines són les conclusions del sistema. Un cop es tenen les respostes que dona el sistema es compararan amb les solucions correctes o amb aquelles solucions que un expert dona pels mateixos casos.

Vegeu també

Els sistemes multiagents s'estudien amb més profunditat en l'assignatura optativa *Aprentatge computacional*.

Bases de coneixements inconsistent i cíclica

Una base de coneixements inconsistent és aquella en què podem deduir una propietat a i també la seva negació $\neg a$.

Una base de coneixements cíclica és aquella que per a deduir un fet necessita deduir-ne un altre que necessita després el primer. Un exemple és el de les dues implicacions següents $a \leftarrow b$ i $b \leftarrow a$. Per a decidir a necessitem deduir b i per deduir b necessitem a .

2. La representació del coneixement

Una qüestió fonamental en els sistemes basats en el coneixement és la representació del coneixement ja que és el que permet als sistemes saber coses sobre el món que els envolta (sobre el domini d'aplicació). Per això s'han desenvolupat formalismes per a representar la informació que necessiten i eines per a manipular-la. La representació del coneixement és el camp de la intel·ligència artificial que estudia aquests formalismes i eines.

Abans de passar a veure els diferents formalismes, es detallaran alguns conceptes generals d'aquest camp. En primer lloc considerem què és la representació del coneixement i quins factors hi prenen part. Donarem per això dues definicions:

1) La primera ens permet de distingir coneixement d'informació i de dades:

“One usually makes a distinction between “data” and “information”. Data consists of raw figures, measurements, and files that do not necessarily answer the questions that its users may have. Information, on the other hand, is somewhat more refined. It is often the result of processing crude data, giving useful statistics for the data, or answering specific questions posed by users. In AI we usually distinguish a third kind of order in memory: “knowledge”. We think of knowledge as a refined kind of information, often more general than that found in conventional databases. But it may be incomplete or fuzzy as well. We may think of knowledge as a collection of related facts, procedures, models and heuristics that can be used in problem solving or inference systems. Knowledge may be regarded as information in context, as information organized so that it can be readily applied to solving problems, perception and learning.”

S. L. Tanimoto (1987). *The elements of artificial intelligence* (pàg. 89). Computer Science Press.

2) La segona definició ens descriu els factors que cal tenir en compte en la representació del coneixement:

“What is a knowledge representation? We argue that the notion can best be understood in terms of five distinctive roles it plays, each crucial to the task at hand:

- A knowledge representation (KR) is most fundamentally a surrogate, a substitute for the thing itself, used to enable an entity to determine consequences by thinking rather than acting, i.e., by reasoning about the world rather than taking action in it.
- It is a set of ontological commitments, i.e., an answer to the question: In what terms should I think about the world? (...) The commitments are in effect a strong pair of glasses that determine what we can see, bringing some part of the world into sharp focus, at the expense of blurring other parts.
- It is a fragmentary theory of intelligent reasoning, expressed in terms of three components: (i) the representation's fundamental conception of intelligent reasoning; (ii) the set of inferences the representation sanctions; and (iii) the set of inferences it recommends (...).
- It is a medium for pragmatically efficient computation, i.e., the computational environment in which thinking is accomplished. One contribution to this pragmatic efficiency is supplied by the guidance a representation provides for organizing information so as to facilitate making the recommended inferences.
- It is a medium of human expression, i.e., a language in which we say things about the world.”

R. Davis; H. Shrobe; P. Szolovits (1993). “What is a Knowledge Representation?” *AI Magazine* (vol. 1, núm. 14, pàg. 17-33).

Com s’afirma en aquesta darrera definició, la representació (com a eina de càlcul), afecta l’eficiència del sistema (l’espai de cerca es pot reduir: menys nombre d’estats o menys nombre d’operadors) i ens determina allò que es pot inferir (què és el que es pot calcular i què és el que no).

Una determinada representació ens pot simplificar enormement el procés de trobar la solució a un problema o bé ens pot impedir trobar una solució al problema.

El tauler d’escacs retallat

Un exemple ben conegut de que una bona representació pot simplificar el problema és el que plantejem a continuació.

Es considera el tauler de la figura següent i es vol recobrir amb unes fitxes de dòmino que ocupen exactament dues posicions del tauler. El recobriment s’ha de fer de manera que no quedi cap posició buida i sense que hi hagi solapament entre les peces. La qüestió és, té solució aquest problema?

Figura 2

Si sols considerem la forma del tauler i la de les peces, haurem d’anar pensant possibles ubicacions de peces en el tauler fins que el problema estigui resolt o bé que s’hagin provat totes les combinacions i sapiguem que cap recobreix el tauler.

Lectura complementària

Podeu trobar el treball de Davies, Schrobe i Szolovits a l’adreça següent:

<http://www.medg.lcs.mit.edu/ftp/psz/k-rep.html>

Una altra alternativa és considerar també el color de les posicions del tauler. En aquest cas si comptem el nombre de posicions blanques i de posicions negres en el tauler (n'hi ha trenta-dues de blanques i trenta de negres) i es té en compte que una fitxa de dòmino ocuparà sempre una posició blanca i una de negra es fa evident la resposta.

Com s'ha dit abans, en aquest mòdul es descriuran alguns dels formalismes de representació del coneixement existents. En particular, es presentaran les regles i els marcs (com un dels formalismes de representació estructurada). A més d'aquests també podem fer servir algun tipus de lògica. Per exemple, podem utilitzar una lògica de primer ordre. En aquest cas el coneixement vindrà expressat mitjançant fórmules en la lògica triada.

Vegeu també

Per a més detalls sobre els diversos tipus de lògica, vegeu el mòdul de lògica de predicats del'assignatura *Lògica*.

2.1. Nivells d'un formalisme de representació del coneixement

Un formalisme de representació del coneixement es pot veure a tres nivells diferents d'abstracció: de coneixement, lògic i d'implementació.

1) **Nivell de coneixement (o epistemològic):** aquest nivell és el que fa servir un agent per a interaccionar amb el seu formalisme de representació del coneixement. Des d'un punt de vista abstracte ens interessa saber allò que sap (que es pot deduir) sense tenir en compte com està implementat.

2) **Nivell lògic:** en aquest nivell tenim la codificació del coneixement en sentències concretes. Aquí ens interessen les propietats lògiques del formalisme. Això és, quin és el seu significat, i la seva capacitat d'expressió. Per exemple, ens podem plantejar si podem expressar que un determinat fet es compleix per a tots els individus d'una determinada població. Si volem fer servir lògica d'ordre zero, això no es podrà representar perquè no hi ha quantificadors. En canvi, amb la lògica de primer ordre sí que es podrà. Si utilitzem una lògica com un formalisme de representació, també considerarem a aquest nivell les propietats dels procediments d'inferència lògica. Per exemple, si el mètode és sòlid o complet.

Mètodes sòlids i complets

Recordeu que un mètode és sòlid quan només es generen oracions implicades.

Recordeu que un mètode és complet quan es poden deduir totes les conseqüències d'un sistema formal.

3) **Nivell de símbol (o implementació):** en aquest nivell es tracta de com representar el coneixement en un ordinador. Això és, quines estructures de dades són adequades per a representar el que ha de saber el sistema i quins algorismes són adequats per a la inferència. Per exemple, en el cas de la lògica s'hauria de triar una representació pels predicats i una altra per a les fórmules. També caldria implementar els algorismes per a fer la inferència. Per exemple, la resolució.

Aquesta divisió en nivells d'abstracció ens permet considerar els requeriments d'un formalisme de la representació de la mateixa manera.

1) **Nivell de coneixement:** des del punt de vista que el sistema ha d'interaccionar amb l'usuari, ha de ser possible expressar el coneixement en nivells de detall diferents. Per exemple, per a un sistema de raonament general sobre l'univers s'ha de poder representar que els planetes donen voltes al voltant del sol. En un sistema per a físics sobre la mateixa qüestió, serà necessari considerar les equacions dels moviments dels cossos en l'espai.

2) **Nivell lògic:** un requeriment és que en una representació la semàntica (el significat) de les expressions ha de ser clara. Les expressions ben formades han de tenir un significat comprensible. S'aplica aquí l'anomenat *principi de composicionalitat* o *principi de Gottlob Frege* que diu:

“El significat d'una frase o una sentència és una funció del significat de les seves parts i de la manera en què es combinen.”

En el cas de la lògica, una fórmula de la forma $p(a) \rightarrow q(b)$ té significat en termes del significat de $p(a)$, de $q(b)$ i de \rightarrow . Un altre requeriment serà que les conclusions del sistema siguin sòlides en relació amb el coneixement. També es demana que el coneixement es pugui expressar de manera modular, de manera que no hagi de ser considerat tot a la vegada. Això facilitarà les tasques de correcció i actualització de la base de coneixements.

3) **Nivell d'implementació:** els requeriments pel nivell d'implementació corresponen a l'eficiència dels processos. Necessitarem que els mecanismes siguin eficients pel que fa a temps (tant l'accés com la inferència) i en quant a espai. Per exemple, caldrà considerar mecanismes d'indexació (els mecanismes que permeten de trobar un predicat en la base de coneixements) que siguin eficients per tal taules de dispersió³ o arbres (binaris) de cerca.

⁽³⁾En anglès, *hash tables*.

A banda d'aquests requeriments, també haurem de tenir en compte que hi hagi bona sintonia entre els nivells. Per exemple, que les propietats que exigim per al nivell lògic la implementació les prevegi.

El nombre de nivells d'un sistema de representació del coneixement

El nombre de nivells d'un sistema de representació del coneixement i de quins són aquests nivells no és una qüestió resolta. No tots els autors coincideixen. Per exemple, hi ha qui seguint Newell (A. Newell (1982). “The Knowledge Level”. *Artificial Intelligence* (núm. 18, pàg. 87-127) considera només el nivell de coneixement i el nivell de símbol. En relació amb el nivell lògic considerat aquí Uschold afirma:

“It is not clear how to view the logical level. From a programming perspective, it seems far from implementation concerns, and thus at the knowledge level. However, from the perspective of building a knowledge base, it is not unreasonable to view a formal representation at the logical level as an implementation of structures identified at the conceptual, ontological, and epistemological levels.”

M. Uschold (1998). *Knowledge level modelling: concepts and terminology*.

2.2. Aspecte formal i aspecte inferencial

En tot formalisme de representació del coneixement es poden distingir dos aspectes. Un aspecte formal i un aspecte inferencial.

L'aspecte formal correspon a com s'emmagatzema la informació quan la tenim representada de manera explícita: quin és el formalisme que es fa servir.

De fet, en aquest aspecte hi cauen totes les qüestions relacionades amb la sintaxi i la semàntica del formalisme.

L'aspecte inferencial correspon a com s'obté la informació que es troba en el sistema però només de manera implícita.

En aquest cas, hi haurà un seguit de regles abstractes (que no dependran del coneixement emmagatzemat) que permetran obtenir la informació a partir d'aquella que és explícita. És important subratllar que encara que el terme *inferencial* és tret de la lògica, no ens restringim a aquests tipus de sistemes.

En el cas de la lògica tindrem que l'aspecte formal correspon a com es formen les expressions: quines són les expressions primitives (el vocabulari: constants, variables, predicats) i com es formen les fórmules. L'aspecte inferencial tractarà dels mètodes que permeten demostrar un predicat com, per exemple, la deducció natural o la resolució.

Aquests dos aspectes corresponen aproximadament a dos dels mòduls que haviem aparegut en l'arquitectura de la figura 1: la base de coneixements i el subsistema d'inferència. En descriure l'arquitectura de la figura s'ha comentat que aquesta divisió en mòduls no sempre és possible a causa de l'estreta relació que hi ha entre la base de coneixements i els procediments d'inferència. Evidentment, aquesta relació també es dona aquí entre els dos aspectes d'un formalisme de representació. L'aspecte formal defineix les expressions primitives i com es construeixen les expressions complexes a partir d'aquestes. Sobre la base d'aquestes primitives i de com es construeixen les expressions complexes es definirà la inferència.

De tota manera, és important diferenciar entre els dos aspectes perquè a l'hora de comparar mecanismes de representació del coneixement cada aspecte té associades propietats diferents. Així, alguns dels requeriments considerats en el nivell lògic es poden dividir per aspectes. Per exemple, l'expressivitat recaurà sobre l'aspecte formal i la solidesa (o la completesa) sobre el que és inferencial.

2.3. Tipus de coneixement

Un sistema pot incloure diferents tipus de coneixement. Una primera divisió és la que ens permet distingir entre el coneixement de domini i el coneixement estratègic. Aquests tipus de coneixement els podem formular així:

a) **Coneixement de domini:** el que correspon al coneixement general sobre el camp d'aplicació del sistema.

b) **Coneixement estratègic:** el que descriu com fer servir el coneixement de domini per a resoldre un determinat problema.

En un sistema de diagnosi mèdica el coneixement de domini correspondrà a tot allò relatiu a les malalties que tracta el sistema i el coneixement estratègic a com s'ha d'enfocar el cas concret d'un malalt per a trobar una diagnosi per a la seva malaltia.

Una altra divisió és la que ens distingeix entre coneixement declaratiu i coneixement procedimental.

a) **El coneixement procedimental:** és el que està lligat a una utilització concreta. Per exemple, podem definir una funció per a calcular el factorial d'un nombre. Vista la funció com un coneixement fet explícit, tenim que aquest coneixement només es pot fer servir per a calcular el factorial d'un nombre. La funció no ens permet de saber quin és el nombre que té com a factorial el 150, o si el factorial de 3 és 54.

b) **El coneixement declaratiu:** és aquell que és independent del seu ús. Si expressem la relació entre un nombre i el seu factorial mitjançant lògica proposicional, aleshores podem utilitzar el coneixement de diverses maneres. En particular, podem raonar sobre quin nombre té com a factorial el 150 i si el factorial de 3 és 54. En aquest cas, el coneixement ens permet de resoldre diferents tipus de preguntes. Algunes de les quals potser no havien estat considerades en el disseny inicial de la base de coneixements.

PROLOG

El llenguatge lògic PROLOG, que es diu declaratiu, ens permet definir el factorial de manera que totes aquestes qüestions es poden formulari resoldre.

