Str. Italiană nr. 28, sect. 2; sect. 2; București - România
weh@spiruharet.ro weh.spiruharet.ro

Categoria de servicii: ***WP 3. Servicii de dezvoltare a competențelor și de formare profesională (Skills Development and Training Services)***

Subcategoria de servicii: ***Sesiuni de instruire pe soluții digitale avansate (Training sessions on high performance digital solutions)***

SESIUNE DE INSTRUIRE PENTRU INTELIGENŢĂ ARTIFICIALĂ (AI)

- Metode fundamentale ale Inteligenței Artificiale -

Parteneri WEH: Universitatea *Spiru Haret*

Trainer de coordonare a transformarii digitale:

ALBEANU GRIGORE

CUPRINS

Str. Italiană nr. 28, sect. 2; sect. 2; București - România
weh@spiruharet.ro weh.spiruharet.ro

1. Introducere in inteligenta artificiala

Continuumul inteligenței mașinilor (MIC) prezintă diferitele tipuri de inteligență a mașinilor în funcție de complexitatea capacităților lor.

Nivelul cel mai de jos al continuumului de inteligență a mașinilor (MIC) conține "**sistemele care acționează**", pe care le definim ca automate bazate pe reguli. Acestea sunt sisteme care funcționează în conformitate cu un scenariu predefinit, adesea urmând reguli de tip "dacă-atunci" programate manual. Printre exemple se numără alarma de incendiu din casa dumneavoastră și sistemul de control al vitezei de croazieră din mașina dumneavoastră. Mașina nu este self-driving.

"**Sistemele care prezic**" sunt sisteme capabile să analizeze date și să le utilizeze pentru a produce predicții probabilistice. Rețineți că o "predicție" este o punere în corespondență a informațiilor cunoscute cu informații necunoscute și nu trebuie neapărat să fie un eveniment viitor. Statisticile alimentează cele mai multe sisteme de predicție, dar predicțiile sunt la fel de bune ca și datele utilizate. Dacă datele dvs. sunt eronate sau dacă alegeți date de eșantionare care nu reprezintă suficient de bine populația țintă, atunci veți obține rezultate eronate. În analiza afacerilor, lipsa integrității datelor și greșelile metodologice sunt extrem de frecvente și îi conduc adesea pe directori (decidenți) la concluzii greșite.

În timp ce "**sistemele care învață**" fac previziuni la fel ca și sistemele statistice, acestea necesită mai puțină inginerie manuală și pot învăța să îndeplinească sarcini fără a fi programate în mod explicit în acest sens. Învățarea automată (ML) și învățarea profundă = în profunzime (DL) conduc majoritatea acestor sisteme, iar acestea pot funcționa la nivel uman sau mai bine decât omul pentru multe probleme computaționale.

Învățarea poate fi automatizată la diferite niveluri de abstractizare și pentru diferite componente ale unei sarcini. Finalizarea unei sarcini necesită mai întâi achiziția de date care pot fi utilizate pentru a genera o predicție despre lume. Această predicție este combinată cu o judecată de nivel superior pentru a executa o acțiune. Rezultatul acestei acțiuni oferă un feedback măsurabil care poate fi reutilizat în punctele de decizie primare pentru a îmbunătăți performanța sarcinii.

Multe dintre aplicațiile de statistică și de învățare automată ale întreprinderilor se concentrează pe îmbunătățirea procesului de predicție. În vânzări, de exemplu, abordările bazate pe învățarea automată pentru evaluarea clienților potențiali pot avea performanțe mai bune decât metodele statistice sau bazate pe reguli. Odată ce mașina a produs o predicție cu privire la profilul unui client, vânzătorul aplică apoi judecata umană pentru a decide cum să dea curs cererii.

"**Sisteme care creează**" - Descoperirile recente în domeniul modelelor de rețele neuronale au inspirat o renaștere a creativității computaționale, computerele fiind acum capabile să producă scrieri originale, imagini, muzică, desene industriale și chiar software de inteligență artificială. https://www.research.autodesk.com/projects/project-dreamcatcher/

"**Sisteme care relaționează**" - Analiza sentimentelor, cunoscută și sub numele de opinion mining sau emotion AI, extrage și cuantifică stările emoționale din textul, vocea, expresiile faciale și limbajul corpului. Cunoașterea stării afective a unui utilizator permite calculatoarelor să răspundă empatic și dinamic, așa cum fac prietenii noștri.

"**Sisteme care stăpânesc**" - Pentru că suntem Sisteme care stăpânesc, oamenii nu au nicio problemă cu acest lucru. Un sistem care stăpânește este un agent inteligent capabil să construiască concepte abstracte și planuri strategice din date puține. Prin crearea unor reprezentări conceptuale modulare ale lumii din jurul nostru, suntem capabili să transferăm cunoștințe dintr-un domeniu în altul, o caracteristică cheie a inteligenței generale. Niciun sistem modern de inteligență artificială nu este o AGI, adică o inteligență generală artificială. În timp ce oamenii sunt Sisteme care stăpânesc, programele actuale de inteligență artificială nu sunt.

