Reprezentarea cunoştinţelor

Un bun sistem de reprezentare a cunoştinţelor într-un anumit domeniu ar trebui să posede următoarele patru proprietăţi:

- ***Adecvare în reprezentare*** **- abilitatea de a reprezenta toate tipurile de cunoştinţe care sunt necesare într-un anumit domeniu.**
- ***Adecvare inferenţială*** **- abilitatea de a manipula structurile reprezentaţionale într-un asemenea mod încât să poată fi derivate structuri noi, corespunzătoare cunoştinţelor noi deduse din cele vechi.**
- ***Eficienţa inferenţială*** **- abilitatea de a încorpora în structura de cunoştinţe informaţii adiţionale care să poată fi folosite pentru a canaliza atenţia mecanismului de inferenţă în direcţiile cele mai promiţătoare.**

- ***Eficiența în achiziție*** **- abilitatea de a achiziționa uşor informații noi. În mod ideal, însusi programul ar trebui să poată controla achiziția de cunoştințe.**

Nici un sistem unic nu poate optimiza toate aceste caracteristici, nu în ultimul rând datorită tipurilor extrem de diverse de cunoştințe care există, majoritatea programelor bazându-se pe tehnici multiple.

Tipuri de cunoştinţe

Cunoştinţe relaţionale simple

Cea mai simplă modalitate de reprezentare a faptelor declarative constă în folosirea unei mulţimi de relaţii de acelaşi tip cu cele utilizate în sistemele de baze de date. Un exemplu de sistem relaţional este cel din Fig. 4.1. Cunoştinţele relaţionale din acest tabel corespund unei mulţimi de atribute şi de valori asociate, care împreună descriu obiectele bazei de cunoştinţe.

Student	Vârstă	An de studiu	Note la informatică
Popescu Andrei	18	I	8-9
Ionescu Maria	18	I	9-10
Hristea Oana	20	I	7-8
Pârvu Ana	19	II	8-9
Savu Andrei	19	II	7-8
Popescu Ana	20	III	9-10

Fig. 4.1

Această reprezentare este simplă deoarece, în sine, nu posedă şi nu furnizează capacitatea de inferență. Dar cunoştințele reprezentate în acest mod pot constitui input-ul adecvat pentru motoare de inferență mult mai puternice. Sistemele de baze de date sunt proiectate tocmai cu scopul de a furniza suportul necesar cunoştințelor relaționale.

Cunoştinţe care se moştenesc

Este posibil ca reprezentarea de bază să fie îmbogăţită cu mecanisme de inferenţă care operează asupra structurii reprezentării. Pentru ca această modalitate de reprezentare să fie eficientă, structura trebuie proiectată în aşa fel încât ea să corespundă mecanismelor de inferenţă dorite. Una dintre cele mai utilizate forme de inferenţă este *moştenirea proprietăţilor*, prin care elemente aparţinând anumitor clase *moştenesc atribute şi valori* provenite de la clase mai generale, în care sunt incluse. Pentru a admite moştenirea proprietăţilor, *obiectele* trebuie să fie organizate în *clase*, iar *clasele* trebuie să fie aranjate în cadrul unei *ierarhii*. În Fig. 4.2 sunt reprezentate cunoştinţe legate de jocul de fotbal, cunoştinţe organizate într-o structură de acest tip. În această reprezentare, liniile desemnează

atribute. Nodurile figurate prin dreptunghiuri reprezintă obiecte şi valori ale atributelor obiectelor. Aceste valori pot fi, la rândul lor, privite ca obiecte având atribute şi valori ş.a.m.d.. Săgețile corespunzătoare liniilor sunt orientate de la un obiect la valoarea lui (de-a lungul liniei desemnând atributul corespunzător).

În exemplul din Fig. 4.2., toate obiectele şi majoritatea atributelor care intervin corespund domeniului sportiv al jocului de fotbal şi nu au o semnificație generală. Singurele două excepții sunt atributul *isa*, utilizat pentru a desemna *incluziunea între clase* şi atributul *instanțiere*, folosit pentru a arăta *apartenența la o clasă*. Aceste două atribute, extrem de folosite, se află la baza *moştenirii proprietăților* ca tehnică de inferență.