Exemple de coneixement declaratiu

Un exemple d'aquest tipus de coneixement el tenim quan considerem el model de l'exemple d'aprenentatge del subapartat 1.1.1. Si tenim com a model $v4 = v1 \wedge (v2 \vee v3)$ podem raonar sobre el valor de $v4$ segons el valor de les altres variables, però també podem raonar sobre el valor de les altres variables quan sabem el de $v4$. Per exemple, si sabem que $v4$ és zero, $v1$ és 1, aleshores podem deduir que tant $v2$ com $v3$ han de ser zero.

Si en lloc d'aquest model tenim una implementació de la funció lògica en un llenguatge concret això no serà possible. Haurem d'aplicar la funció tal com l'hem dissenyada.

Coneixement declaratiu:

$$v4 = v1 \wedge (v2 \vee v3)$$

Coneixement estratègic:

```
funcio expressió (v1: boolea; v2; boolea; v3: boolea) retorna boolea es
    retorna (v1 and (v2 or v3))
ffuncio.
```

2.4. L'aspecte inferencial i la cerca

Com s'ha dit, un dels aspectes d'un sistema de representació del coneixement és l'aspecte inferencial.

En la implementació dels mecanismes d'inferència apareixen problemes de cerca. Això és així perquè la inferència es farà per a complir un objectiu i en el procés de complir-lo se'ns presentaran sovint diverses alternatives de les quals n'haurèm de triar una. En l'entorn de la representació del coneixement al problema de la cerca se li diu a vegades el problema del control.

Per exemple, en el cas de fer servir la lògica d'enunciats com a eina de representació del coneixement i la deducció natural com a forma d'inferència, tindrem que l'objectiu pot ser la demostració d'un enunciat. Per a fer la demostració farem un conjunt de passos, i en cada pas seleccionarem una regla d'entre les que podem aplicar. Com en tot problema de cerca, no totes les alternatives que podem aplicar seran bones. Hi haurà regles que ens portaran cap a la demostració de l'enunciat i n'hi haurà que no. Aquí interessarà, com en els problemes de cerca, seleccionar les regles que ens porten cap a l'objectiu al més aviat possible. Més endavant veurem que això també passa en els altres mecanismes de representació.

S'ha dit abans que el coneixement estratègic descriu com fer servir el coneixement de domini per a resoldre un determinat problema. Això correspon, de fet, a guiar la cerca per tal que ens acostem cap a l'objectiu el més ràpidament possible.

A vegades aquest coneixement es trobarà de manera implícita en l'aspecte inferencial del formalisme de representació del coneixement, però de vegades apareix de manera explícita. Per exemple, en el cas de la lògica i la deducció natural podem tenir una implementació que per a decidir quina regla s'ha d'aplicar se segueixi un esquema prefixat. Per exemple, primer es prova d'introduir una conjunció, després si això no va bé intenta d'introduir una disjunció... Això correspon a un coneixement estratègic expressat de manera procedimental perquè per a canviar com es fa la tria s'ha de reescriure de nou el mecanisme d'inferència. Una altra alternativa és que el sistema a l'hora de decidir quina regla aplica faci servir un conjunt de recomanacions de manera explícita. Per exemple, recomanacions en la línia dels consells pràctics que es donen a l'assignatura de lògica per a demostrar un enunciat:

Vegeu també

Vegeu els conceptes de *resolució de problemes* i de *cerca* en el mòdul "Resolució de problemes i cerca" d'aquesta assignatura.

Els diferent tipus de regles

Recordem que hi ha dos tipus de regles en la lògica d'enunciats: les regles d'introducció i les regles d'eliminació. Per exemple, regla d'introducció de la conjunció, de la disjunció... i regles d'eliminació de la conjunció, de la disjunció...

PROLOG

El llenguatge PROLOG, vist des d'aquest punt de vista, utilitza un esquema prefixat o la forma de realitzar la inferència segueix sempre un mateix esquema.

Vegeu també

Vegeu els consells pràctics que es donen en el mòdul "Lògica d'enunciats" de l'assignatura Lògica.

Consell 4.a) Si en les premisses apareixen implicacions, pot anar bé que intenteu d'obtenir-ne els conseqüents aplicant la regla d'eliminació de la implicació. Per a això pot ser necessari resoldre abans el subproblema d'obtenir-ne l'antecedent.

Consell 4.b) Si en les premisses apareixen disjuncions, pot ser escaient que plantegeu tot el problema com una prova per casos, aplicant la regla d'eliminació de la disjunció.

En aquest cas un canvi en les recomanacions provocarà un canvi en la forma de selecció de les regles, sense necessitar de canviar el mecanisme d'inferència.

El fet de tenir el coneixement estratègic de manera explícita presenta alguns avantatges:

a) Es pot fer servir coneixement estratègic específic d'un domini, de manera que en canviar el domini podem canviar la forma d'atacar un problema. Per exemple, en diagnosi mèdica podem considerar que si una malaltia és més freqüent que una altra comencem mirant la malaltia més freqüent. Si canviem a un problema de diagnosi d'avaries de cotxes, podem fer servir unes altres heurístiques. Quan el coneixement estratègic és explícit podem canviar de domini sense canviar el mecanisme d'inferència: n'hi haurà prou de canviar aquest coneixement.

b) Es pot reutilitzar el coneixement de domini en aplicacions diferents. Quan canvia l'aplicació però el domini es manté, canviarà la part del coneixement estratègic. El coneixement del domini es pot mantenir. Per exemple, un sistema de diagnosi mèdica i un sistema per a ajudar a què els estudiants de medicina aprenguin sobre els símptomes d'una malaltia pot fer servir el mateix coneixement sobre les relacions símptomes-malalties. La manera de fer les deduccions hauran de ser diferents en deduir les primeres sobre malalties i les segones sobre símptomes.

c) Un dels usos dels sistemes basats en el coneixement és donar a més de la conclusió una explicació del perquè aquella solució és la seleccionada. El coneixement estratègic ajuda en la construcció de l'explicació. Ens diu com hem de fer les seleccions.

2.5. Altres aspectes sobre els formalismes de representació del coneixement

El desenvolupament de sistemes basats en el coneixement ha obligat a desenvolupar mètodes de raonament per a tractar els diferents tipus de situacions en què es poden trobar. Per exemple, alguns sistemes han de poder raonar sobre el temps (sobre processos que s'allarguen en el temps, que són anteriors o simultanis); d'altres han de tractar amb falta de coneixement (no es coneix tota la informació que seria desitjable per a resoldre un problema). També n'hi ha que han de tenir en compte el que saben els altres per tal de prendre decisions. A continuació es repassen algunes tècniques que són adequades per alguns tipus de problemes:

a) Raonament temporal: aquestes tècniques són adequades quan un sistema ha de considerar processos amb una extensió temporal, i ha de ser capaç de raonar sobre aquests. Per a tractar aquests aspectes hi ha en l'actualitat diferents formalismes. Alguns dels problemes a què s'adrecen és la necessitat de representar informació temporal parcial o de raonar sobre possibles passats o futurs.

b) Raonament espacial: l'espai és un altre aspecte que és necessari considerar en alguns sistemes. Per exemple, han de poder raonar sobre la posició dels objectes o la forma. Alguns han de permetre de tractar representacions a diferent nivells.

c) Metaraonament: a vegades no n'hi ha prou amb raonar sobre aspectes externs, sinó que és també necessari raonar sobre el coneixement. Tant sobre el propi coneixement com sobre el coneixement que tenen els altres. També han de tenir en compte l'anomenat *coneixement comú* (el coneixement que comparteixen de manera implícita un grup d'agents). La lògica epistèmica tracta aquest tipus de raonament.

d) Raonament amb informació incompleta: la informació desitjable per a resoldre un problema no està sempre disponible. Hi ha vegades en què alguns fets es desconeixen però tot i així és necessari raonar i prendre decisions. Per a tractar aquestes situacions s'han desenvolupat diverses eines. Podem subratllar els mètodes de raonament monòton, de raonament sota incertesa i de raonament aproximat. Els mètodes de raonament sota incertesa i de raonament aproximat són explicats amb detall en el mòdul IV d'aquesta assignatura.

Els mètodes de raonament no monòton permeten fer inferències no sòlides (inferències que extreuen coneixement nou) a partir del coneixement incorporat en el sistema. Aquest coneixement s'expressa utilitzant regles no estàndard, per exemple, les regles per defecte⁴. Aquest tipus es diu *no monòton* perquè la incorporació de nou coneixement en el sistema no implica que es pugui deduir més que no es podia abans. Recordem que en la lògica quan s'incorporen nous fets sempre es poden obtenir noves conseqüències (que s'afegeixen a les anteriors). Per tant, existeix monotonia a l'hora d'afegir informació en el sistema.

⁽⁴⁾En anglès, *default rules*.

En els mètodes que fem servir regles per defecte això no és d'aquesta manera perquè es poden haver aplicat inferències no sòlides. Aleshores quan s'obté una nova informació pot ser que aquesta estigui en contradicció amb les suposicions que s'han fet. Per tant, les conclusions obtingudes prèviament s'hauran de revisar. És per aquesta raó que s'anomena *raonament no monòton*. Els mecanismes de raonament no monòton estan explícits en alguns dels llenguatges de programació declaratius per a la programació lògica i de bases de dades deductives.

La necessitat que la informació d'un sistema sigui consistent encara que la inferència no sigui sòlida o s'utilitzin regles per defecte va portar al disseny dels anomenats *sistemes de manteniment de la veritat* (TMS)⁵. Aquests sistemes tenen constància dels passos de deducció i quan algunes proposicions deixen de ser vàlides les abandonen. Cal tenir en compte que si un sistema ha deduït A a partir d'una regla per defecte, i en un pas posterior hem deduït B a partir de A i d'altra informació, quan nou coneixement ens fa reconsiderar A no n'hi ha prou amb retirar A del conjunt de fets coneguts sinó que també haurem de retirar B. Un sistema de manteniment de la veritat en retirar A també retiraria B.

⁽⁵⁾TMS és la sigla de l'expressió anglesa *Truth Maintenance Systems*.

El problema del sentit comú

Es diu que els sistemes basats en el coneixement són fràgils perquè fóra del seu domini d'aplicació no funcionen bé i a més no saben quina és la frontera del seu coneixement. Això és, que no poden detectar quan una situació està fora del coneixement que tenen incorporat i que per tant les seves inferències no són útils. Això també provoca l'anomenada *manca de sentit comú*: en algunes ocasions les respostes del sistema són absurdes perquè falta coneixement que en podríem dir elemental. Fets i relacions que tothom té assumits: falta de sentit comú.

Hi ha investigadors que creuen que aquest problema es pot resoldre fent bases de coneixement més grans. La construcció de bases de coneixement a gran escala segueix aquesta línia per aconseguir comportaments intel·ligents.

Aquest és el cas del projecte CYC que dirigeix en Lenat. El sistema pretén de capturar tot el coneixement –tant explícit com implícit– que apareix en un conjunt d'entrades de l'enciclopèdia Britànica. L'interès principal rau a formular la informació subjacent que els autors dels articles suposen als seus lectors. Per a aconseguir-ho, al final del 1997 s'havien entrat ja més de 4.000.000 assercions. S'espera que la formulació del coneixement implícit permeti al sistema llegir la resta de l'enciclopèdia i capturar el sentit sense intervenció humana. CYC és un projecte ambiciós, que quan va començar (l'any 1984) havia de durar deu anys i costar cinquanta milions de dòlars. Tanmateix, encara avui no es compleixen els objectius marcats a l'inici del projecte.

Els resultats obtinguts fins avui han fet que aquesta aproximació a la intel·ligència artificial hagi estat fortament criticada. Aquesta crítica ha vingut especialment d'entre aquells que no creuen que la intel·ligència artificial es pugui aconseguir basant-se únicament amb la hipòtesi de Newell i Simon.

Lectura complementària

Trobareu informació sobre el projecte CYC a l'adreça següent:

<http://www.cyc.com/>

Vegeu també

Vegeu la hipòtesi de Newell i Simon en l'apartat 2 del mòdul "Què és la intel·ligència artificial" d'aquesta assignatura.

De tota manera, amb independència de la utilitat pràctica real de CYC, el projecte ha hagut d'encarar els problemes que apareixen en qualsevol sistema basat en el coneixement que fa servir bases de coneixement de grans dimensions. En particular, s'han hagut de trobar algorismes d'inferència eficients, i trobar maneres de fer manteniment i actualització de la base de coneixements.

3. Sistemes basats en regles

Els sistemes experts o sistemes basats en regles són un dels mètodes més coneguts de sistemes basats en el coneixement i són la base de moltes de les aplicacions existents. S'apliquen sobretot a problemes de classificació (identificació) i diagnosi.

En aquests sistemes el coneixement ve representat mitjançant fets i regles. Els fets es corresponen a coneixement sobre els objectes, i les regles permeten establir relacions entre aquests objectes. Les regles tenen la forma:

```
si <premissa> llavors <conclusió>
```

on tant la premissa com la conclusió són afirmacions sobre fets. A partir dels fets i de les regles, el sistema d'inferència conclourà sobre nous fets (o corroborarà la certesa d'uns fets). Les regles sovint inclouen elements per a representar la incertesa sobre les relacions.

Com exemple de regles, a continuació se'n donen dues del sistema Spong-ia per a classificació d'esponges marines.