"**Sisteme care evoluează**" - se referă la sistemele care dau dovadă de inteligență și capacități supraomenești, cum ar fi capacitatea de a-și schimba în mod dinamic propriul design și arhitectura pentru a se adapta la condițiile schimbătoare din mediul lor.

1.1. Inteligenţa si inteligenta artificiala

Inteligența artificială este știința de a construi mașini care să facă lucruri ce ar necesita inteligență dacă ar fi făcute de oameni.

Inteligența este mai greu de definit, fiind un termen generic pentru multe capacități înrudite:

- capacitatea de a raționa, a planifica, a rezolva probleme, a gândi abstract, a înțelege idei complexe,
- a învăța repede și a învăța din experiență.

Teoria Inteligenţelor multiple: Psihologul Howard Gardner a identificat 7 tipuri distincte de inteligență (Frames of Mind, 1983) și mai recent încă 2 (1995):

- Lingvistică: Se referă la capacitatea și plăcerea de a citi, scrie, povesti sau rezolva cuvinte încrucișate;

- Logico-matematică: Presupune descoperirea modelelor, categoriilor și relațiilor. Se manifestă, de exemplu, în rezolvarea problemelor aritmetice sau în jocurile de strategie;

- Spațială: Se referă la posibilitatea de a gândi în imagini și la ușurința rezolvării unor probleme geometrice spațiale;

- Corporal-chinestezică: Implică o mare sensibilitate în identificarea și prelucrarea senzațiilor fizice, de exemplu, a simți ritmul unui dans;

- Muzicală: Presupune existența urechii muzicale, a posibilității de a percepe și distinge sunete care par la fel altor persoane

- Interpersonală: Este dovedită de spiritul de conducător, de ușurința comunicării și de existența empatiei, adică a capacității de a înțelege sentimentele altora;

- Intrapersonală: Reflectă o bună cunoaștere a propriilor sentimente și posibilități;

- Naturală: Capacitatea de a recunoaște modele în natură, de a înțelege diferite specii și de a clasifica obiecte naturale: biologi etc.

- Existențială: Capacitatea de a trata problemele filosofice ale vieții: scriitori, filosofi, oameni cărora le place să citească și să-și pună întrebări, predicatori etc.

Într-un interviu, în 2016, Gardner a menționat că ia în calcul adăugarea inteligenței pedagogice.

1.2. Definitii ale inteligentei artificiale

Inteligența este o măsură a capacității de a atinge scopuri (de a rezolva probleme) într-un mediu complex și dinamic.

Există patru direcții de abordare a inteligenței artificiale:

* **A acționa omenește**: De exemplu, testată cu testul Turing. Un arbitru, om, se angajează întro conversație în limbaj natural cu alți doi participanți la experiment, unul om și altul mașină. Dacă arbitrul nu poate spune cu siguranță cine este omul și cine este mașina, aceasta se spune că a trecut testul. Contraexemplu: camera chinezească a lui Searle. Un om folosește mulțimea de reguli ale unui calculator care trece testul Turing pentru a răspunde la întrebări în limba chineză. Omul nu cunoaște această limbă. Un program care manipulează simboluri, chiar dacă se comportă inteligent, nu înțelege, nu are stări mentale și intenții;

* **A gândi omenește**: Construirea de sisteme care funcționează intern în mod similar cu mintea omenească, nu doar să rezolve probleme, ci să le rezolve în același mod ca oamenii. Se folosesc, de exemplu, rezultate din științele cognitive. O teorie este că mintea este o colecție de agenți care reprezintă procese diferite, posibil concurente;

* **A gândi rațional**: Folosirea reprezentărilor logice, cu avantajele formalizării și puterii de raționament. Dificultățile sunt reprezentarea cunoștințelor nesigure sau informale și învățarea din date afectate de zgomot;

* **A acționa rațional**: A descoperi acțiunea optimă, care aduce utilitatea maximă, indiferent de natura prelucrărilor interne. Acțiunile raționale sunt studiate de majoritatea cercetărilor actuale, deoarece comportamentul este observabil și mai ușor de testat științific decât gândirea, iar raționalitatea este clar definită.

Pentru a crea inteligența artificială generală, există o serie de direcții potențiale care ar trebui agregate într-un mecanism unitar:

-Metode conexioniste (rețele profunde);

-Metode simbolice (metode logice, ontologii);

-Metode probabilistice (rețele bayesiene);

-Metode evolutive (algoritmi genetici/evolutivi);

-Metode de analogie (învățare prin transfer).

