

BAZE DE CUNOȘTINȚE

SISTEME DE REPREZENARE ȘI PROCESARE
A CUNOȘTINȚELOR

M C
I O
H L
A H
E O
L N
A

TIPURI DE CUNOȘTINȚE

Pentru a putea rezolva problemele complexe de IA este nevoie atât de o mare cantitate de cunoștințe, cât și de mecanisme specifice pentru procesarea acestor cunoștințe cu scopul găsirii de soluții la problemele studiate.

Cunoștințele au o natură mai complexă decât datele și informațiile și necesită o contribuție activă din partea oamenilor pentru gestionarea și coordonarea lor.

Există două tipuri de entități tratate în sistemele de reprezentare a cunoștințelor:

- **fapte:** reprezentarea cunoașterii presupune existența unor fapte (adevaruri) care descriu o anumită situație sau univers de cunoaștere
- **reprezentări ale faptelor într-un anumit formalism:** aceste entități sunt procesate de mecanisme dedicate

TIPURI DE CUNOȘTINȚE

1. Cunoștințe relaționale simple
2. Cunoștințe care permit moștenirea
3. Cunoștințe inferențiale
4. Cunoștințe procedurale

TIPURI DE CUNOȘTINȚE

1. Cunoștințe relaționale simple

Cea mai simplă modalitate de reprezentare a faptelor declarative folosește schemele de relații utilizate în sistemele de baze de date.

Astfel, cunoștințele relaționale sunt descrise de o mulțime de attribute și caracterizate de valorile asociate acestora.

Această reprezentare nu include și mecanisme de inferență. Ele însă au devenit intrări pentru motoare de inferență puternice -> SGBDurile.

Nume	Prenume	Vârstă	Adresă
Ionescu	Ion	30	Craiova

TIPURI DE CUNOȘTINȚE

1. Cunoștințe relaționale simple
2. Cunoștințe care permit moștenirea
3. Cunoștințe inferențiale
4. Cunoștințe procedurale

TIPURI DE CUNOȘTINȚE

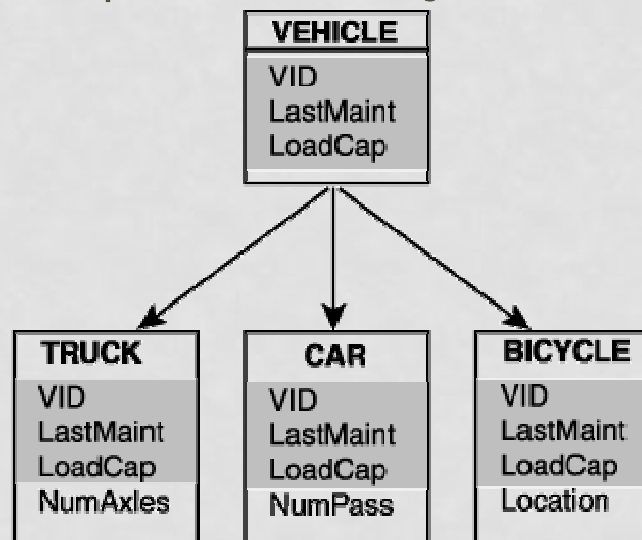
2. Cunoștințe care permit moștenirea

Cunoștințele despre obiecte, atribute și valorile acestora sunt deseori mult mai complexe decât permit reprezentările cu scheme de relații.

Una din cele mai utilizate forme de inferență este **moștenirea proprietăților**, prin care elementele care aparțin anumitor clase moștenesc atribute și valori provenite de la clase mai generale, în care sunt incluse.

TIPURI DE CUNOȘTINȚE

2. Cunoștințe care permit moștenirea



Pentru a face posibilă moștenirea proprietăților, **obiectele** trebuie să fie organizate în **clase**, iar clasele trebuie să fie organizate în cadrul unei **ierarhii**.

TIPURI DE CUNOȘTINȚE

2. Cunoștințe care permit moștenirea

În IA acest tip de cunoștințe stă la baza a două formalisme de reprezentare și raționament:

- rețele semantice
- baze de cunoștințe cu cadre

TIPURI DE CUNOȘTINȚE

1. Cunoștințe relaționale simple
2. Cunoștințe care permit moștenirea
3. Cunoștințe inferențiale
4. Cunoștințe procedurale

TIPURI DE CUNOȘTINȚE

3. Cunoștințe inferențiale

Moștenirea proprietăților este un mecanism de inferență puternic însă nu suficient pentru a putea construi un mecanism de raționament complet.