El sistema Spong-ia

El sistema Spong-ia es descriu a l'obra següent:

M. Domingo (1995). "An Expert System Architecture for Identification in Biology". Monografies de l'IIIA.

És un sistema que serveix per a classificar esponges marines. Això és, donat un exemplar determinar a quina classe pertany.

```
R002 If skeleton/pres and skeleton/chemical = (silica)
    then conclude Demospongiae is possible
R003 If skeleton/fibre
    then conclude Demospongiae is sure
```

En aquestes regles, la conclusió utilitza els termes *possible* i *sure* per a representar la incertesa que hi ha entre els antecedents i la conclusió:

Figura 3

Els sistemes basats en regles tenen una arquitectura molt semblant a la donada en la figura 1 com a general per a tots els sistemes basats en el coneixement (compareu l'arquitectura de la figura 3 amb la de la figura 1). Disposen d'una base de coneixement formada per una base de regles i una memòria de treball, a més d'un intèrpret que correspon al sistema d'inferència. A continuació es detallen una mica més aquests elements.

a) Base de regles: la base de regles correspon a una part del que hem anomenat *l'aspecte formal del formalisme de representació del coneixement*. S'hi guarda tota la informació general relativa al domini d'aplicació. Per tant, hi haurà totes les regles necessàries per a resoldre els problemes que tracta el sistema. En el cas de la classificació d'esponges marines, hi tindrem totes les regles que relacionen les característiques de les esponges, i les que ens permeten concloure la seva classificació. La informació de la base de regles és independent de l'aplicació del sistema a casos concrets.

b) Memòria de treball: correspon a l'altra part de l'aspecte formal dels sistemes basats en regles. S'inclouen aquí tots aquells fets que poden ser rellevants pel problema concret que es vol resoldre. A diferència del contingut de la base de regles, aquí la informació és temporal i correspon al problema concret que es vol resoldre. En el cas de la classificació d'esponges, tindrem fets (informació) de l'esponja que estem classificant en un moment donat (per exemple, dades sobre l'esquelet).

c) Intèrpret: això correspon a l'aspecte inferencial del sistema. Permet deduir nous fets a partir de la informació existent en la memòria de treball i de les regles de la base de regles.

En aquesta descripció no hem inclòs ni el mòdul d'explicació ni el d'adquisició del coneixement tot i que a vegades aquests sistemes també els inclouen. En aquest cas, la funció d'aquests mòduls és la mateixa que la descrita en l'apartat 1 d'aquest mòdul. A continuació descrivim amb una mica de detall el procediment d'inferència en un sistema basat en regles.

3.1. Aspecte inferencial en un sistema basat en regles

Com s'ha dit, l'intèrpret correspon a l'aspecte inferencial d'un sistema basat en regles. Molt sovint, la inferència s'aconsegueix mitjançant un esquema cíclic de recuperació –refinament– execució. Detallem a continuació cadascun d'aquests passos:

a) **Recuperació:** primer se seleccionen de la base de regles aquelles regles que sobre la base de la informació de què es disposa, se sap que es poden aplicar. Sovint en aquesta fase no s'obté una única regla sinó que se n'obté un conjunt. Al conjunt de regles seleccionades se l'anomena *conjunt de conflicte*⁶.

b) **Refinament:** atès que els sistemes basats en regles acostumen a aplicar només una regla cada vegada, s'haurà de seleccionar una regla del conjunt.

c) **Execució:** un cop s'ha seleccionat la regla aquesta s'aplica.

Un cop s'ha executat una regla, es passarà un altre cop a la recuperació. El cicle acabarà un cop s'hagi pogut demostrar allò que es volia o bé el conjunt de conflicte sigui buit (per tant no hi ha cap regla que es pugui aplicar). Observeu que si s'arriba a un punt amb un conjunt de conflicte buit, com que no hi haurà regla per aplicar, no hi haurà cap canvi en la memòria de treball. Si no hi ha cap canvi en la memòria de treball, el proper conjunt de conflicte serà també buit. Per tant, d'una situació amb un conjunt de conflicte buit no es pot sortir.

La descripció del funcionament del sistema que hem presentat presenta el problema del control ja comentat anteriorment. A cada pas del procés hem de triar entre les diferents regles que es poden aplicar. Així, en general tenim un objectiu (en l'exemple que hem considerat, determinar la classe a la qual pertany una esponja) i un problema de cerca ja que només un subconjunt de les regles (i aplicades en un cert ordre) ens porta a la solució.

Per a ajudar i dirigir la cerca, els sistemes consideren diferents estratègies. Algunes d'aquestes són les següents:

a) **Ordre textual:** l'ordre en què apareixen les regles en la base de regles determina l'ordre d'aplicació. En aquest cas, podem controlar l'estratègia de la cerca ordenant les regles de manera adequada. De fet, el llenguatge PROLOG es pot veure com un sistema que segueix aquesta estratègia perquè allò que hem definit primer en el programa s'aplica primer.

b) **Obstinància**⁷: impedeix que una regla s'apliqui més d'una vegada. Quan una regla ja ha estat aplicada queda anul·lada. La idea darrere l'obstinància és que cada vegada que s'aplica una regla a uns mateixos fets s'obtenen les mateixes conclusions. S'ha d'anar alerta amb l'ús d'aquesta estratègia perquè quan una regla té variables la seva aplicació pot donar diferents resultats atès que podem instanciar-la amb objectes diferents. Això és semblant al que passa amb la lògica proposicional. Si tenim $p(a)$, $p(b)$, $\forall x(p(x) \rightarrow q(x))$ podem obtenir tant $q(a)$ com $q(b)$. En aquest cas per a obtenir $q(a)$ i $q(b)$ hem de considerar

⁽⁶⁾En anglès, *conflict set*.

Els sistemes difusos

No sempre els sistemes basats en regles apliquen una sola regla cada vegada; els sistemes difusos acostumen a aplicar totes les regles a la vegada (els sistemes difusos s'estudien en el mòdul Incertesa i raonament aproximat).

⁽⁷⁾En anglès, *refractoriness*.

$\forall x(p(x) \rightarrow q(x))$ dues vegades, una amb $p(a)$ i l'altra amb $p(b)$. D'altra banda, si el coneixement no és monòton aquesta estratègia també pot provocar problemes.

c) **Recència:** se seleccionen les regles que es refereixen als fets més recents de la memòria de treball. Per a implementar aquesta estratègia és necessari que els predicats de la base de fets incloguin una marca de temps⁸ per a poder resoldre els conflictes. Quan hi ha diverses regles que es poden aplicar, es miren les marques de temps dels predicats que apareixen en l'antecedent. Se selecciona la regla on els predicats tenen la darrera marca de temps.

⁽⁸⁾En anglès, *time tag*.

d) **Especificitat:** se seleccionen les regles que corresponen a objectes més específics. Aquí es considera un objecte A més específic que B quan A és un tipus dels objectes B (per exemple, fariem servir abans una regla referida a *cotxe* que una de més general sobre *vehicles* quan les dues coses són aplicables) o bé A és un element (un exemple particular) de B (per exemple, abans farem servir una regla referida al *cotxe de l'alcalde* que una sobre *cotxes* si les dues són aplicables). També es poden seleccionar aquelles regles més específiques. Una de les maneres d'avaluar l'especificitat de les regles és comptant el nombre de conjuntands (predicats en l'antecedent que apareixen units amb una conjunció) de la regla. Com més conjuntands més específica serà la regla. Així, una regla de la forma "si el terra està mullat i no plou llavors ..." s'aplicarà abans que una de la forma "si el terra està mullat llavors ...".

Una alternativa diferent per a controlar la cerca i fer que sigui més eficient és fer servir coneixement estratègic. Per exemple, el llenguatge Milord-II per a desenvolupament de sistemes experts utilitza metaregles (regles de segon nivell) per a dirigir la cerca.

El llenguatge Milord-II

Milord-II és un llenguatge per a la construcció de sistemes experts desenvolupat a l'institut d'investigació en intel·ligència artificial (<http://www.iiia.csic.es/~puyol/MilordII/>).

A banda del problema de seleccionar quina és la regla que cal aplicar en un moment donat, hi ha altres qüestions relacionades amb la manera de fer la inferència. Una d'aquestes és que aquesta es pot fer dirigida per les dades o bé dirigida pels objectius. En el primer cas les regles se seleccionen i s'apliquen tenint en compte allò que sabem. En el segon cas, l'èmfasi és en allò que volem deduir. En el cas de sistemes basats en regles, això correspon al que s'anomena *raonament cap endavant* i *raonament cap endarrere*. Aquesta nomenclatura es basa en la idea que les regles tenen un sentit: dels antecedents a les conclusions. A continuació detallem aquests dos tipus de raonament:

1) **Raonament cap endavant:** partim de la informació disponible (el coneixement que tenim sobre l'objecte que volem classificar o diagnosticar) i anem aplicant regles fins que aconseguim deduir allò que es vol deduir (la classe o la diagnosi corresponent). Es diu de *raonament cap endavant* perquè el que mirem per saber si una regla es pot aplicar és si l'antecedent es compleix. Si es compleix afegirem a la memòria de treball les conclusions de la regla. Així anem dels antecedents de les regles a les conclusions.

Així, per exemple, imaginem un sistema en què, segons el que hem comentat anteriorment, podem definir una base de regles que estigui formada per deu regles:

- R1. $A \wedge B \rightarrow E$
- R2. $A \wedge C \rightarrow F$
- R3. $\neg A \wedge B \rightarrow D$
- R4. $\neg A \wedge \neg B \rightarrow E$
- R5. $B \wedge C \rightarrow F$
- R6. $\neg B \wedge \neg C \rightarrow \neg F$
- R7. $D \wedge \neg F \rightarrow H$
- R8. $D \wedge F \rightarrow G$
- R9. $E \wedge F \rightarrow H$
- R10. $E \wedge \neg F \rightarrow G$

D'aquest sistema coneixem la memòria de treball inicial que està formada pels fets A, B i C i es vol esbrinar si amb aquesta informació es pot demostrar F . A fi d'ajudar en el procés d'inferència, se sap que les regles s'ordenen segons la seva numeració. Durant cada etapa realitzada per l'interpret, les regles que s'han utilitzat s'eliminen de la base de regles (el que es coneix com a obstinància). Si hi ha més d'una regla seleccionada, s'executarà la de la numeració més baixa.

Si per a fer la inferència el sistema aplica un raonament cap endavant, llavors ens trobarem que partim d'una memòria de treball inicial: $MT0 = \{A, B, C\}$. Aquests fets, que en aquest moment són certs, fa que esdevinguin aplicables tres regles (R1, R2, R5) en la base de regles. Això genera el conjunt conflicte inicial $CC0 = \{R1, R2, R5\}$. Per a resoldre aquest conflicte en aquest exemple, s'han establert diferents eines: les regles s'eliminen de la base de regles un cop aplicades i s'aplica inicialment la de la numeració més baixa. Per aquest motiu, se selecciona R1 i s'elimina de la base de regles. En el cicle d'inferència següent, la memòria de treball, un cop aplicada la regla 1, esdevé $MT1 = \{A, B, C, E\}$ i, per tant, les regles aplicables són R2 i R5, generant un nou conjunt conflicte $CC1 = \{R2, R5\}$. Aplicant la mateixa estratègia de resolució de conflictes, se selecciona la regla R2 i s'elimina de la base de regles aplicables. En el tercer cicle d'inferència, la memòria de treball consta dels següents fets $MT2 = \{A, B, C, E, F\}$ i, per tant, el nou conjunt conflicte és $CC2 = \{R5, R9\}$. Se selecciona R5 i s'elimina de la base de regles. Com que es volia demostrar F , en aquest moment, en mirar la memòria de treball el sistema s'aturaria.

Nota

Podem practicar amb l'exercici 1 d'autoavaluació.

2) Raonament cap endarrere: en aquest cas es parteix de l'objectiu (per exemple, d'una hipòtesi sobre la classe a la qual pertany un objecte o de la seva diagnosi) i seleccionem les regles que conclouen sobre aquest objectiu. La idea és que allò que hi ha a la conclusió serà cert si es compleixen les condicions que apareixen en l'antecedent d'una regla. Així, per a poder demostrar la conclusió hem de demostrar els antecedents de les regles. Aleshores es torna a re-

petir el mateix procés plantejant nous objectius (els antecedents de les regles anteriors). D'aquesta manera considerem, de fet, les regles de les conclusions als antecedents.

Imaginem que a partir del sistema de regles de l'exemple anterior volguéssim demostrar el fet G , aplicant el raonament cap enrere en el procés d'inferència si, en aquest cas, la base de fets inicials fos $\{\neg A, \neg B, \neg C\}$.

En aquest cas, per a demostrar G , seleccionem dues regles que conclouen aquest objectiu: $R8$ i $R10$. Si seguim utilitzant les mateixes eines per a ajudar en la cerca, és a dir, que les regles s'apliquen per la numeració i que, un cop aplicades, s'eliminen de la base de regles, seleccionariem la regla $R8$ i per tant, com a nous subobjectius, hauríem de poder incloure a la base de fets D i F . Per a incloure D , haurem d'aplicar $R3$, però en la nostra base de fets hi tenim $\neg B$ i això faria que si $\neg B$ és cert, B no pugui ser-ho. De manera que, el sistema repetiria el procés però ara aplicant $R10$. Per a concloure G , hauríem de poder afegir a la base de fet E i $\neg F$, que passen a ser els nostres nous subobjectius. Per a demostrar E , podem aplicar $R1$ i $R4$. $R1$ no es compleix ja que la nostra base de fets inicials conté $\neg A$ i $\neg B$, fet que invalida l'aplicació d' $R1$. En canvi, activa $R4$ que incorporaria a la base de fets $\{\neg A, \neg B, \neg C, E\}$. Finalment, per a demostrar l'altre subobjectiu $\neg F$, cal veure si $R6$ és aplicable, que en tenir $\neg B$ i $\neg C$ a la base de fets si que és aplicable i permet demostrar $\neg F$.

Nota

Podeu practicar amb l'exercici 1 d'autoavaluació.

No tots els sistemes permeten els dos tipus d'encadenament. En general, si podem escollir, la tria s'haurà de fer sobre la base de la informació de què disposem i, com en qualsevol problema de cerca, quina de les dues opcions té un factor de ramificació més petit. El control estratègic permet en alguns casos de seleccionar entre les dues opcions.

3.2. Anàlisi dels sistemes basats en regles

Les regles són una de les formes de representació del coneixement més usades per a construir sistemes basats en el coneixement. Algunes de les raons que s'apunten per aquest èxit és que són una manera senzilla de representar allò que l'expert utilitza per a resoldre un problema. Per exemple, les relacions entre símptomes i diagnosi, i, en el cas particular de la medicina, les relacions entre símptomes i malalties.

Un altre avantatge dels sistemes basats en regles és la seva modularitat. El fet que la base de coneixements sigui independent del procés d'inferència permet modificar la base de dades sense haver de tocar res relatiu a l'aspecte inferencial. A més, el coneixement temporal (la informació que està emmagatzemada en la memòria de treball) és independent de la permanent (la que està representada en les regles).