Str. Italiană nr. 28, sect. 2; sect. 2; București - România
weh@spiruharet.ro weh.spiruharet.ro

1.3. Istoricul inteligentei artificiale

- 1943: primul model de neuron artificial (McCulloch & Pitts)
- 1949: învățarea neuronală (Hebb)
- 1950, 1951: studii asupra jocului de șah (Shannon, Turing)
- 1951: SNARC, primul calculator neuronal (Minsky & Edmonds)
- 1956: Termenul inteligență artificială: McCarthy, workshop-ul de la Dartmouth College
- Logical Theorist (Newell & Simon)
- Entuziasmul timpuriu

  + 1957: General Problem Solver

  + 1958: McCarthy: limbajul Lisp, time sharing, Advice Taker (teoretic)

  + 1959: Geometry Theorem Prover (Gelertner)

  + 1965: Metoda rezoluției (Robinson)

  + Program de dame care învață din jocuri

  + Program pentru analiză matematică
- Rosenblatt (1957)

  + Clasificator liniar

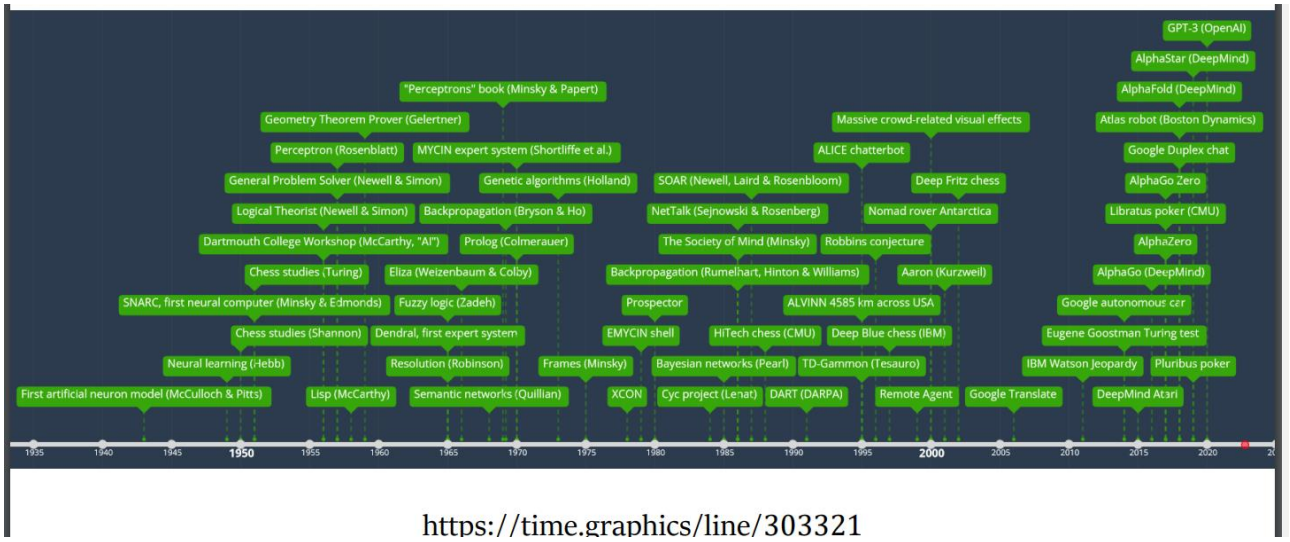
  + Teorema de convergență a perceptronului (1962)

  + Algoritm de antrenare
- Dendral, primul sistem expert (1965) - Sistemele bazate pe cunoștințe
- Psihiatrul computerizat (Weizenbaum & Colby, 1966) – ELIZA
- 1965: Logica vagă / fuzzy (Zadeh) ν 1968: Reprezentarea cunoașterii prin rețele semantice (Quillian) ν 1970: Limbajul Prolog (Colmerauer) ν 1970: Sistemul expert MYCIN (Shortliffe et al.) ν Scris în Lisp, 450 reguli ν Infecții sangvine, explicații (“why?”), factori de încredere ν EMYCIN (1979), shell de sisteme expert ν 1973: Algoritmii genetici (Holland) ν 1975: Reprezentarea cunoașterii prin cadre (Minsky) ν 1980: Prospector, detectarea depozitelor minerale
- 1984: Proiectul Cyc (Lenat), colecție de cunoștințe de „bun simț”; în 2006: 47.000 concepte și 306.000 fapte
- 1985: Rețele bayesiene (Pearl), raționament probabilistic

Str. Italiană nr. 28, sect. 2; sect. 2; București - România
weh@spiruharet.ro weh.spiruharet.ro