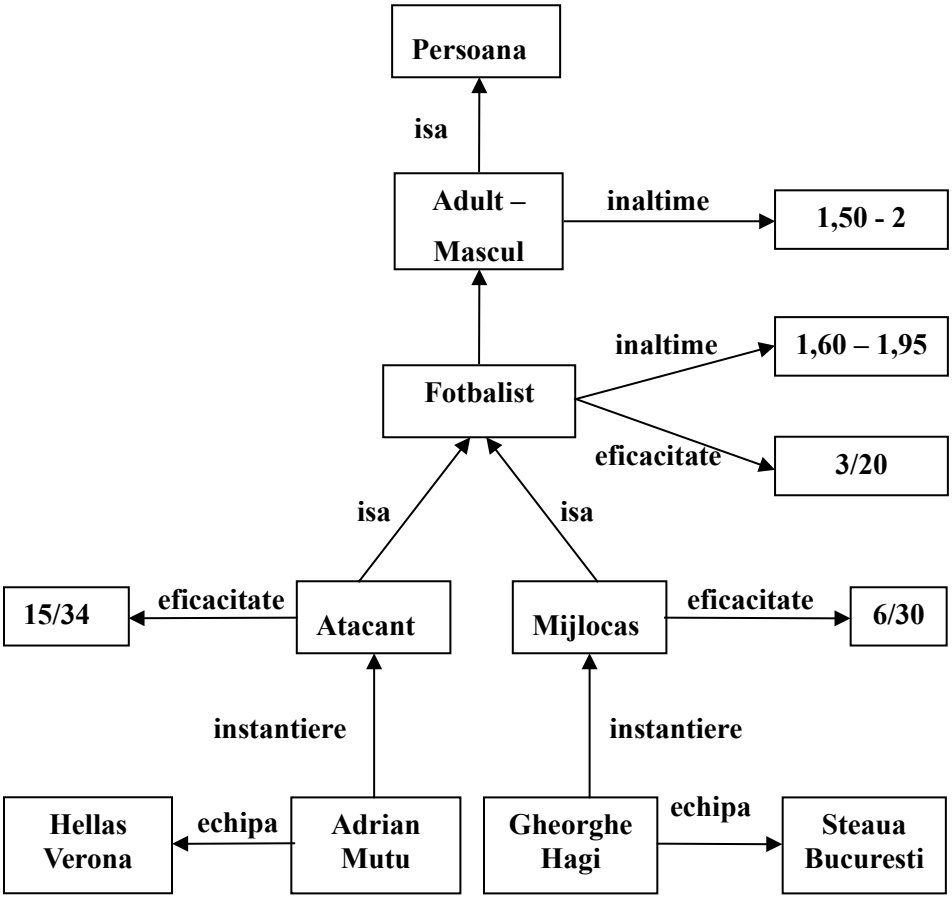

Fig. 4.2

Utilizând această tehnică de inferență, *baza de cunoştințe* poate asigura atât regăsirea faptelor care au fost memorate în mod explicit, precum şi a faptelor care derivă din cele memorate în mod explicit, ca în următorul exemplu:

eficacitate(Adrian_Mutu) = 15/34

Pentru a găsi răspunsul la această interogare, întrucât nu există nici o valoare pentru eficacitate memorată în mod explicit corespunzător lui Adrian_Mutu, a fost urmat atributul *instanțiere* până la *Atacant* şi a fost extrasă valoarea memorată acolo. Se poate acum observa una dintre caracteristicile negative ale moştenirii proprietăților, şi anume aceea că această tehnică de inferență poate produce valori implicite care nu sunt garantate a fi corecte, dar care reprezintă cele mai bune aproximări în lipsa unor informații exacte. Această tehnică de inferență continuă să fie printre cele mai folosite.

În acest exemplu, structura corespunzătoare reprezintă o *structură de tip* "*slot-and-filler*". Ea mai poate fi privită şi ca o *rețea semantică* sau ca o colecție de *cadre*. În cel din urmă caz, fiecare cadru individual reprezintă colecția atributelor şi a valorilor asociate cu un anumit nod.

O diferențiere exactă a acestor tipuri de reprezentări este greu de făcut datorită marii flexibilități care există în reprezentarea cunoştințelor. În general, termenul de sistem de cadre implică existența unei mai mari structurări a atributelor şi a mecanismelor de inferență care le pot fi aplicate decât în cazul rețelelor semantice.

Moştenirea proprietăților este o formă de inferență puternică, dar nu este suficientă pentru a construi un sistem de reprezentare complet, care, de cele mai multe ori, combină mai multe tehnici de reprezentare a cunoştințelor.

Cunoştinţe inferenţiale

Puterea logicii tradiționale este adesea utilă pentru a se descrie toate inferențele necesare. Astfel de cunoştinţe nu sunt utile decât în prezența unei proceduri de inferență care să le poată exploata. Procedura de inferență necesară în acest caz este una care implementează regulile logice de inferență standard. Există multe asemenea proceduri, dintre care unele fac raționamente de tipul "*înainte*", de la fapte date către concluzii, iar altele raționează "*înapoi*", de la concluziile dorite la faptele date. Una dintre procedurile cele mai folosite de acest tip este *rezoluția*, care foloseşte strategia contradicției.