També apunta en aquest sentit l'organització modular que alguns llenguatges de sistemes experts⁹ permeten per a les regles. Les regles es poden agrupar en mòduls. Això facilita la implementació dels sistemes i el posterior refinament o actualització. Aquesta modularització té un avantatge afegit que és que facilita els accessos a la informació i a la inferència. En aplicar el procés a un mòdul, s'augmenta l'eficiència de la recuperació i es disminueix el conjunt de conflicte.

⁹El llenguatge Milord-II esmentat abans permet aquesta organització.

Les regles tenen una sintaxi restringida. Això fa que la capacitat expressiva sigui limitada i que, per tant, hi hagi coses que no es puguin expressar. Tanmateix, aquesta restricció en l'estructura del coneixement permet l'existència de programes que operen amb regles. Per exemple, hi ha programes per a tractar la validesa d'un conjunt de regles (per a veure si és consistent o si conté cicles) o per a construir automàticament regles a partir d'exemples.

Un altre avantatge és la facilitat per a construir una explicació del perquè les conclusions es deriven dels fets, atès que els sistemes experts construeixen un camí (la llista de regles activades) que lliga allò que volem demostrar amb la informació de què disposem. Un cop construït el camí és possible reconstruir el raonament aplicat, i explicar-lo.

De tota manera, no sempre n'hi ha prou amb la informació que donen les regles per a construir l'explicació. A vegades també és rellevant el perquè s'ha considerat una hipòtesi abans que l'altre o perquè no se n'ha considerat cap tercera. En aquests casos, a més de considerar el coneixement de domini, va bé el coneixement estratègic per a construir les explicacions. Per tant, convindrà que aquest tipus de coneixement estigui representat de manera explícita.

Per a poder construir les explicacions, a més dels coneixement de domini i de control, necessitem el que se'n diu *coneixement de suport*.

Aquest coneixement explica el perquè les regles són bones. A més, permet raonar sobre per què es prenen unes determinades decisions. Això facilita que en determinades condicions una regla es pugui saltar.

Exemple d'explicació d'una regla

Per exemple, podem considerar la regla següent del sistema MYCIN:

```
si el pacient té menys de 8 anys llavors no recomanar tetracina
```

L'explicació d'aquesta regla està en el fet que aplicar la tetracina a infants que els hi creixen els ossos els hi enfosqueix les dents. Evidentment, en casos de necessitat aquesta regla es podria anul·lar. Si el sistema coneix el perquè de la regla, la podrà anul·lar.

4. Sistemes amb representació estructurada

En els formalismes de representació considerats fins ara el coneixement s'estructura en unitats independents (predicats en la lògica de primer ordre i regles en els sistemes de regles). No hi ha lligams entre les diferents unitats que defineixen la base de coneixement.

Els sistemes de marcs¹⁰, les xarxes semàntiques¹¹ i els guions¹² són formalismes de representació alternatius que permeten relacionar les unitats a partir de les quals es defineix el coneixement. Aquestes relacions corresponen a una modelització de les interconnexions entre els objectes del domini.

⁽¹⁰⁾En anglès, *frame*.

⁽¹¹⁾En anglès, *semantic networks*.

⁽¹²⁾En anglès, *scripts*.

Les diferències entre els tres tipus de sistemes són poques i la frontera entre si no és clara. Tots tres tenen una estructura semblant: uns conceptes i relacions entre si. Mentre que en el cas de les xarxes semàntiques se subratllen les relacions, en els marcs es dóna més importància als conceptes. Per altra banda, els guions corresponen a representacions de seqüències d'esdeveniments (per a ser utilitzats principalment per a la comprensió d'històries).

Exemple de guió

En el treball inicial de Schank s'utilitza com a exemple de guió l'anada a un restaurant. El guió és una descripció estereotipada (abstracta) del fet d'anar al restaurant i permet després entendre un text concret descrivint l'anada, i també preveure allò que passarà a continuació. Un altre exemple del treball de Schank és el següent:

"II. We saw the Packers-Rams game yesterday. The Packers won on a dive play from the two with three seconds left. Afterwards they gave the game ball to the fullback.

(...) It is perhaps simplest to point out that the second paragraph is incomprehensible to someone who does not know football yet makes perfect sense to someone who does. In fact, football is never mentioned, yet questions like «What kind of ball was given to the fullback?» and «Why was it given?» are simple enough to answer, as long as the script is available."

R. Schank (1975). "The Structure of Episodes". A: D. G. Bobrow; A. Collins (ed.), *Memory, Representation and Understanding: Studies in Cognitive Science*. Nova York: Academic Press.

En aquest apartat es descriuran els sistemes de representació estructurats centrant-nos en el cas dels marcs.

Els orígens dels llenguatges de marcs estan en els treballs d'M. Minsky sobre percepció en els sistemes de visió. Minsky, per la seva banda, estava inspirat en algunes teories de la psicologia com els esquemes de F. C. Bartlett: per a analitzar una situació nova, un ésser humà consulta les estructures elementals

que integren les situacions a les quals ja s'hi ha encarat abans. Això és, es tria l'estructura que li sembla més propera a la situació actual i, si és necessari, la modifica per a adaptar-la a la nova situació.

Els marcs representen conceptes que defineixen aquestes situacions o, fins i tot, representen les situacions mateixes. Permeten tenir en compte els valors típics (alguns llenguatges es basen en la idea del prototipus), les excepcions, la informació incompleta o redundant. A més, per tal de poder representar com canvia l'entorn, permeten modificacions, inclusions o supressions de les seves propietats. Aquestes modificacions provocaran reajustaments en l'estructura dels marcs.

4.1. Aspecte formal

Des del punt de vista de l'aspecte formal, un sistema de marcs és una xarxa on cada node representa un objecte (un concepte o un individu) i els arcs corresponen a les relacions que es poden establir entre els objectes. Els objectes es representen en marcs, que són un conjunt de camps¹³ que el defineixen.

(13)En anglès, *slots*.

Aquesta forma de representació estructura el coneixement de manera semblant a com s'estructura la informació en un llenguatge orientat a l'objecte.

Els llenguatges d'objectes

Vist de manera àmplia, els llenguatges d'objectes són tots aquells que porten a una estructuració de l'univers d'una aplicació en termes d'objectes. Aquesta interpretació permet englobar als llenguatges de marcs en el paraigües dels orientats a l'objecte. En els llenguatges d'objectes:

“Trois grandes familles apparaissent naturellement, chacune privilégiant un point de vue sur la notion d'objet:

- Le point de vue *structurel*: l'objet est un type de données, qui définit un modèle pour la structure de ser représentants physiques et un ensemble d'opérations applicables à cette structure.
- Le point de vue *conceptuel*: l'objet est une unité de connaissance, représentant le prototype d'un concept.
- Le point de vue *acteur*: l'objet est une entité autonome et active qui se reproduit par copie.”

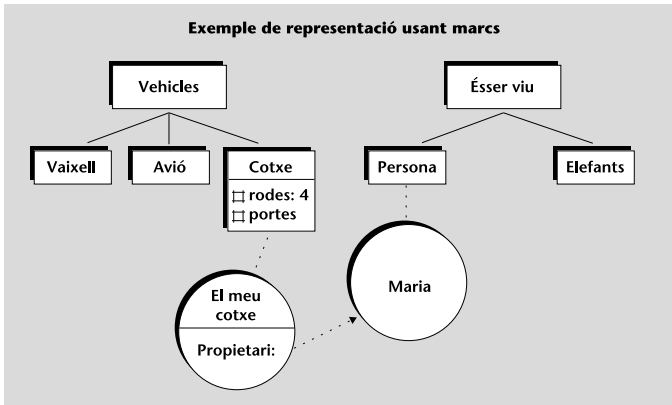
G. Masini; A. Napoli; D. Colnet; D. Léonard; K. Tombre (1989). *Les langages à objets*. París: InterEditions.

Els primers són els llenguatges de classes, els segons són els de marcs i els tercers els d'actors (o d'agents).

Unes de les relacions que s'estableixen entre els objectes són les que també apareixen en aquests llenguatges: relacions de subclasse, superclasse i d'instància. La relació de subclasse correspon a quan un concepte és un cas particular d'un altre. La relació de superclasse és la relació inversa: quan un concepte és més general que un altre. La relació d'instància la trobem quan un marc correspon a un individu particular d'una classe.

La figura 4 dona un exemple de representació fent servir marcs. En aquest exemple, el concepte cotxe és una subclasse del concepte vehicle, i vehicle és una superclasse del concepte cotxe. Per altra banda, el marc corresponent al meu cotxe representa una instància del concepte cotxe (el meu és un cas particular dels cotxes). També la propietària del cotxe, la Maria, és una instància de persona (un cas particular de persona).

Figura 4



Sovint de camps n'hi ha de dos tipus: els de membre⁽¹⁴⁾ i els propis⁽¹⁵⁾. Els primers són aquells camps que són a cada instància d'un objecte (cada instància –cada membre de la classe– en té la seva pròpia còpia i la modifica lliurement), i els segons són aquells que són propis del marc (i, per tant, que comparteixen totes les instàncies). Són com les variables de classe d'un llenguatge orientat a l'objecte. Els camps poden emmagatzemar informació de tipus molt diversa. Alguns dels elements que típicament s'hi poden incloure són aquests:

⁽¹⁴⁾En anglès, *member slot*

⁽¹⁵⁾En anglès, *own slot*.

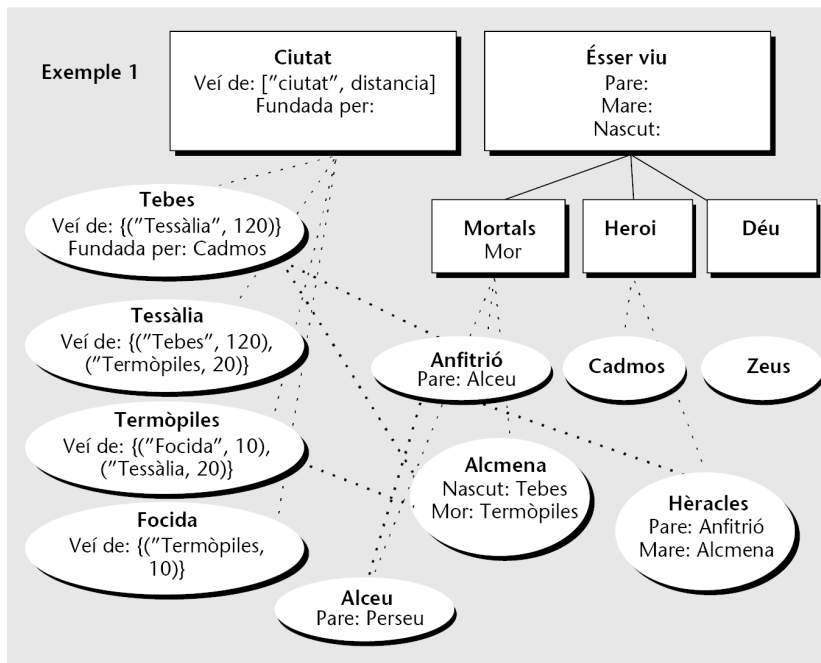
- Valors d'un domini qualsevol.
- Referència a altres marcs. En aquest cas, el valor indica un objecte del mateix sistema. Per exemple, en la figura 4 tenim que el camp propietari del marc cotxe es refereix a una persona concreta: la Maria. Aquest tipus de camp és el que ens permet lligar els conceptes entre si per a formar l'estructura dels conceptes.
- Restriccions sobre valors. En lloc de representar el camp amb un valor concret es disposa d'una restricció sobre els valors possibles. Poden ser restriccions sobre el tipus de valor (per exemple, el valor ha de ser un enter), sobre quin tipus d'instància es pot lligar a un marc (per exemple, el propietari d'un cotxe ha de ser una persona) o bé mitjançant connectives lògiques (per exemple, el valor pot ser 30, 31 o 32).
- Funcions. El camp té associada una funció que es crida quan és necessari calcular el valor del camp. El fet de lligar procediments a les estructures és una característica important dels sistemes de marcs perquè integren coneixement procedural que és molt difícil de representar de manera decla-

rativa. Així, els marcs permeten d'incloure tant coneixement procedural com declaratiu.

Exemple de marcs 1

Imaginem que volem modelitzar els diferents elements de la mitologia grega utilitzant un sistema de marcs. Sabem que a l'antiga Grècia hi havia les ciutats de Tebes, Tessàlia, Termòpiles i Fòcida. Per a cada ciutat ens interessa conèixer qui la va fundar i les seves comunicacions. Per a emmagatzemar les comunicacions fixem-nos que també podem definir un camp com una llista. Sabem que Tebes va ser fundada per Cadmos i que està a 120 km de Tessàlia. De Tessàlia sabem que està a 20 km de Termòpiles i 120 km de Tebes. De Termòpiles que està a 10 km de Fòcida i 20 km de Tessàlia i Fòcida que està a 10 km de Termòpiles. La mitologia grega considera que els déus, els herois i els mortals són éssers vius, dels quals ens interessa qui són els seus pares (pare i mare) i on han nascut. Alguns dels personatges de la mitologia grega són Zeus, que era un déu, i Cadmos, que era considerat un heroi igual que Hèracles. D'Hèracles sabem que la mare era Alcmena i el pare Amfitrió. La mare Alcmena era mortal i va néixer a Tebes i morí a Termòpiles. El pare Amfitrió també era mortal i el seu pare era Alceu. Alceu era mortal i el seu pare era Perseu. Podríem pensar en un sistema de marcs, amb la representació gràfica següent:

Figura 5



Exemple de marcs 2

Imaginem que volem modelitzar mitjançant marcs el famós joc *Pokémon Go*. Aquest joc permet capturar, emmagatzemar i entrenar per lluitar diferents *Pokémons*. Aquests *Pokémons* són emmagatzemats per cada jugador particular a la seva *PokéDex* (magatzem amb tots els *Pokémons*). De tots els diferents tipus de *Pokémons*, nosaltres en considerarem tres: els *Pokémon* verí (o poisson) (P), els *Pokémon* aigua (o water) (W) i els *Pokémon* herba (o grass) (G). Qualsevol *Pokémon* del joc serà caracteritzat per tres propietats:

- El nivell d'atac: se suposa que es refereix a la capacitat de destruir el contrari en una lluita i que es pot millorar entrenant el *Pokémon*. Pot prendre els valors de baix, moderat o alt (valor per defecte: baix).
- La força: mesurada amb nombres enters (unitat N).
- Si és un *Pokémon* que prové d'un altre (evolucionat) o és un tipus de *Pokémon* en si mateix (no evolucionat). Suposarem que pot prendre dos valors: No o Sí.