- 1986: Se impune algoritmul backpropagation (Rumelhart, Hinton & Williams) pentru antrenarea rețelelor neuronale de tip perceptron multi-strat, descoperit de fapt în 1969 (Bryson & Ho)
- 1987: SOAR (Newell, Laird & Rosenbloom)
- 1988: Cartea Societatea minții (Minsky)
- 1988: HiTech (CMU) îl învinge pe marele maestru de șah Denker
- Rețea neuronală care învață să citească texte în engleză (Sejnowski & Rosenberg, 1986) - 309 neuroni, 18 629 conexiuni
- 1991: În timpul războiului din Golf, planificarea trupelor ce implicau 50.000 de vehicule militare, transportatoare și trupe, a fost realizată cu ajutorul unui sistem de IA, DART
- 1995: TD-Gammon (Tesauro), joc de table
- 1995: ALVINN, autovehicul autonom (4585 km prin SUA)
- 1995: A.L.I.C.E. chatterbot
- 1997: Deep Blue, dezvoltat de IBM, l-a învins pe campionul mondial Gari Kasparov
- 1999: Sistem expert în timp real pentru gestionarea sarcinilor de rutină de reparare ale unei sonde spațiale (Remote Agent pentru Deep Space 1)
- 2002: Kramchik și Deep Fritz fac remiză
- 2006: Google Translate, acum peste 100 de limbi
- 2007: Urban Challenge, mașini autonome, cu premiu de 2 milioane $ (96 km prin zone urbane)
- 2009: Google introduce subtitrări automate în videoclipurile YouTube
- 2014: Eugene Goostman / Evghen Gustman – un program care pretinde a fi un băiat ucrainian de 13 ani, trece testul Turing (păcălește 33% din arbitri)
- 2015. Învățare cu întărire profundă - Google DeepMind (2015) - Programul a învățat să joace 49 de jocuri Atari 2600 urmărind direct doar afișajul și scorul
- AlphaGo - 2017: a câștigat împotriva lui Ke Jie, cel mai bun jucător de go din lume. AlphaGo Zero (2017) a învățat go jucând cu el însuși, fără a învăța din mutările jucătorilor experți umani
- AlphaGo Zero a învins AlphaGo (100-0)

- AlphaZero a învățat șah în 4 ore și a învins motorul de șah Stockfish (28 victorii, 72 remize, 0 înfrângeri)
- Inteligența artificială a atins abilități superioare oamenilor pentru practic toate jocurile
- Generare de texte și imagini: OpenAI – partener Microsoft
- GPT-2 (2019); GPT-3 (2020), https://app.inferkit.com/demo; ChatGPT (2022), https://chat.openai.com; DALL-E (2021); DALL-E 2 (2022), https://openai.com/dall-e-2; DALL-E 3 (2023)



https://time.graphics/line/303321

1.4. Concluzii

Inteligența este o măsură a capacității de a atinge scopuri (de a rezolva probleme) într-un mediu complex și dinamic. IA Reprezintă o mulțime de strategii folosite pentru a optimiza interacțiunea cu mediul.

În IA există abordări bazate pe gândire sau comportament, respectiv modelarea minții umane sau raționalitate.

S-au făcut progrese substanțiale în:

- Recunoașterea modelelor și învățare
- Recunoașterea imaginilor și limbajului
- Problemele de planificare și raționament
- Multe probleme sunt încă nerezolvate
- IA-ul este un domeniu de cercetare interesant!

2. Metode de cautare

2.1. Rezolvarea problemelor prin cautare

15 puzzle

Turnurile din Hanoi

Găsirea rutelor

Navigarea roboților

Căutare în lățime: cele mai scurte căi în rețele sociale, web crawling

 Căutare în adâncime: parcurgerea unor componente ierarhice ordonate topologic

 Descoperirea fișierelor dintr-o structură de directoare

 Compilare inteligentă în proiecte mari, ținând seama de dependențe: detecția ciclurilor

2.2. Cautarea neinformata (oarba)

Căutarea limitată în adâncime presupune o căutare în adâncime cu limitare la k niveluri. Sub această adâncime, nodurile nu mai sunt expandate.

Căutarea iterativă în adâncime realizează succesiv căutări limitate în adâncime pentru $k = 0, 1, 2, \ldots$ până se găsește o soluție. Căutarea iterativă în adâncime combină avantajele căutărilor în lățime și adâncime: este completă și optimă, iar în cazul cel mai defavorabil , complexitatea de timp este $O(b^d)$, comparabilă cu a căutării în lățime, puțin mai lentă din cauza repetițiilor, iar complexitatea de spațiu este $O(b \cdot d)$, foarte bună, comparabilă cu a căutării în adâncime.

Căutarea bidirecțională în lățime are complexitate mult mai bună decât căutarea în lățime: $O(b^{d/2}) \ll O(b^d)$, însă operatorii din căutarea înapoi pot fi mai dificil de identificat decât în căutarea înainte.

Căutarea de cost uniform ordonează frontiera în funcție de distanța nodurilor față de starea inițială, dată de funcția g. Este utilizată pentru grafuri ponderate, în care muchiile pot avea costuri diferite.

2.3. Cautarea informata (euristica)

O *euristică* este o metodă care furnizează rapid o soluție, nu neapărat optimă. Este o metodă aproximativă, spre deosebire de un algoritm exact optim. Deși nu garantează găsirea soluției optime, metodele euristice găsesc de obicei o soluție acceptabilă, deseori chiar soluția optimă.

Metodele euristice de căutare utilizează cunoștințe specifice fiecărei probleme pentru a ordona nodurile, astfel încât cele mai „promițătoare" noduri sunt plasate la începutul frontierei.

Funcții utilizate în general:

- $f(n)$ este un cost estimat. Cu cât este mai mic $f(n)$, cu atât este mai bun nodul n;
- $g(n)$ este costul căii de la nodul inițial la n. Este cunoscută;
- $h(n)$ este estimarea costului căii de la n la un nod scop. Este o estimare euristică.

Tipuri de căutare:

- Căutarea de cost uniform (neinformată): $f(n) = g(n)$;
- Căutarea greedy: $f(n) = h(n)$;
- Căutarea A*: $f(n) = g(n) + h(n)$. Reunește ideile căutărilor de cost uniform și greedy.