În general, logica furnizează o structură puternică în cadrul căreia sunt descrise legăturile dintre valori. Ea se combină adesea cu un alt

limbaj puternic de descriere, cum ar fi o ierarhie de tip *isa*.

Cunoştinţe procedurale

Reprezentarea cunoştinţelor descrisă până în prezent s-a concentrat asupra faptelor statice, declarative. Un alt tip de cunoştinţe extrem de utile sunt *cunoştinţele procedurale sau operaţionale*, care specifică ce anume trebuie făcut şi când.

Cunoştinţele procedurale pot fi reprezentate în programe în diverse moduri. Cea mai simplă modalitate este cea sub formă de cod, într-un anumit limbaj de programare. În acest caz, maşina foloseşte cunoştinţele atunci când execută codul pentru a efectua o anumită sarcină. Acest mod de reprezentare a cunoştinţelor procedurale nu este însă cel mai fericit din punctul de vedere al adecvării inferenţiale, precum şi al eficienţei în achiziţie. Din această cauză s-au căutat alte modalităţi, diferite, de reprezentare

a cunoştinţelor procedurale, astfel încât acestea să poată fi manipulate relativ uşor atât de către oameni, cât şi de către alte programe.

Cea mai folosită tehnică de reprezentare a cunoştinţelor procedurale în programele de inteligenţă artificială este aceea a utilizării *regulilor de producţie*. Atunci când sunt îmbogăţite cu informaţii asupra felului în care trebuie să fie folosite, regulile de producţie sunt mai procedurale decât alte metode existente de reprezentare a cunoştinţelor.

Regulile de producţie, numite şi reguli de tip if-then, sunt instrucţiuni condiţionale, care pot avea diverse interpretări, cum ar fi:

- **if precondiţie P then concluzie C**
- **if situaţie S then acţiune A**
- **if condiţiile C1 şi C2 sunt verificate then condiţia C nu este verificată.**

Regulile de producţie sunt foarte utilizate în proiectarea *sistemelor expert*.

A face o distincție clară între cunoştințele declarative şi cele procedurale este foarte dificil. Diferența esențială între ele este dată de modul în care cunoştințele sunt folosite de către procedurile care le manipulează.

Clase de metode pentru reprezentarea cunoştinţelor

Principalele tipuri de reprezentări ale cunoştinţelor sunt reprezentările *bazate pe logică* şi cele de tip "*slot-filler*" ("deschizătură-umplutură").

Reprezentările bazate pe logică **aparțin unor două mari categorii, în funcție de instrumentele folosite în reprezentare, şi anume:**

- **Logica - mecanismul principal îl constituie inferența logică.**
- **Regulile (folosite, de pildă, în sistemele expert) - principalele mecanisme sunt "*înlănțuirea înainte*" şi "*înlănțuirea înapoi*". O regulă este similară unei implicații logice, dar nu are o valoare proprie (regulile sunt *aplicate*, ele nu au una dintre valorile „true" sau „false").**

Reprezentările de tip slot-filler **folosesc două categorii diferite de structuri:**

- **Rețele semantice şi grafuri conceptuale - o reprezentare "distribuită" (concepte legate între ele prin diverse relații). Principalul mecanism folosit este *căutarea*.**
- **Cadre şi scripturi - o reprezentare structurată (grupuri de concepte şi relații); sunt foarte utile în reprezentarea tipicității. Principalul mecanism folosit este *împerecherea* (potrivirea) *şabloanelor* (tiparelor).**

Reprezentarea cunoştinţelor şi sistemele expert

Un *sistem expert* este un program care se comportă ca un expert într-un domeniu relativ restrâns. Caracteristica majoră a sistemelor expert, numite şi *sisteme bazate pe cunoştinţe*, este aceea că ele se bazează pe cunoştinţele unui expert uman în domeniul care este studiat. Mai exact, ele se sprijină pe cunoştinţele expertului uman asupra strategiilor de rezolvare a problemelor specifice domeniului. Astfel, la baza sistemelor expert se află utilizarea în rezolvarea problemelor a unor mari cantităţi de cunoştinţe specifice domeniului.