Els *Pokémon* tipus verí suposarem que són sempre evolucionats i tenen una força màxima de 10 N; els *Pokémon* d'aigua i els *Pokémon* d'herba són sempre no evolucionats. El nivell d'atac d'un *Pokémon* d'aigua és sempre alt; en canvi, els *Pokémon* d'herba tenen un nivell d'atac per defecte, mig. Finalment, a causa de restriccions en la força, aquest tipus de *Pokémon* (herba) els suposarem que per defecte la força és de 3 N. Però hi ha *Pokémon*

que combinen les capacitats d'uns i altres: *Pokémon* aigua-verí, que són no evolucionats, i els *Pokémon* herba-verí.

Una possible modelització de la informació proporcionada mitjançant marcs, podria estar formada per les classes i subclasses següents:

- **Pokémons:** classe arrel de la jerarquia. Aquesta classe té els camps Evolucionat (No, Sí), Força N (enter) i Nivell d'atac (baix, moderat o mig, alt; valor per defecte: baix). A partir de la classe es definirien tres subclasses:
 - **Pok_aigua:** subclasse de Pokémon. Té el valor propi No a Evolucionat i el valor propi alt a Nivell d'atac. Per tant aquests valors no es poden canviar en les subclasses i instàncies d'aquesta classe.
 - **Pok_veri:** subclasse de Pokémon. Té el valor per defecte Sí a Evolucionat, i la Força està entre 0 i 10 N.
 - **Pok_herba:** subclasse de Pokémon. Té el valor propi No al camp Evolucionat, el valor per defecte mig a Nivell d'atac i el valor per defecte de Força és 3 N.

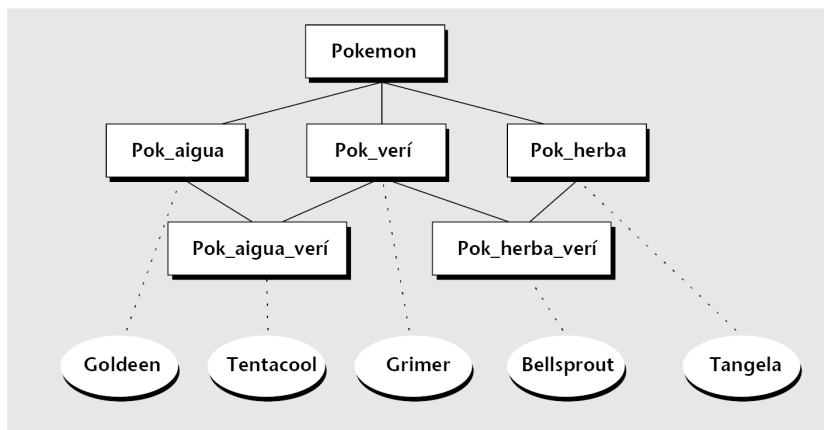
Finalment, definiríem dues subclasses que hereten de dues classes a la vegada:

- **Pok_aigua_veri:** subclasse de les classes Pok_aigua i Pok_veri . Té el valor per defecte No al camp Evolucionat.
- **Pok_herba_veri:** subclasse de les classes Pok_herba i Pok_veri.

Fent servir un sistema de marcs com aquest podríem representar la informació que un jugador té a la seva PokéDex, mitjançant instàncies. Per exemple, un jugador té un Grimer, que és un Pokémon de tipus verí amb 6 N de Força, el Tentacool, que és un Pokémon de tipus aigua i verí que té 2 N de força, el Goldeen, que és un Pokémon d'aigua, el Bellsprout, que és un Pokémon que té les característiques dels Pokémon herba i també del Pokémon verí, i la Tangela, que és un Pokémon de tipus herba.

De manera que el sistema de marcs, en general es podria representar de la manera següent:

Figura 6



Tindria les instàncies següents:

- Goldeen: instància de Pok_aigua.
- Tentacool: instància de Pok_aigua_veri. Força: 2 N.
- Grimer: instància de Pok_veri. Força: 6 N.
- Bellsprout: instància de Pok_herba_veri.
- Tangela: instància de Pok_herba.

Tots els camps serien tipus membre excepte aquells que ja s'ha especificat que són camps propis.

Els sistemes de marcs, a més dels camps tenen els anomenats procediments *dimonis*¹⁶. Aquests procediments es criden com a efecte secundari d'alguna actuació rellevant en la base de coneixements. Els dimonis estan generalment associats a les descripcions d'objectes i quan l'objecte canvia (o és creat o eli-

⁽¹⁶⁾En anglès, *demons*.

minat), s'activaran automàticament. Per exemple, poder fer servir els dimonis si en variar el valor d'un camp s'ha de fer una comprovació del seu tipus, o s'han de fer proves per a assegurar la consistència de la nova base de coneixements. Aquest seria el cas d'un sistema d'ajuda a la conducció si volguéssim controlar el nivell de benzina del cotxe: tindríem un procediment dimoni associat al nivell de benzina del cotxe de manera que quan la benzina del cotxe estigués per sota d'un llindar, s'activés una certa variable.

Per exemple, imagineu un sistema de marcs que modelitza els equips i jugadors d'una lliga de futbol. S'ha definit la classe:

Jugador_futbol: superclasse de home_adult.

Camps:

- Categoria (Excel·lent, no Excel·lent), per defecte no Excel·lent. Dimoni que en modificarà el valor.
- Equip (tots els de la lliga).
- Assistencies_partit: enter.
- Recuperacions_partit: enter.
- Percentatge_minuts: real.

Per a calcular la categoria, es defineix un dimoni que determina el valor quan el número de recuperacions de pilota més el d'assistències supera el valor de 10 i el *percentatge de minuts de joc* per partit supera el 70% de la durada del partit, llavors el jugador es considera excel·lent.

```
{
  jugador_futbol.categoria:: tipus
  (ARGUMENTS: jugador)
  (COS:
  Si
    (jugador.Assistencies_partit + jugador.Recuperacions_partit>10)
    ^ (jugador.Percentatge_minuts>70)
  LLAVORS
    (jugador.categoria=Excel·lent)
  )
}
```

4.2. Aspecte inferencial

La interrogació a un sistema de marcs acostuma a ser per saber si un objecte satisfà una determinada propietat o per saber el valor que un determinat camp associa a un objecte. En ambdós casos, el problema es redueix a trobar el camp que respon a la pregunta. Per a trobar aquest camp els sistemes de marcs es basen en l'herència, el funcionament de la qual és el mateix que trobem en els llenguatge orientats a l'objecte.

Així, el mecanisme general per a trobar el camp que pot respondre a la interrogació consisteix a mirar primer si el camp està definit en el mateix objecte i si té cap valor associat. Quan això no passa, es consultaran altres marcs d'acord amb les relacions d'herència. La consulta es fa resseguint les relacions establertes entre objectes. Això és, primer es mirarà la superclasse de l'objecte, després la superclasse de la superclasse, i així successivament fins que es trobi el camp –la resposta– que cerquem o ja no trobem cap altre objecte més general on consultar.

Aquest mecanisme es complica quan es permet, com és força freqüent, l'herència múltiple.

Tenim herència múltiple quan un objecte és instància de més d'una classe o bé si és subclasse de més d'una classe. En aquests casos, hi poden haver conflictes per a determinar la resposta del sistema perquè segons l'ordre en què es considerin les relacions de superclasse el sistema retornarà una cosa o una altra. Això és degut al fet que podem trobar diversos marcs amb el camp pel qual s'interroga. Quan els valors associats a aquests camps són diferents, el resultat variarà segons el marc que se seleccioni.

De fet, el cas de tenir herència múltiple correspon a un problema de cerca (un problema de control) perquè per a trobar el resultat podem considerar diferents alternatives (diferents relacions classe-superclasse), i no totes ens porten a un resultat (i quan hi porten no té perquè ser sempre el mateix).

Des d'un punt de vista abstracte, el problema de trobar un determinat camp correspon a fer un recorregut per un graf fins que trobem un node que satisfà una determinada propietat.

En aquest cas tenim que els nodes són objectes i els arcs són les relacions amb les quals l'herència treballa (o sigui, relacions de superclasse i d'instància). Aleshores, partint des del node corresponent a l'objecte pel qual s'interroga hem de trobar un node que contingui el camp. Quan només hi ha herència simple el conjunt de marcs i relacions correspon a un arbre (o un bosc –un conjunt d'arbres– si no hi ha una única arrel). Per tant, cap a l'arrel només hi ha un únic camí. Així, en fer la cerca, no hi ha mai alternatives a considerar. En el cas de l'herència múltiple això no és així. Hi ha nodes en els quals hi ha més d'un camí cap a l'arrel.

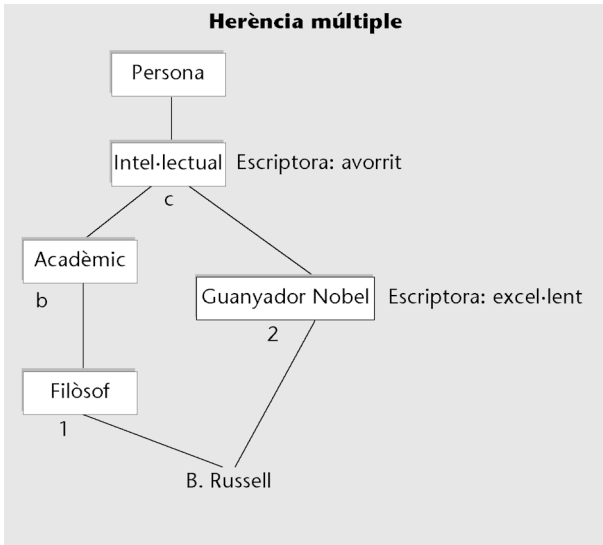
Vegeu també

Vegeu els aspectes de la cerca esmentats en el subapartat 2.4 d'aquest mòdul.

Vegeu també

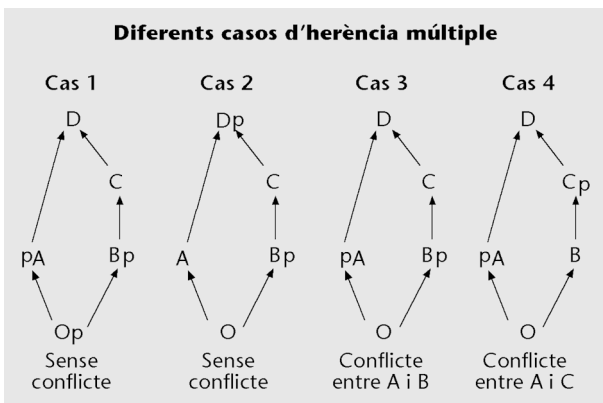
Vegeu els algorismes de cerca dels apartats 2 i 3 del mòdul "Resolució de problemes i cerca" d'aquesta assignatura

Figura 7



Vist el problema com un recorregut en un graf, podem aplicar-hi els algorismes de cerca vistos en el mòdul de cerca. Per exemple, podem aplicar els algorismes d'amplada i profunditat. Tanmateix, s'ha de tenir en compte que mètodes diferents poden donar solucions diferents. Per exemple, si considerem el cas de la figura 5 i la pregunta "quin tipus d'escriptura té B. Russell?" obtindrem resultats diferents quan la cerca és en amplada que quan és en profunditat. En el primer cas, per a trobar el camp "tipus escriptura" visitarem els nodes 1 i 2 i obtindrem com a resultat: excel·lent. En canvi, si la cerca és en profunditat obtindrem avorrida perquè els nodes que visitarem seran els marcats com a b i c.

Figura 8



Casos amb herència múltiple

A la figura es presenten alguns casos d'herència múltiple que val la pena de considerar. Adaptat d'H. Reighgelt (1991). *Knowledge Representation: An AI perspective*. Ablex Publishing Corporation.

De totes maneres, la manera de fer el recorregut no és arbitrària i hi ha algunes condicions que cal tenir en compte. En la figura 6 es presenten els casos que cal considerar en l'herència múltiple.

En la figura 6 considerem que estem demanant si l'objecte O satisfà la propietat p. Aleshores hi haurà conflicte si la resposta no es troba a l'objecte O i la propietat apareix a dos altres objectes O1 i O2 als quals s'arriba per dos camins diferents (i a més no és possible arribar a un objecte des de l'altre). Així en el cas (1) no hi haurà conflicte perquè la propietat està en el mateix objecte O, i en el cas (2) no n'hi haurà perquè tot i que es pot arribar a l'objecte D per un

camí que no passa per B, n'hi ha un que sí que hi passa. Tampoc no hi hauria conflicte en el cas (2) si la propietat estigués a C en lloc de D. En els casos (3) i (4) hi ha conflicte. En el (3) el conflicte és entre A i B, mentre que en el (4) n'hi ha entre A i C. El nombre de passos que és necessari fer per a trobar un objecte no influeix en el fet que hi hagi conflicte.

D'acord amb aquests casos, els algorismes han de tenir en compte les cruïlles on es troben diversos camins: abans de considerar el que hi ha en l'objecte associat a la cruïlla s'han d'haver tingut en compte els nodes que hi ha en els camins que hi porten des de l'origen. En la figura 5 tenim que el marc corresponent a "intel·lectual" apareix en una cruïlla i, per això, abans de dir que l'escriptura és avorrida, hem d'haver visitat totes les subclasses d'"intel·lectual". En aquest cas, com que el marc "guanyador-de-nobel" també respon a la pregunta s'haurà d'haver fet servir per a respondre a la pregunta sobre el tipus d'escriptura.