Pentru a garanta găsirea soluției optime, funcția h trebuie să fie *admisibilă*: niciodată mai mare decât costul real. Pentru metodele euristice de căutare, problema principală este găsirea celei mai bune funcții h, care să dea estimări cât mai apropiate de realitate. Cu cât h este mai bună, numărul de noduri expandate este mai mic.

2.4. Concluzii

Metodele neinformate nu utilizează cunoștințe specifice fiecărei probleme și deci tratează toate problemele la fel

Metodele informate sau euristice utilizează cunoștințe specifice fiecărei probleme și expandează mai întâi cele mai promițătoare noduri.

Funcțiile euristice care direcționează căutarea pot fi construite prin simplificarea problemei inițiale, adică eliminarea unor constrângeri.

3. Metode de inferenta in logica propozitionala si predicativa

3.1. Logica propozitionala

Teorema de completitudine a calculului propozițional spune că în calculul propozițional mulțimea teoremelor coincide cu mulțimea tautologiilor. Noțiunea de teoremă este de natură sintactică, în timp ce noțiunea de tautologie are o natură semantică. Teorema subliniază faptul că aceste noțiuni sunt echivalente: orice tautologie poate fi dedusă pe cale sintactică și orice propoziție întotdeauna adevărată este o teoremă și poate fi folosită ulterior pentru inferențe.

Dintre modalitățile clasice de inferență propozițională amintim:

- *Modus Ponens:* premise: $P \rightarrow Q$ și P, concluzie: Q;
- *Modus Tollens:* premise: $P \rightarrow Q$ și non Q, concluzie: non P.

Rezoluția propozițională este o regulă puternică de inferență pentru logica propozițională, cu ajutorul căreia se poate construi un demonstrator de teoreme corect și complet. Ea poate fi aplicată însă doar după aducerea premiselor și concluziei într-o formă standardizată, numită *formă normal conjunctivă* (FNC).

Transformarea în FNC are următoarele etape:

1. Se înlocuiesc relațiile de implicație și echivalență;
2. Se introduc negațiile în paranteze;
3. Se mută conjuncțiile în afara disjuncțiilor.

3.2. Logica predicatelor

Pentru *logica predicatelor de ordin întâi*, o teoremă se poate demonstra prin raționament înainte, raționament înapoi (dacă clauzele sunt implicații) sau prin rezoluție predicativă (pentru clauze generale).

Pentru toate aceste metode, clauzele trebuie aduse de asemenea în forma normal conjunctivă. Transformarea în FNC are următoarele etape:

1. Se înlocuiesc relațiile de implicație și echivalență;
2. Se introduc negațiile în paranteze și peste cuantificatori;
3. Se standardizează variabilele, astfel încât fiecare cuantificator să aibă propria variabilă;
4. Se mută toți cuantificatorii la stânga formulei, fără a le schimba ordinea relativă;
5. Se elimină cuantificatorii existențiali, fiind înlocuiți cu funcții Skolem;
6. Se elimină prefixul rămas de cuantificatori universali;
7. Se mută conjuncțiile în afara disjuncțiilor;
8. Fiecare termen disjunctiv se consideră o propoziție separată;
9. Se standardizează din nou variabilele între termenii disjunctivi distincți.

3.3. Despre PROLOG
(https://athena.ecs.csus.edu/~mei/logicp/Programming_in_Prolog.pdf, https://silp.iiita.ac.in/wp-content/uploads/PROLOG.pdf)

```
fib( 1, 1).

fib( 2, 1).

fib( N, F) :-
  N > 2,
  N1 is N-1, fib( N1, F1),
  N2 is N-2, fib( N2, F2),
  F is F1 + F2.
```

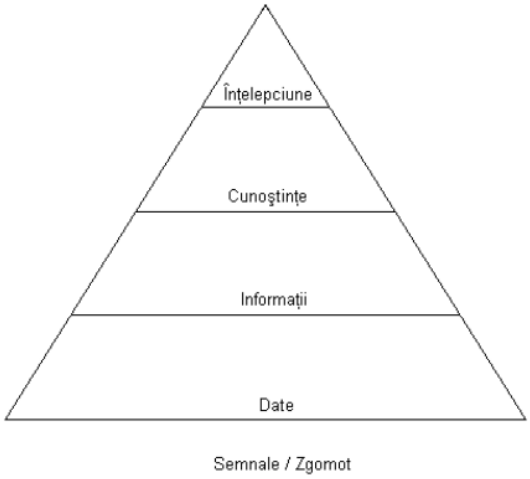
Exemplu de program care verifică dacă un număr aparţine şirului lui Fibonacci (Codul lui Da Vinci!)

Concluzie. Demonstrarea automată a teoremelor face apel la Logica matematică şi computaţională.

4. Reprezentarea cunoasterii
4.1. Reprezentarea simbolica

Ierarhia cunoașterii

În funcție de nivelul de complexitate semantică, cunoașterea poate fi structurată pe mai multe niveluri, sub forma unei piramide:



Semnalele nu au semnificație simbolică. De exemplu, o scriere necunoscută.