Este deci necesară, îmbogățirea acestor cunoștințe cu proceduri de inferență care implementează regulile logice de inferență.

Exista mai multe astfel de proceduri, unele folosesc raționamente “înainte”, raționamente “înapoi” sau mixte. O altă tehnică utilizată este cea a “rezoluției” care folosește strategia contradicției.

TIPURI DE CUNOȘTINȚE

1. Cunoștințe relaționale simple
2. Cunoștințe care permit moștenirea
3. Cunoștințe inferențiale
4. Cunoștințe procedurale

TIPURI DE CUNOȘTINȚE

4. Cunoștințe procedurale

Cunoștințele enumerate pana acum sunt formalisme ale **faptelor declarative**. Un alt tip de cunoștințe extrem de utile sunt **cunoștințele procedurale sau operaționale**, care specifică ce anume trebuie făcut și când.

Cea mai folosită tehnică de reprezentare a cunoștințelor procedurale în programele de IA este aceea a utilizării **regulilor de producție**.

TIPURI DE CUNOȘTINȚE

4. Cunoștințe procedurale. Reguli de producție

Regulile de producție sunt numite și reguli de tip **if-then**, care pot fi îmbogățite cu informații asupra modului în care pot fi folosite. Acestea sunt cunoștințe/instrucțiuni condiționale de tipul:

IF conditie THEN concluzie

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

CLASE DE METODE PENTRU REPREZENTAREA CUNOȘTINȚELOR

- **Logic based representation** – first order predicate logic, Prolog
- **Procedural representation** – rules, production system
- **Network representation** – semantic networks, conceptual graphs
- **Structural representation** – scripts, frames, objects

CLASE DE METODE PENTRU REPREZENTAREA CUNOȘTIȚELOR – MECANISME DE INFERENȚĂ

Reprezentările bazate pe logică aparțin unor două mari categorii de inferență, în funcție de instrumentele folosite:

- Logica – unde mecanismul principal îl constituie **inferența logică**
- Regulile (utilizate cu precădere în sistemele expert) – principalele mecanisme sunt **înlănțuirea înainte** și **înlănțuirea înapoi**.
- Reprezentări de tip slot-filler folosesc două categorii diferite de reprezentări
 - Retelele semantice și grafurile conceptuale – reprezentări de tip graf, principalul mecanism de inferență este **cautarea**.
 - Cadrele și scripturile – reprezentări structurate, principalul mecanism de inferență este **potrivirea sabloanelor**.

REPREZENTARI DE TIP OBIECTE STRUCTURATE

Rețelele semantice (semantic network SN): o reprezentare distribuită în care obiectele/conceptele sunt legate între ele prin diverse relații. SN exprimă cunoștințele sub forma relațiilor binare care există între concepte => pentru a înțelege un concept trebuie să știm cum se “leaga” de celelalte concepte ale domeniului de cunoaștere. SN sunt folosite și în lingvistică, ele fiind dezvoltate inițial pentru a reprezenta semnificația cuvintelor din limbajului natural. În acest caz, nodurile rețelei sunt asociate cuvintelor care descriu concepte, iar arcele descriu relațiile cu alte cuvinte care indică semnificația enunțului. De aici vine și denumirea de rețele semantice.

REPREZENTARI DE TIP OBIECTE STRUCTURATE

Cadre: o reprezentare structurată în care obiectele sunt organizate în grupuri de concepte legate prin relația de moștenire. Un cadru este un șablon general, în care datele noi sunt interpretate în termenii sau conceptele experienței dobândite anterior.

Se știe că oamenii nu interpretează noile situații construind de fiecare dată o structură nouă de cunoștințe, ci folosesc structuri memorate, derivate din experiențele anterioare. Pentru a analiza și înțelege o experiență nouă se caută șabloane deja existente care se particularizează în funcție de datele curente.

În plus, aceste reprezentări includ posibilitatea atașării de proceduri pentru calcularea valorilor unor anumite atribute în funcție de valorile altora.