Una manera de resoldre aquest problema és mitjançant un algorisme d'ordenació topològica. Aquest algorisme, donat un graf no cíclic calcula una llista ordenada de nodes en la qual es mantenen les relacions que hi ha entre els nodes del graf. Aquesta llista és l'anomenada *llista de precedències*.

L'ordenació topològica

L'ordenació topològica és una ordenació dels vèrtexs d'un graf dirigit acíclic, tal que si hi ha un arc de v_i a v_j , llavors v_j apareix després de v_i en l'ordenació. Podeu trobar més informació sobre aquest concepte en l'obra següent:
Weiss (1992). *Data Structures and Algorithm Analysis*. The Benjamin/Cummings Publishing Company, Inc.

```
funcio ordenació topològica (graf) retorna llista de nodes es
    siguin (a,b) els arcs del graf si el node "a" precedeix al "b"
    llista de precedència := llista buida
    mentre quedin arcs fer
        Seleccionar un arc (a,b) tal que no existeixi cap parella (*, a)
        ;; això vol dir que no hi ha cap dependència sobre "a",
        ;; i si hi era, les dependències ja han estat visitades
        llista de precedència := llista de precedència + "a"
        ;;afegeix el node "a" a la llista de precedència
        eliminar totes les parelles de la forma (a, *)
    fmentre
    retorna la llista de precedència
ffuncio
```

En aquest procediment, els parells (a, b) són els establerts pel graf. En el nostre cas, tindrem (a, b) si b és la superclasse de a o si a és una instància de b . Així, en el cas de la figura 7 tenim que el conjunt de parelles és:

```
(B-Russell, Filòsof), (Filòsof, Acadèmic), (B-Russell, Nobel),
(Acadèmic, Intel·lectual), (Nobel, Intel·lectual),
(Intel·lectual, Persona).
```


El procediment construeix la llista prenent cada vegada un node que no tingui dependències de cap altre node i després elimina totes les dependències que depenen d'aquest node. Això és, hem de seleccionar un arc en què l'element de l'esquerra no aparegui a la dreta en cap parell. En l'exemple de la figura 5, comencem seleccionant l'arc (B-Russell, Filòsof) perquè B-Russell no apareix mai a la dreta d'un parell. A continuació, eliminem tots els parells de la forma (B-Russell, *) –n'hi ha dos–, la llista queda:

```
(Filòsof, Acadèmic), (Acadèmic, Intel·lectual), (Nobel, Intel·lectual),
(Intel·lectual, Persona).
```

S'afegeix a la llista de precedència B-Russell.

Passem a seleccionar un altre arc. Ara hi ha dues alternatives, perquè podem escollir tant l'arc (Filòsof, Acadèmic) com l'arc (Nobel, Intel·lectual). Si triem (Filòsof, Acadèmic), suposem que recorrem l'arbre d'esquerra a dreta, eliminem tots els parells de la forma (Filòsof, *) i la llista de parells serà:

```
(Acadèmic, Intel·lectual), (Nobel, Intel·lectual), (Intel·lectual, Persona)
```

I la llista de precedència passa a ser:

```
B-Russell, Filòsof
```

Passem a seleccionar un altre arc, de nou podríem seleccionar (Acadèmic, Intel·lectual) o (Nobel, Intel·lectual) però estem recorrent l'arbre d'esquerra a dreta, per tant, seleccionarem (Acadèmic, Intel·lectual). I eliminem els parells (Acadèmic, *):

La llista de parelles serà:

```
(Nobel, Intel·lectual), (Intel·lectual, Persona)
```

I la llista de precedències:

```
B-Russell, Filòsof, Acadèmic
```

A continuació, només podem escollir (Nobel, Intel·lectual) i la llista de precedències ens queda:

```
B-Russell, Filòsof, Acadèmic, Nobel(Escriptura_excel·lent)
```

Si recorrem l'arbre de dreta a esquerra, després d'afegir B-Russell a la llista de precedències, escollim l'arc (Nobel, Intel·lectual) i llavors la llista de parelles ens quedarà:

```
(Filòsof, Acadèmic), (Acadèmic, Intel·lectual), (Intel·lectual, Persona)
```

I la llista de precedència serà:

```
B-Russell, Nobel
```

A continuació, seleccionarem un altre arc i només podem seleccionar (Filòsof, acadèmic). Eliminem (Filòsof, *) de la llista de parelles i afegim Filòsof a la llista de precedència.

Llista de parelles:

```
(Acadèmic, Intel·lectual), (Intel·lectual, Persona)
```

Llista de precedència:

```
B-Russell, Nobel, Filòsof
```

Seleccionem la parella (Acadèmic, Intel·lectual), eliminem (Acadèmic, *) i afegim Acadèmic a la llista de precedència.

Llista de parelles:

```
(Intel·lectual, Persona)
```

Llista de precedència:

```
B-Russell, Nobel, Filòsof, Acadèmic
```

I, finalment, visitem l'arc (`Intel·lectual`, `Persona`) i la llista de precedència queda:

```
B-Russell, Nobel (Escriptura_excel·lent), Filòsof, Acadèmic,
Intel·lectual (Escriptura_avorrit)
```

És important subratllar que l'algorisme anterior no fixa el resultat per a qualsevol graf. Quan en un dels passos d'iteració hi ha més d'un node que només apareix a l'esquerra d'una parella s'ha de seleccionar un dels nodes. La manera de fer la tria afectarà el resultat. Aquest cas es dona en l'exemple en el segon pas quan s'ha de triar entre els arcs (`Nobel`, `Intel·lectual`) i (`Filòsof`, `Acadèmic`). Aquesta tria determina l'ordenació relativa entre els nodes `Nobel` i `Filòsof`. Però en ambdues llistes de precedència la propietat `escriptura_excel·lent` surt sempre abans. Per tant, el conflicte és resoluble ja que ambdós camins porten al mateix resultat.

A més de les operacions que permeten avaluar les propietats d'un objecte, els sistemes de marcs disposen d'algorismes tant per a afegir nous conceptes com per a construir instàncies d'objectes ja existents. La implementació d'aquestes operacions es pot fer sobre la base de funcions associades a altres marcs de manera que el sistema sigui completament homogeni. D'aquesta manera les operacions estaran definides en els marcs del mateix sistema. Això és semblant al que passa en alguns llenguatges orientats a l'objecte (el cas paradigmàtic és `Smalltalk`) on tot es pot definir de manera homogènia així que totes les operacions són definides dins els mateixos objectes del llenguatge.

En l'exemple 2 de marcs que modelitzava el joc de *Pokémon Go* presentat permetia conèixer característiques dels *Pokémons*. Però, per exemple, per a conèixer el nivell d'atac del *Pokémon Bellsprout* calia aplicar l'ordenació topològica per a resoldre el conflicte que es produeix entre el valor heretat de la classe *Pokémon* (baix) i el valor heretat de la subclasse *Pok_herba* (mig). Fixem-nos, però, que aquest conflicte era resoluble mitjançant l'algoritme d'ordenació topològica, ja que aplicant l'algoritme els parells (*a*, *b*) establerts pel graf són:

```
(Bellsprout, Pok_herba_verí), (Pok_herba_verí, Pok_verí), (Pok_herba_verí,
Pok_herba), (Pok_verí, Pokemon), (Pok_herba, Pokemon)
```

Lectura complementària

Una bona descripció de l'estructuració de classes fent èmfasi en l'homogeneïtat del sistema d'objectes d'`Smalltalk` es dona a l'obra següent:

T. Hopkins; B. Horan (1995). *Smalltalk: An introduction to application development using VisualWorks*. Prentice-Hall.

De manera que si recorrem el graf d'esquerra a dreta, primer se selecciona l'arc (Bellsprout, Pok_herba_veri) i s'afegeix Bellsprout a la llista de precedència. A continuació, la llista de parells queda:

```
Pok_herba_veri, Pok_veri), (Pok_herba_veri, Pok_herba), (Pok_veri, Pokemon),
(Pok_herba, Pokemon)
```

Després se selecciona entre l'alternativa (Pok_herba_veri, Pok_veri), o l'alternativa (Pok_herba_veri, Pok_herba). Es tria (Pok_herba_veri, Pok_veri), perquè estem recorrent l'arbre d'esquerra a dreta, s'afegeix Pok_herba_veri a la llista de precedència i s'eliminen les parelles (Pok_herba-veri, *). La llista de parelles resultant és:

```
(Pok_veri, Pokemon), (Pok_herba, Pokemon)
```

Finalment, se selecciona un arc entre les alternatives (Pok_veri, Pokemon) i (Pok_herba, Pokemon). Com que ho recorrem d'esquerra a dreta se selecciona primer (Pok_veri, Pokemon). La llista de parells resultant és: (Pok_herba, Pokemon) i s'afegeix Pok_veri a la llista de precedències. Finalment, se selecciona (Pok_herba, Pokemon). Per tant, la llista de precedència serà:

```
Bellsprout, Pok_herba_veri, Pok_veri, Pok_herba
```

Quan revisem la llista de precedència trobem la propietat nivell d'atac definida a Pok_herba amb valor mig.

Si haguéssim recorregut el graf de dreta a esquerra, els parells establerts pel graf serien els mateixos, però després d'haver afegit Bellsprout a la llista de precedència tindríem la llista de parells següents:

```
(Pok_herba_veri, Pok_veri), (Pok_herba_veri, Pok_herba), (Pok_veri, Pokemon),
(Pok_herba, Pokemon)
```

Seleccionaríem `(Pok_herba_veri, Pok_herba)` i afegiríem `Pok_herba_veri` a la llista de precedència eliminant les parelles `(Pok_herba-veri, *)`. A continuació, la llista de parelles resultant seria:

```
(Pok_veri, Pokemon), (Pok_herba, Pokemon)
```

En aquest cas, al recórrer el graf de dreta a esquerra, escolliríem primer `(Pok_herba, Pokemon)`. La llista resultant seria `(Pok_veri, Pokemon)` i afegiríem `Pok_herba` a la llista de precedència, seleccionant finalment `(Pok_veri, Pokemon)`.

Per tant, la llista de precedència seria:

```
Bellsprout, Pok_herba_veri, Pok_herba, Pok_veri
```

Quan revisem la llista de precedència trobem la propietat nivell d'atac definit a `Pok_herba` amb valor mig, un node abans que en el cas anterior però amb el mateix valor. Per tant, esdevé un conflicte resoluble.

Nota

Reviseu el problema 7 d'autoavaluació per a veure un exemple de conflicte no resoluble.

4.3. Anàlisi dels sistemes de marcs

Des d'un punt de vista formal els sistemes de marcs es poden veure com una lògica de primer ordre. De tota manera, hi ha dificultats a formalitzar aquesta equivalència quan els camps dels marcs tenen associats procediments o funcions. L'equivalència no impedeix que els marcs es facin servir sovint per raons d'eficiència i claredat. Això és així perquè aquests llenguatges permeten una representació gràfica del coneixement i la visualització de la inferència. A més, els llenguatges de consulta són simples impedint que es facin consultes complexes a la base de coneixements (la qual cosa en facilita el tractament).

La restricció no apareix sols a l'hora de fer les consultes sinó que també apareix en el coneixement que es pot incloure a la base de coneixements. Les restriccions en el llenguatge impedeixen d'expressar algunes de les proposicions que podríem escriure en lògica de primer ordre. Per exemple, no es poden traduir les sentències "o bé el color del cotxe és blau o bé el cotxe té més de cinc anys" i "el meu cotxe és blau o el teu és verd". La primera no es pot representar perquè relaciona dos camps d'un mateix marc (el marc corresponent al cotxe), i la segona no es pot representar perquè relaciona dos marcs diferents (tindrem un marc per a cadascun dels cotxes). El llenguatge també presenta dificultats per a representar alguns quantificadors existencials.

Les restriccions en l'aspecte formal provoca que aquí no aparegui el problema de la semidecidibilitat perquè o bé una propietat és a la jerarquia o bé no hi és. Aquesta limitació de l'expressivitat en favor de l'eficiència és un cas semblant al del llenguatge Prolog (que es fa servir més sovint que els demostradors de teoremes de lògica de primer ordre).

Per altra banda, l'estructuració del coneixement en jerarquies d'objectes fa que sigui molt fàcil la representació de les excepcions i també del raonament per defecte. Quan d'un objecte no es coneix una propietat, es mirarà de trobar mitjançant l'herència. En el cas de l'exemple de la figura 5, tenim que per defecte l'estil d'escriptura de qualsevol Nobel és excel·lent. Si per a una persona concreta en descobrim després l'estil, el definirem en el propi marc. Això farà que a partir d'aquell moment ja no es faci servir el coneixement per defecte sinó el propi i específic.

5. Sistema de raonament basat en casos

Una manera alternativa de resolució de problemes és considerar el conjunt d'experiències passades i resoldre les noves situacions de manera semblant a com es van resoldre les anteriors. El raonament basat en casos¹⁷ es fonamenta en aquesta idea.

⁽¹⁷⁾En anglès, *case-based reasoning*.

Per a portar-la a terme, el sistema emmagatzema unes experiències prèvies, i per a cadascuna d'aquestes guarda la solució que se li ha aplicat. Aleshores, quan se li planteja una nova situació, se cercarà una situació semblant entre aquelles que ja s'han vist abans i s'adaptarà la seva solució a la nova situació. En aquest tipus de sistemes, a cada experiència o situació se l'anomena cas, i al conjunt de casos *base de casos*.

Aquests sistemes s'han aplicat a diferents tipus de problemes. Algunes de les aplicacions més característiques són el disseny, la diagnosi i la planificació. En el cas del disseny d'un objecte, l'experiència prèvia sobre la construcció d'uns altres objectes permet saber com tractar les restriccions (per exemple, que unes components no poden estar adjacents a unes altres, o que el pes total de l'artefacte ha d'estar per sota d'un determinat límit). En diagnosi, els casos permeten evitar errors anteriors, reaprofitar decisions ja preses o bé focalitzar la cerca (en determinades condicions, només cal considerar un determinat conjunt de causes per a uns símptomes). Hi ha aplicacions a diagnosi de falles de maquinari i a diagnosi mèdica. Com s'ha vist en l'apartat 1.1.1 del mòdul "Resolució de problemes i cerca", la planificació consisteix a construir un pla a partir d'operadors. Per a fer-ho s'han de tenir en compte els requisits de cada operador. El raonament basat en casos permet usar plans ja construïts prèviament o adaptar-los a la nova situació.