Datele sunt secvențe de simboluri cu sens, cu valoare pur sintactică. De exemplu: „Temperatura este de 10 grade. Plouă.”

Informațiile sunt date acumulate într-un context cu semnificație, având caracteristici semantice. De exemplu: „Temperatura a scăzut cu 10 grade și apoi a început să plouă.”

Cunoștințele reprezintă un domeniu extins de aplicare a informațiilor. De exemplu: „Dacă umiditatea este foarte mare și temperatura scade mult, atmosfera nu mai poate reține vaporii de apă și deci începe să plouă."

Înțelepciunea presupune înțelegerea principiilor, valorilor. De exemplu: „Cel mai bun lucru pe care îl poți face când plouă este să lași ploaia să-și urmeze cursul." / "The best thing one can do when it rains is to let it rain." (H. W. Longfellow)

Înțelegerea se poate defini din punct de vedere comportamental ca abilitatea de a aplica un concept sau o procedură. Din punct de vedere cognitiv, se poate defini ca având mai multe perspective asupra unui concept. Mai multe perspective cresc șansele de aplicare în mai multe situații sau domenii.

4.2. Sisteme expert

Cunoștințele unui nespecialist despre un fenomen sunt globale, amorfe, nestructurate. Pentru un expert, cunoștințele despre același fenomen sunt organizate, precise, punctuale și sistematizate. Un domeniu de expertiză poate avea între 50.000 și 100.000 de piese de cunoaștere specifice ($\approx 70.000 \pm 20.000$).

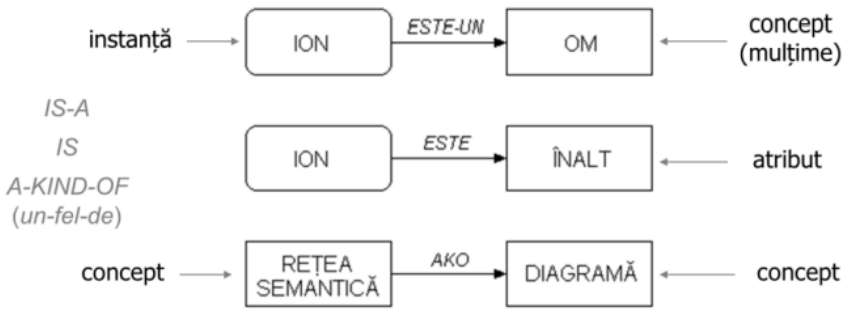
Un *sistem expert* este un program care utilizează cunoștințe și proceduri de inferență pentru a rezolva probleme suficient de dificile pentru a necesita în mod normal intervenția unui expert uman în vederea găsirea soluției. Pe scurt, sistemele expert sunt programe care înmagazinează cunoștințe specializate, provenite de la experți.

Un sistem expert conține trei module principale, care formează așa-numitul *sistem esențial*: *baza de cunoștințe* (descrierea elementelor și relațiilor), *motorul de inferență* (care aplică procedurile de căutare, construiește raționamentul și este independent de baza de cunoștințe) și *baza de fapte* (memoria auxiliară sau temporară).

4.3. Retele semantice

O *rețea semantică* este o modalitate grafică de reprezentare a cunoașterii în modele de noduri, semnificând concepte, interconectate prin arce etichetate, care precizează relațiile dintre concepte.

Rețelele semantice simple folosesc etichete care descriu relațiile fundamentale dintre concepte, instanțe și atribute:

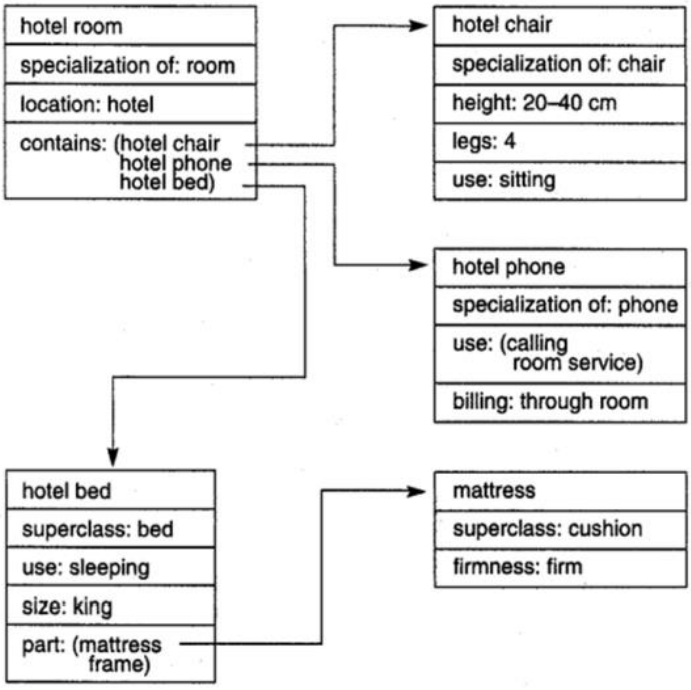


Str. Italiană nr. 28, sect. 2; sect. 2; București - România
weh@spiruharet.ro weh.spiruharet.ro

4.4. Reprezentarea prin cadre

Teoria cadrelor (engl. "frame theory") propune o reprezentare în care sunt cuprinse atât informații declarative cât și procedurale pentru reprezentarea *situațiilor stereotipe*. Un *cadru* este un șablon general, în care datele noi sunt interpretate în termenii sau conceptele experienței dobândite anterior.