BAZE DE CUNOȘTINȚE

O bază de cunoștințe poate fi privită ca o colecție de entități având o structură sintactică în concordanță cu o anumită **metodă de reprezentare a cunoștințelor**, astfel încât printr-un efectuarea unui calcul numit **raționament** sau **inferență** să se poată deduce noi informații cu privire la obiectele sau conceptele reprezentate în bază.

MOTORUL DE INFERENTA

Mecanismul de inferență urmărește o serie de obiective majore, cum ar fi:

- Alegerea strategiei de control în funcție de problema curentă;
- Actualizarea automata a elementelor din baza de cunostinte.

Motorul de inferenta este proiectat pentru a manipula cunostintele din BC insa nu depinde de continutul efectiv al acestora ci de modul de reprezentare al lor.

SISTEME BAZATE PE CUNOȘTINȚE

În sistemele simbolice de IA, operațiile de bază sunt reprezentate de prelucrările care manipulează cunoștințele din baza de cunoștințe. Astfel de sisteme se numesc sisteme bazate pe cunoștințe, caracteristica lor definitorie fiind că există o separare netă între baza de cunoștințe și modulul de inferență (de raționament).

Un același modul de inferență poate fi utilizat pentru baze de cunoștințe diferite, dacă toate acestea respectă un anumit formalism de reprezentare.

Una din cele mai grele probleme ale dezvoltării unui astfel de sistem este construirea bazei de cunoștințe, ce rezultă în urma procesului de achiziție de cunoștințe.

SISTEME BAZATE PE CUNOȘTINȚE

Deosebirea dintre un program clasic și unul bazat pe cunoștințe, constă în faptul că în primul caz cunoștințele sunt implicite, pe când în al doilea caz, ele sunt explicite.

Sistemele bazate pe cunoștințe sunt aplicații care rezolvă inteligent probleme complexe și în acest scop, în procesul de proiectare trebuie să se aibă în vedere două aspecte esențiale:

- găsirea unei reprezentări a cunoștințelor cât mai adecvată clasei de probleme
- construirea unui modul de prelucrare a cunoștințelor, numit și **motor de inferență**.

SISTEME DE REPRESENTARE ȘI PROCESARE A CUNOȘTINȚELOR

Din punct de vedere intuitiv, un sistem de reprezentare și procesare a cunoștințelor (SRPC) este o colecție de componente care cooperează astfel încât sistemul să fie capabil să raționeze, adică să dea un răspuns la o interogare.

Orice asemenea sistem utilizează o bază de cunoștințe și implicit o anumită metodă de reprezentare a cunoștințelor.

SRPC & BAZE DE CUNOȘTINȚE

Raportul dintre o bază de cunoștințe și un SRPC este următorul:

un SRPC nu conține o anumită bază de cunoștințe ci definește (printre altele) un formalism de reprezentare și rationament, deci implicit definește totalitatea bazelor de cunoștințe corecte din punctul de vedere al formalismelor pe care le reprezintă și pe care le poate procesa.

SRPC & SGBD

Pe cazul particular al bazelor de date, un SRPC devine un SGBD:

- la fel ca SGBD-urile, si SRPC-urile folosesc un singur formalism de reprezentare pentru bazele de date/baze de cunoștințe
- noțiunii de interogare de la SGBD îi corespunde noțiunea de deducție la SRPC

SISTEME DE REPREZENTARE ȘI PROCESARE A CUNOȘTINȚELOR

Sistemul $\mathbf{S} = (L_{KB}, L_Q, L_{Ans}, L_I, \vdash, \mathbf{Ans}, \mathbf{Upd})$ se numeste sistem de reprezentare și procesare a cunoștințelor, unde:

- L_{KB} este mulțimea tuturor bazelor de cunoștințe considerate corecte
- L_Q este limbajul de interogare
- L_{Ans} este limbajul de raspuns
- $\vdash$ este relatia de deductie realizata de S
- $\mathbf{Ans}: L_{KB} \times L_Q \rightarrow L_{Ans}$ este functia de raspuns
daca $K \vdash w$ atunci $\mathbf{Ans}(K, w)$ contine toate solutiile la
interogarea w pentru $K \in L_{KB}$
- $\mathbf{Upd}: L_{KB} \times L_I \rightarrow L_{KB}$ este functia de actualizare
 $K_2 = \mathbf{Upd}(K_1, w)$ pentru $K_1, K_2 \in L_{KB}$

Vă mulțumesc!