La manera de definir el coneixement d'aquests sistemes és força diferent del coneixement general que es representa mitjançant regles o marcs. Aquí es diu que el coneixement és operacional. En un cas, la informació correspon a com fer servir el coneixement (com s'aplica) en una situació concreta. La solució és per a un cas concret. Aquesta diferència en el tipus de coneixement fa que per a determinades aplicacions els sistemes basats en casos siguin més adequats que els sistemes que només fan servir coneixement general.

Per exemple, en determinats entorns el coneixement general pot ser massa abstracte i difícil d'aplicar per a situacions concretes. A més, amb aquest coneixement operacional és més fàcil tractar situacions excepcionals (incloent el corresponent conjunt de casos a la base de casos) o de dominis amb coneixement incomplet.

La representació del coneixement partint de la base de casos, no impedeix que el sistema inclogui coneixement general de domini (per exemple, regles) per a resoldre una determinada situació. Per exemple, es pot fer servir coneixement general de domini per a adaptar la solució d'un cas existent o bé per a construir una solució quan no es troba cap cas similar.

Des del punt de vista de la fase d'adquisició del coneixement, els sistemes de raonament basat en casos també presenten avantatges atès que es pot reduir tot el procés de modelització. Ara, en no necessitar coneixement de domini no fa falta que un expert el defineixi amb la subseqüent despesa de recursos. En el seu lloc n'hi ha prou amb definir un conjunt de casos amb les seves solucions: la descripció d'un conjunt d'experiències. Acostuma a ser més senzill d'obtenir per part d'un expert aquestes experiències que no pas el coneixement general per a resoldre aquestes mateixes experiències.

Per altra banda, aquests sistemes poden aprendre de la seva pròpia experiència en poder emmagatzemar solucions de nous casos. D'aquesta manera quan un sistema torna a trobar una determinada situació no és necessari recalcular-ne la solució. El sistema pot reconèixer aquesta situació i podrà aplicar-hi la solució calculada anteriorment. També es pot emmagatzemar informació sobre els èxits o els fracassos de solucions donades anteriorment. L'ús de la informació sobre les experiències rellevants pot ser d'utilitat quan el sistema s'encari a nous problemes. La facilitat d'integrar les fases d'aprenentatge i raonament fa que sovint es consideri el raonament basat en casos un subcamp de l'aprenentatge automàtic.

De tota manera, aquests sistemes presenten també alguns inconvenients en relació amb els altres mecanismes de representació de coneixement i de resolució de problemes. S'ha de subratllar que un sistema de casos normalment no explora tot l'espai de solucions. Això, d'una banda, limita el camp d'aplicació del sistema i, de l'altra, les solucions que s'obtenen no tenen perquè ser globalment òptimes (poden ser òptims locals). A més, s'ha de tenir en compte que el sistema estarà esbiaixat segons els casos que hi ha a la base de coneixements.

5.1. Aspecte inferencial

L'estructura dels sistemes de raonament basat en casos és diversa tot i que mantenen alguns trets comuns. En general, el sistema té una estructura cíclica en què es consideren els passos següents (vegeu la figura 9):

Sistemes de raonament basat en casos

Hi ha sistemes de raonament basat en casos que sí que usen coneixement per a calcular solucions de nous problemes o per a determinar quin cas s'assembla al que anem a resoldre.

1) Recuperació¹⁸: es recuperen els casos precedents que són semblants al cas que hem de resoldre, i se'n selecciona el millor.

(18) En anglès, *retrieval*.

2) Reutilització¹⁹: es decideix si la solució del cas seleccionat es pot aplicar directament i si no és així se l'adapta a la nova situació.

(19) En anglès, *reuse*.

3) Revisió²⁰: s'aplica la solució al problema en curs i s'avalua. Pot passar que la solució no s'adapti completament i caldrà revisar-la.

(20) En anglès, *revision*.

4) Retenció²¹: s'emmagatzema el cas i la seva solució per a poder-la utilitzar més endavant.

(21) En anglès, *retain*.

Figura 9

Ara comentarem cadascun d'aquests passos amb una mica més de detall:

1) **Recuperació**: en principi, la recuperació d'un cas s'ha de fer sobre la base de la utilitat de la seva solució (per a construir la nova solució). De tota manera, com que això és difícil de saber, la recuperació es fa sobre la base de les característiques que descriuen un cas. Per exemple, en un sistema mèdic recuperarem casos de pacients amb símptomes semblants i un expedient mèdic similar. En general, calcularem la similitud entre els casos de la base de casos i aquell que volem resoldre i d'entre tots els considerats retornarem aquell que és més semblant. Per a calcular aquesta semblança es pot fer servir coneixement de domini (o simplement comparació sintàctica entre els valors del diferents camps dels casos).

Perquè la recuperació es pugui fer correctament i ràpidament els casos hauran d'estar indexats. El més senzill és considerar una organització de casos plana. Això correspon a comparar el cas actual amb tots els casos de la base però, evidentment, quan hi ha molts casos això provoca problemes per la quantitat de temps que necessitem. Una alternativa és utilitzar índexs jeràrquics amb els subseqüents problemes que comporta l'estructuració dels casos (en especial quan se n'afegeixen de nous) i de trobar el cas més adequat per a un problema donat (si l'estructuració no és bona no trobarem el cas més semblant). En aquest cas, els índexs han de tenir en compte quines són les característiques més rellevants en termes de trobar la solució.

2) **Reutilització**: si la solució que tenim ja es pot aplicar a la situació actual s'aplica. Si això no passa, haurem d'adaptar-la. Un dels mètodes per a fer l'adaptació es basa a disposar d'un conjunt de regles definides de manera que els seus antecedents siguin les possibles diferències entre un cas recuperat i un cas plantejat, i els conseqüents siguin les modificacions que s'ha de fer a la solució del cas recuperat per adaptar-la al cas plantejat. Amb aquesta informació, quan hem d'adaptar una solució seguirem els dos passos següents: (1) determinar les diferències entre el cas actual i aquell que hem recuperat; (2)

Funcionament d'un sistema basat en casos

Aquesta figura s'ha adaptat de A. Aamodt; E. Plaza (1994). "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches". *AI Communications* (núm. 7, pàg. 39-59).

aplicar les regles per a decidir les transformacions que cal fer a la solució del cas recuperat. Una altra alternativa per a l'adaptació és que per a cada cas tinguem indicacions de com s'ha obtingut la solució. Aquestes indicacions es fan servir per a trobar la solució del nou cas.

3) Revisió: l'aplicació de la solució al problema a resoldre realimenta el sistema ja que ens dóna una avaluació de la solució (podem saber si és correcta, si no ho és i, en alguns casos, disposar d'un grau de correcció). A vegades aquesta avaluació permet modificar la solució donada (adaptar-la o trobar-ne una de millor que no presenti problemes).

4) Retenció: el cas s'emmagatzema amb la solució calculada. A més també s'emmagatzemaran els resultats de la revisió. Quan els casos estiguin organitzats mitjançant un índex, el procés d'aprenentatge haurà d'adaptar aquestes estructures perquè el nou cas també pugui ser recuperat.

Els passos que s'han descrit aquí apareixen en la majoria de sistemes. De tota manera, no tots els sistemes presenten sempre les mateixes característiques. Per exemple, la manera d'emmagatzemar els casos tampoc no és homogènia. Hi ha sistemes que guarden experiències concretes mentre que d'altres guarden formes generalitzades creades a partir d'un conjunt de casos similars.

En general el comportament d'un sistema de raonament basat en casos dependrà fortament dels casos que s'incloguin en la base de casos, la manera en què es recuperen els casos perquè se seleccionin els que són rellevants i en els mecanismes d'adaptació de les solucions.

Lectura complementària

Per a més informació sobre raonament basat en casos podeu consultar les obres següents:

A. Aamodt; E. Plaza (1994). "Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches". *AI Communications* (núm. 7, pàg. 39-59), en suport digital a: <http://www.iiia.csic.es/es/staff/enric-plaza>

6. Sistemes de raonament basat en models

Una alternativa per a tractar problemes de diagnosi és relacionar les causes dels problemes amb els símptomes observables sense entrar en detall en el funcionament del sistema que diagnosticuem. Per exemple, en el cas de diagnosi mèdica podem concloure sobre una malaltia a partir d'unes dades que ens subministra el pacient i d'un conjunt de proves que se li fan. En el cas de diagnosticar la falla d'un aparell, es pot fer el diagnòstic a partir de l'anàlisi d'uns indicadors. En aquest cas el coneixement del sistema ve donat per un conjunt de relacions causa–efecte. Per exemple, regles que ens relacionen les causes amb els efectes o xarxes bayesianes que ens relacionen les diferents variables del problema. Una altra alternativa és considerar casos semblants com passa en els sistemes de raonament basat en casos.

Tanmateix, aquestes solucions no entren en els detalls de per què la falla provoca aquells símptomes. Aquest coneixement superficial que només relaciona causes i efectes sense entrar en els detalls de la relació planteja dificultats en algunes aplicacions. Per exemple, en un sistema expert es poden aplicar regles al cas que volem resoldre quan les regles no són les més adients. En alguns casos, una anàlisi més detallada del problema podria seleccionar la solució correcta.

Una alternativa és tenir més en compte els components d'allò que volem diagnosticar i les seves relacions. Els sistemes de raonament basats en models segueixen aquesta línia. Consideren un model detallat de l'estructura de l'aparell i una descripció de les seves components.

Per exemple, en el cas d'un sistema per a la diagnosi de circuits es prenen descripcions de les components que els defineixen i de com es comporten. A més, necessitarem saber com s'ha construït el circuit a partir dels components. El model permet de raonar sobre el circuit, i, en particular, a partir de la comparació entre el comportament del model i el del sistema real es pot fer una diagnosi.

La diferència entre els sistemes basats en regles i els basats en models correspon a la diferència entre dos tipus diferents de coneixement: el coneixement superficial²² i el coneixement profund²³. Una definició d'aquest tipus de coneixement és la que es dona a continuació:

Lectura complementària

J. de Kleer; B. C. Williams (1987). "Diagnosing Multiple Faults". *Artificial Intelligence* (núm. 32, pàg. 97-130) i també: <http://www.ai-cbr.org/>

⁽²²⁾En anglès, *shallow knowledge*.

⁽²³⁾En anglès, *deep knowledge*.

Shallow knowledge: This is the knowledge that a human expert might acquire by experience, without regard to the underlying reasons.

Deep knowledge: It is the fundamental building block of understanding. A number of deep rules might make up the causal links understanding a shallow rule.

A. A. Hopgood. (1993). *Knowledge-Based Systems for Engineers and Scientists*. CRC Press.

6.1. Aspecte formal

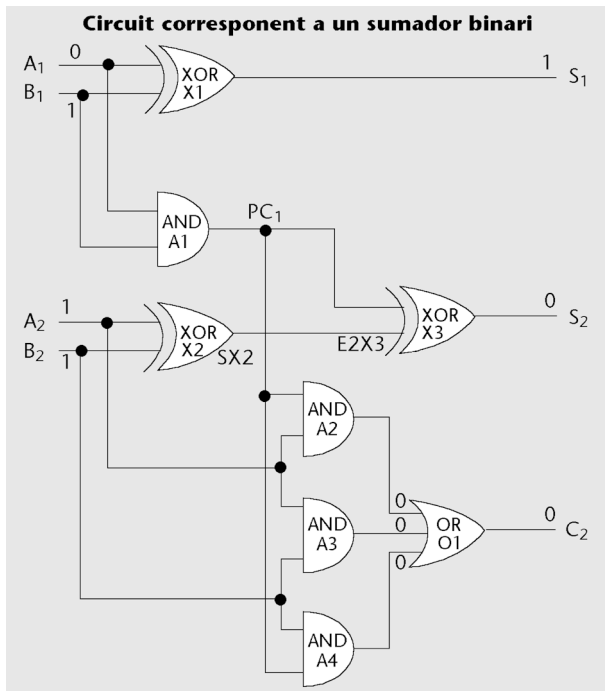
A continuació veurem amb una mica més de detall els elements relacionats amb els sistemes basats en models i com es fan servir per a fer diagnosi d'una fallada. Comencem amb el model del sistema, les possibles causes de fallada i les observacions:

a) **El model.** En primer lloc és necessari construir el model de l'aparell. Per tal de modelitzar un aparell es fa servir la descripció de l'estructura física juntament amb els models dels constituents. Aquí s'entén per *constituent* tot allò que defineix el sistema (components, processos, etc.). Per a representar el model, qualsevol mètode pot ser adequat (una representació en un llenguatge de programació o en un llenguatge de representació del coneixement).

b) **Les possibles causes.** Per a poder analitzar el sistema és necessari disposar d'informació sobre què és el que pot fallar en els diferents constituents i en l'estructura que els relaciona. Per això es considera per a cada constituent un conjunt de les possibles diferències entre el model i l'aparell. Això estableix la granularitat de l'error que detectarem. La diagnosi només considerarà aquelles possibles causes que hagin estat declarades.

c) **Les observacions.** Necessitem, a més, un conjunt de mesures sobre el sistema. Aquestes mesures són les que ens permetran de determinar quins components no funcionen correctament.

Figura 10



Per a il·lustrar aquest mecanisme de raonament considerem un exemple sobre detecció de falles en circuits. En particular, prendrem el que es dona en la figura 10 que correspon a un sumador binari.

La modelització d'aquest circuit ha de definir quines portes (AND, OR, XOR) es fan servir, quin és el seu comportament (mitjançant taules de veritat o expressions calculables en un llenguatge de programació) i com estan relacionades les portes (per exemple, que el resultat de la porta A1 està connectat amb el de la porta A2). A més, la modelització pot incloure les connexions. Per exemple, que hi ha connexions entre la porta A1 i el punt de contacte PC₁, i entre la sortida de la porta X2 (anomenada SX₂) i la segona entrada a la porta de X3 (denotada E2X3).