De exemplu, pentru o cameră de hotel, reprezentarea prin cadre este următoarea:



4.5. Reprezentarea prin scenarii

Un *scenariu* (engl. "script") este o modalitate de reprezentare a cunoașterii ce descrie o *secvență stereotipă de evenimente* într-un anumit context, adică situații care se repetă, păstrând aceeași structură.

De exemplu, pentru un restaurant:

Name: *Restaurant*
Props: Tables, Menu, Food, Bill, Money, Tip
Roles: Customer, Waiter, Cook, Cashier, Owner
Entry Conditions: Customer hungry; Customer has money
Results: Customer has less money; Owner has more money; Customer is not hungry

Elementele unui scenariu: condiții de intrare, rezultate, proprietăți (lucrurile care „sprijină" desfășurarea scenariului, de exemplu: într-un restaurant există mese, scaune, meniuri), roluri (acțiunile pe care le îndeplinesc participanții, de exemplu: chelnerul ia comanda, clientul plătește), scene (împărțirea scenariului pe aspecte temporale, de exemplu: intrarea în restaurant, comanda, luarea mesei).

Str. Italiană nr. 28, sect. 2; sect. 2; București - România
weh@spiruharet.ro weh.spiruharet.ro

4.6. Ontologii

O *ontologie* reprezintă o mulțime finită de obiecte și concepte (sau clase), împreună cu relațiile dintre ele, inclusiv ierarhiile de clase.

Web Ontology Language (OWL) este un limbaj de tip XML pentru definirea și instanțierea ontologiilor web. Permite descrierea claselor și reprezentarea proprietăților și instanțelor.

Semantic Web este o idee de web în care cunoștințele pot fi gestionate în mod automat.

4.7. Concluzii

Inteligența în general și inteligența artificială în special necesită cunoaștere

Reprezentarea cunoașterii poate fi dificilă: cunoașterea este voluminoasă, greu de caracterizat cu precizie și în permanentă schimbare

Modalități tradiționale de reprezentare a cunoașterii sunt:

- Rețelele semantice: apropiate de limbajul natural, permițând ierarhizarea conceptelor
- Cadrele: pentru situații stereotipe
- Scenariile: pentru secvențe de evenimente stereotipe
- O ontologie reprezintă o mulțime de obiecte și concepte (sau clase), împreună cu relațiile dintre ele

5. Metode de rationament cu informatii incomplete

- Incompletitudinea unei informații se exprimă pe două scări:
  - Scara impreciziei
    - Conținutul informațional
    - Informație precisă, cu o singură valoare, sau nu
  - Scara incertitudinii
    - Încrederea care i se acordă informației
    - Informație certă sau nu

5.1. Logica clasica si logica fuzzy

Logica tradițională consideră că un obiect poate aparține sau nu unei mulțimi. *Logica fuzzy* permite o interpretare mai flexibilă a noțiunii de apartenență. Astfel, un obiect poate aparține mai multor mulțimi în grade diferite.

Mulțimile fuzzy permit elementelor să aparțină parțial unei clase sau mulțimi. Fiecărui element i se atribuie un *grad de apartenență* la o mulțime. Acest grad de apartenență poate lua valori între 0 (nu aparține mulțimii) și 1 (aparține total mulțimii). În cazul în care gradul de apartenență ar fi doar 0 sau 1, mulțimea fuzzy ar fi echivalentă cu o mulțime binară.

Fie X *universul discursului*, cu elemente notate x. O mulțime fuzzy A a universului de discurs X este caracterizată de o funcție de apartenență $\mu_A(x)$ care asociază fiecărui element x un grad de apartenență la mulțimea A:

$$\mu_A(x) : X \rightarrow [0,1]$$

De exemplu, pentru variabila lingvistică „tânăr", avem universul discursului $X = \{0, 20, 30, 50\}$ și următoarea funcție de apartenență: $A = 0/1 + 0{,}9/20 + 0{,}7/30 + 0/50$, cu semnificația: o persoană de 20 de ani aparține mulțimii oamenilor tineri în proporție de 90%, una de 30 de ani în proporție de 70% iar una de 50 de ani nu aparține mulțimii (gradul său de apartenență este 0). Aceste lucruri se reprezintă grafic ca în figura 1.