Sistemele expert trebuie să poată rezolva probleme care necesită cunoştinţe într-un anumit domeniu. Prin urmare, ele trebuie să posede aceste cunoştinţe într-o anumită formă. Din această cauză ele se mai numesc şi sisteme bazate pe cunoştinţe. Nu toate sistemele bazate pe cunoştinţe constituie însă sisteme expert. Un sistem expert trebuie, în plus, să fie capabil *să explice* utilizatorului comportamentul său şi deciziile luate, la fel cum o fac experţii umani. Această caracteristică referitoare la generarea explicaţiilor este necesară în special în domeniile în care intervine *incertitudinea*, cum ar fi diagnosticarea medicală. Numai în acest fel utilizatorul poate detecta o posibilă fisură în raţionamentul sistemului.

O caracteristică suplimentară care este adesea cerută sistemelor expert este, prin urmare, abilitatea de a trata incertitudinea sau starea de a fi incomplet. Astfel, informațiile asupra problemei care trebuie rezolvată pot fi incomplete sau de natură a nu inspira încredere, iar relațiile din domeniul problemei pot fi aproximative. Spre exemplu, un anumit medicament poate genera anumite probleme, dar *cel mai adesea* nu o face.

Pentru a construi un sistem expert este necesară, în general, definirea următoarelor două funcții:

- **o *funcție de rezolvare a problemei*, funcție capabilă să folosească cunoştințele specifice domeniului şi care trebuie să poată trata incertitudinea.**
- **o *funcție de interacțiune cu utilizatorul*, care asigură şi generarea de explicații asupra intențiilor sistemului şi a deciziilor luate de acesta în timpul şi după procesul de rezolvare a problemei.**

Fiecare dintre aceste funcții poate fi foarte complicată şi depinde în mod nemijlocit de domeniul de aplicație şi de cerințele practice care pot să apară.

Structura de bază a unui sistem expert

Un sistem expert conține trei module principale, şi anume:

- **o bază de cunoştințe;**
- **un motor de inferență;**
- **o interfață cu utilizatorul.**

- ***Baza de cunoştinţe*** **cuprinde cunoştinţele specifice unui domeniu, inclusiv fapte simple referitoare la domeniul de aplicaţie, reguli care descriu relaţiile şi fenomenele proprii domeniului şi, eventual, metode, euristici şi idei pentru rezolvarea problemelor domeniului.**
- ***Motorul de inferenţă*** **utilizează în mod activ cunoştinţele din baza de cunoştinţe.**
- ***Interfaţa cu utilizatorul*** **asigură comunicarea dintre utilizator şi sistem, oferind utilizatorului o privire asupra procesului de rezolvare a problemei executat de către motorul de inferenţă. De cele mai multe ori motorul de inferenţă şi interfaţa sunt privite ca un unic modul, numit *shell* (înveliş).**

Structura de bază a unui sistem expert este prezentată în Fig. 4.3:

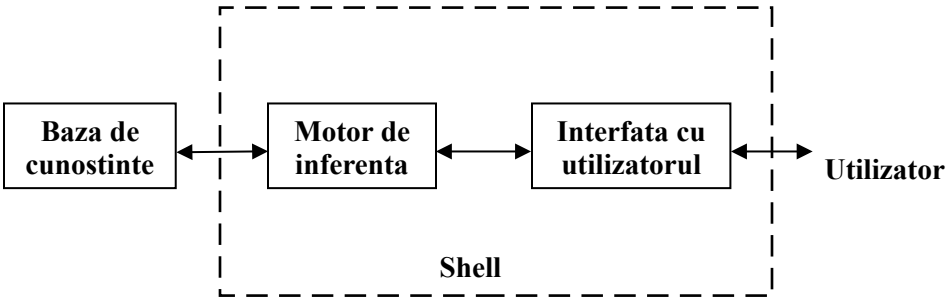
Baza de cunostinte
Motor de inferenta
Interfata cu utilizatorul
Utilizator
Shell

Fig. 4.3

Schema din Fig. 4.3 separă cunoştinţele de algoritmii care folosesc aceste cunoştinţe. Această separare este convenabilă din următoarele motive: baza de cunoştinţe depinde în mod clar de aplicaţie. Pe de altă parte, învelişul este, în principiu, independent de domeniu. O modalitate de a dezvolta sisteme expert pentru diverse aplicaţii, în aceste condiţii, constă din dezvoltarea unui *shell* care poate fi folosit în mod universal, cu utilizarea unei baze de cunoştinţe diferite corespunzător fiecărei aplicaţii. Desigur, toate bazele de cunoştinţe utilizate vor trebui să se conformeze unui acelaşi formalism care este înţeles de către *shell*. În practică, această abordare nu va avea succes decât în cazul în care domeniile de aplicaţie sunt extrem de similare, dar, chiar dacă modificări ale învelişului vor fi necesare, atunci când se trece de la un domeniu la altul, principiile de bază pot fi reţinute.