També cal definir les possibles causes d'error. Per exemple, que fallen les portes (o que també poden fallar les connexions).

A més, s'han d'especificar aquells punts en els quals és possible consultar el valor que està transmetent el circuit. Per exemple, podem observar les dades en cadascuna de les connexions (o en punts concrets de les connexions si suposem que aquestes també poden estar subjectes a fallades).

6.2. Aspecte inferencial

Quan el sistema no funciona correctament, hi haurà divergència entre les observacions i allò que el model prediu (quin valor haurien de prendre les observacions). En aquest cas el sistema plantejarà hipòtesis sobre les causes de la di-

vergència entre els valors. Com que es disposa dels models, serà possible avaluar les hipòtesis (corroborar-les o rebutjar-les) i finalment seleccionar aquelles que més bé es corresponen amb les observacions.

Per exemple, en la figura 8 es representa un cas de funcionament erroni. Quan les entrades al circuit tenen els valors $A1 = 0$, $B1 = 1$, $A2 = 1$ i $B2 = 1$ la sortida és $S1 = 1$, $S2 = 0$ i $C2 = 0$. En aquest cas, el sistema detectaria que la sortida $C2$ és incorrecta perquè en aplicar el seu model es preveu un valor d'1. El sistema formularà la hipòtesi que l'error es troba en la porta $O1$, però aquesta hipòtesi és rebutjada perquè les entrades d'aquesta porta són totes 0 i el model per a una porta OR diu que en aquest cas el resultat ha de ser zero. Per tant, $O1$ compleix les expectatives del model. Aplicant els models a les portes $A2$, $A3$ i $A4$ el sistema detecta que el resultat de $A3$ hauria de ser 1. Aquest funcionament incorrecte explica el resultat incorrecte del sumador. Per tant, formula la hipòtesi que falla la porta lògica $A3$.

Un sistema de diagnosi basat en models ha de trobar el conjunt de fallades més probable per a explicar la conducta que s'observa al sistema. Per a fer-ho planteja diferents alternatives i rebutja aquelles que no concorden amb el model i manté les que són corroborades pel model.

A causa que el nombre de possibles errors és gran, l'espai d'estats és considerable. Per exemple, en el cas de la figura 8 es té que si l'error només es pot trobar en alguna de les 8 portes existents, el nombre d'estats possibles és 2^8 . Això és així perquè qualsevol subconjunt de les 8 portes és una possible diagnosi.

Activitats

1. Suposant que els enunciats de la taula que veieu més avall corresponen a regles d'un sistema basat en regles, i suposant que la base de fets inicials és $\{E, F\}$, és a dir, que E i F es compleixen, considereu la demostració d' A .

a) Demostreu A quan l'interpret del sistema basat en regles funciona amb raonament cap endavant. Suposem que l'estratègia de selecció de regles es basa en la recència i quan dues regles tenen la mateixa recència fem servir l'ordre en què apareixen a la taula. A més, farem servir l'obstinància per tal de no repetir l'aplicació de les regles ja considerades.

Taula 2

Conjunt de sentències en lògica d'enunciats
$B \wedge C \rightarrow A$
$D \wedge C \rightarrow A$
$E \rightarrow C$
$E \rightarrow J$
$G \wedge H \rightarrow C$
$I \rightarrow H$
$I \rightarrow G$
$F \rightarrow D$
$J \wedge K \rightarrow B$
$L \rightarrow F$
$F \rightarrow K$

2. Suposant que els enunciats de la taula que veieu més avall corresponen a regles d'un sistema basat en regles, i suposant que la base de fets inicials és $\{A, D\}$, és a dir, que sabem que A i D es compleixen, considereu la demostració de Q , aplicant encadenament cap enrere.

Taula 3

Conjunt de sentències en lògica d'enunciats
R1. $P \rightarrow Q$
R2. $E \rightarrow B$
R3. $R \rightarrow Q$
R4. $M \wedge N \rightarrow Q$
R5. $A \wedge B \rightarrow P$
R6. $A \rightarrow M$
R7. $C \rightarrow M$
R8. $D \rightarrow N$

3. Tenim un llenguatge de representació del coneixement basat en marcs on hi ha herència múltiple. Suposeu que volem saber el valor d'un camp d'un marc. Quina relació té aquest problema amb una cerca en un espai d'estats (el problema del control). Què seria aquí un estat, els operadors, la funció objectiu (hi ha un únic estat final en aquest problema?).

4. Considereu el sistema de marcs de l'exemple del subapartat 4.1 d'aquest mòdul. Sobre aquest sistema responeu a les qüestions següents:

a) Com trobem l'avi d'Hèracles?

b) Doneu el graf amb les relacions d'herència que el sistema de marcs utilitzarà quan l'interroguem.

5. Com es pot representar raonament no monòton i per defecte en un sistema de marcs?

6. Segons el sistema de marcs del joc del *Pokémon Go*, representat en l'exemple 2, quin *Nivell d'atac* i quanta *Força* té el Grimer del jugador?

7. Explica per què es genera un conflicte irresoluble quan s'aplica l'algoritme d'ordenació topològica per a determinar si la instància *Bellsprout* és *Evolucionat*?

Solucionari

1. Per a construir la solució hem de procedir d'acord amb el que faria l'interpret. Hem de tenir en compte que el conjunt de regles està definit amb les de la taula de l'enunciat i que la memòria de treball només té a l'inici de l'execució a E i a F.

Per a procedir com l'interpret hem de començar mirant quines regles es poden aplicar d'acord amb el que hi ha a la memòria de treball. Trobem que les regles seleccionables són $E \rightarrow C$, $E \rightarrow J$, $F \rightarrow D$ i $F \rightarrow K$. Com que tots els fets són del mateix instant (els hi associem una marca de temps igual a zero) seleccionem la regla d'acord amb l'ordre. Per tant, triem $E \rightarrow C$. En aplicar-lo obtenim C i l'afegim a la memòria de treball amb una marca de temps igual a 1.

A continuació tornem a mirar les regles que es poden aplicar. Són les mateixes. Ara, però, com que la regla $E \rightarrow C$ ja no es pot aplicar (perquè apliquem l'estratègia d'obstinància) seleccionarem i aplicarem $E \rightarrow J$. Això ens fa afegir J a la memòria de treball i li assignarem una marca de temps igual a 2.

En el tercer pas les regles seleccionades són les mateixes però ara només podem aplicar $F \rightarrow D$. Així, afegim D a la memòria de treball amb la marca 3.

En el quart pas d'iteració les regles que podem aplicar són $D \wedge C \rightarrow A$, $E \rightarrow C$, $E \rightarrow J$, $F \rightarrow D$ i $F \rightarrow K$. D'acord amb l'estratègia de recència, seleccionem $D \wedge C \rightarrow A$ ja que les marques de temps de D i C ens diuen que són més recents que les de E i F. En aplicar la regla seleccionada obtenim A i per tant acaba l'execució.

En la taula següent es dona un resum de l'execució. Per a cada objecte de la memòria de treball es dona la marca de temps.

Taula 4

Resum de l'execució		
Regles	Regla seleccionada	Memòria de treball
		(E, 0) (F, 0)
$E \rightarrow C$, $E \rightarrow J$, $F \rightarrow D$, $F \rightarrow K$	$E \rightarrow C$	(E, 0) (F, 0) (C, 1)
$E \rightarrow C$, $E \rightarrow J$, $F \rightarrow D$, $F \rightarrow K$	$E \rightarrow J$	(E, 0) (F, 0) (C, 1) (J, 2)
$E \rightarrow C$, $E \rightarrow J$, $F \rightarrow D$, $F \rightarrow K$	$F \rightarrow D$	(E, 0) (F, 0) (C, 1) (J, 2) (D, 3)
$E \rightarrow C$, $E \rightarrow J$, $F \rightarrow D$, $F \rightarrow K$ $D \wedge C \rightarrow A$	$D \wedge C \rightarrow A$	(E, 0) (F, 0) (C, 1) (J, 2) (D, 3) (A, 4)

2. El nostre objectiu és Q. Aplicant encadenament cap enrere tenim:

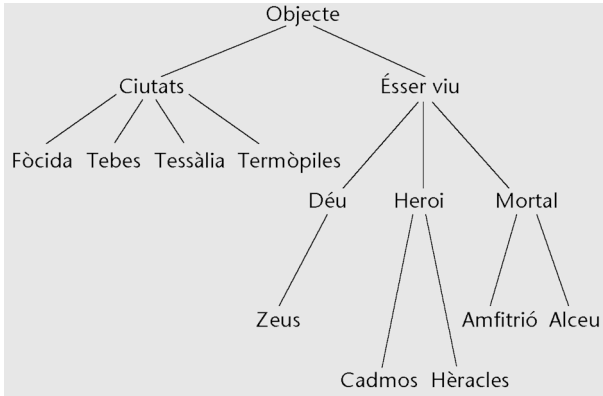
- Cicle 1: Conjunt conflicte {R1, R3, R4}, seleccionem R1 i canviem el subobjectiu a P.
- Cicle 1.1: Conjunt Conflicte per P: {R5} Per a demostrar P hem de demostrar A i B. A és cert, per tant, hem de demostrar B.
- Cicle 1.1.1: Seleccionem R2. Per a demostrar B llavors E hauria de ser cert i no ho podem demostrar.
- Cicle 2. Tornem a l'objectiu Q. Conjunt conflicte {R3 i R4}, seleccionem R3 i canviem el subobjectiu a R. No podem demostrar R.
- Cicle 3. Tornem a l'objectiu Q. Seleccionem R4 i canviem als subobjectius M i N. Per a demostrar M, hem de demostrar A. A és cert, per tant M és cert (aplicant la regla 6). Per altra banda, per a demostrar N hem de demostrar D. D és cert ja que està a la base de fets inicials. Finalment, sabent que M i N són certs demostrem que Q és cert.

3. El problema que hem de resoldre és el de cercar un marc que conté un determinat camp. Com que el que farem és consultar nodes fins que trobem el que ens interessa, podem definir l'estat com la visita a un node. Amb aquesta interpretació un operador correspondrà a passar d'un node a un altre. Considerarem que es pot passar d'un node A a un altre node B si B és una superclasse de A o si A és una instància de B. La funció objectiu serà mirar si el marc en el qual ens trobem conté ara el camp que es demana.

4. Les respostes a les qüestions de l'exercici són aquestes:

a) Primer anirem al marc corresponent a Hèracles i mirarem si hi ha el camp avi. Com que no hi és anirem després a cercar-lo a la classe de la qual és instància (la classe heroi) però tampoc no hi és. Seguim mirant la superclasse d'heroi que és ésser-viu. Allà hi trobem una funció i l'apliquem. Com que la funció ens diu que l'avi és el pare del pare del marc l'hem d'aplicar al marc que consultàvem: Hèracles. En fer-ho, la funció ens indica que hem de trobar el pare d'Hèracles. Això es troba en el mateix marc: és Amfitrió. Passem, doncs, a cercar el pare d'aquest darrer personatge. El trobem en el seu marc. És Alceu.

b) El graf d'herències és el de la figura següent:



5. L'herència permet tractar fàcilment el raonament per defecte. Quan tenim un marc, hereta de la superclasse (o de la classe de la qual és instància) tots els valors per defecte. Quan es descobreix que un determinat valor per defecte no és bo definirem un camp amb el mateix nom en el marc. Això provoca raonament no monòton. El que abans es deduïa ara no.

6. El *Grimer* tindria un *Nivell d'atac* baix, valor que heretaria de la classe *Pokemon* i tindria 6 N de *Força*, valor que estaria definit en la pròpia instància.

7. Les llistes de precedència obtingudes quan recorrem el graf per a la instància *Bellsprout* eren, quan el graf es recorria d'esquerra a dreta: *Bellsprout*, *Pok_herba_veri*, *Pok_veri*, *Pok_herba*, i quan el graf es recorria de dreta a esquerra: *Bellsprout*, *Pok_herba_veri*, *Pok_herba*, *Pok_veri*. Llavors si mirem quines classes tenen definit el camp *Evolucionat*, veiem que en el primer cas *Pok_veri* té definit el camp *Evolucionat* i el valor és *Sí*, mentre que, en el segon cas, *Pok_herba* té definit el camp *Evolucionat* i el valor és *No* i ambdós estan al mateix nivell. Per tant, el conflicte és irresoluble.

Glossari

aspecte formal *m* Aspecte que correspon a com s'emmagatzema la informació quan la tenim representada de manera explícita en un formalisme de representació del coneixement.

aspecte inferencial *m* Aspecte que correspon a com s'obté la informació que es troba en el sistema però només de manera implícita en un formalisme de representació del coneixement.

base de coneixements *f* Magatzem de coneixement dels sistemes basats en el coneixement.

compartir el coneixement *m* Englobar tots els aspectes relatius al fet que diferents sistemes puguin compartir el coneixement i no sigui necessari començar un model de nou quan es construeix un nou sistema. També és necessari compartir el coneixement quan es tenen diferents agents autònoms i es volen comunicar.

completesa *f* Mètode que és complet quan es poden deduir totes les conseqüències d'un sistema formal.

conjunt de conflicte *m* Conjunt de regles seleccionades per un sistema basat en regles en la fase de recuperació.

enginyeria del coneixement *f* Àrea de la intel·ligència artificial que estudia el procés de construcció de sistemes basats en el coneixement.

inconsistència *f* Una base de coneixements és inconsistent si podem deduir una propietat *a* i també la seva negació $\neg a$.

obstinància *f* Estratègia de selecció de regles en sistemes basats en regles (vegeu II.3.1).

procediment dimoni *m* Procediment que es crida com a efecte secundari d'alguna actuació rellevant en la base de coneixements.

recència *f* Estratègia de selecció de regles en sistemes basats en regles.

solidesa *f* Mètode que és sòlid quan només es generen oracions implicades.

Bibliografia

Bibliografia bàsica

Stefik, M. (1995). *Introduction to Knowledge Systems*. Morgan Kaufman.

Bibliografia complementària

Reighgelt, H. (1991). *Knowledge Representation: An AI perspective*. Ablex Publishing Corporation.

Luger, G.F. (1998). *Artificial Intelligence*. Addison-Wesley.