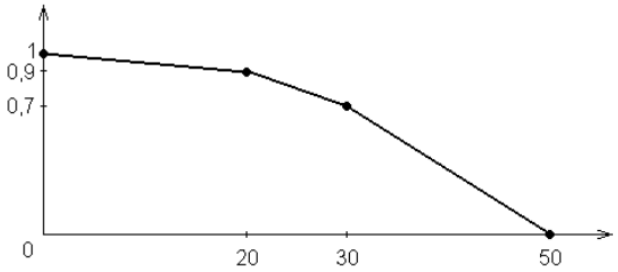


Figura 1. *Funcție de apartenență pentru mulțimea oamenilor tineri*

Pentru a reprezenta o mulțime fuzzy, trebuie să-i definim mai întâi funcția de apartenență. În acest caz, o mulțime fuzzy A este complet definită de mulțimea tuplelor:

$$A = \{(x, \mu_A(x)) \mid x \in X\}$$

Dacă X este o mulțime finită $X = \{x_1, \ldots, x_n\}$, atunci se mai folosește notația:

$$A = \mu_1 / x_1 + \ldots + \mu_n / x_n$$

5.2. Operatii cu multimi vagi

Fie A o submulțime fuzzy a universului de discurs X. Se numește *suportul* lui A submulțimea strictă a lui X ale cărei elemente au grade de apartenență nenule în A:

$$supp(A) = \{x \in X \mid \mu_A(x) > 0\}.$$

Înălțimea lui A se definește drept cea mai mare valoare a funcției de apartenență:

$$h(A) = \sup_{x \in X} \mu_A(x).$$

Se numește *nucleul* lui A submulțimea strictă a lui X ale cărei elemente au grade de apartenență unitare în A:

$$n(A) = \{x \in X \mid \mu_A(x) = 1\}.$$

Fie A și B submulțimi fuzzy ale lui X. Spunem că A este o *submulțime* a lui B dacă: $\mu_A(x) \le \mu_B(x), \forall x \in X$.

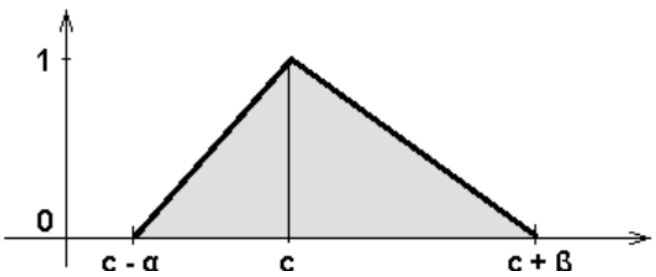
De multe ori, oamenii nu pot caracteriza precis informațiile numerice, folosind formulări precum „aproape 0”, „în jur de 100” etc. În teoria mulțimilor fuzzy, aceste numere pot fi reprezentate ca submulțimi fuzzy ale mulțimii numerelor reale.

Un *număr fuzzy* este o mulțime fuzzy a mulțimii numerelor reale, cu o funcție de apartenență convexă și continuă și suport mărginit.

O mulțime fuzzy A se numește *număr fuzzy triunghiular* cu centrul c, lățimea la stânga $\alpha > 0$ și lățimea la dreapta $\beta > 0$ dacă funcția sa de apartenență are forma:

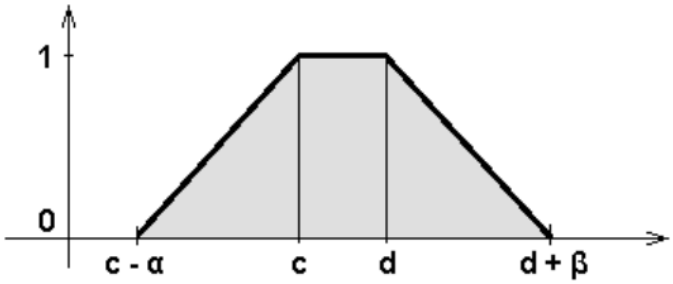
$$\mu_A(x) = \begin{cases} 1 - \dfrac{c - x}{\alpha}, & c - \alpha \le x \le c \\ 1 - \dfrac{x - c}{\beta}, & c < x \le c + \beta \\ 0, \text{ altfel} \end{cases}$$

Folosim notația: $A = (c, \alpha, \beta)$. Este evident că $supp(A) = (c - \alpha, c + \beta)$. Semnificația unui număr fuzzy triunghiular cu centrul c este „x este aproximativ egal cu c”.



O mulțime fuzzy A se numește *număr fuzzy trapezoidal* cu intervalul de toleranță $[c,d]$, lățimea la stânga $\alpha > 0$ și lățimea la dreapta $\beta > 0$ dacă funcția sa de apartenență are forma:

$$\mu_A(x) = \begin{cases} 1 - \dfrac{c-x}{\alpha}, c-\alpha \le x \le c \\ 1, c < x \le d \\ 1 - \dfrac{x-d}{\beta}, d < x \le d+\beta \\ 0. \text{ altfel} \end{cases}$$



5.3. Tipuri de inferenta

În logica fuzzy și raționamentul aproximativ, cea mai importantă regulă de inferență este *Modus Ponens generalizat*.

În logica clasică, această regulă de inferență este de forma $(p \wedge (p \to q)) \to q$, adică:

regulă: dacă *p,* atunci *q*
premisă: *p*
concluzie: *q*

În logica fuzzy, regula de inferență corespunzătoare este următoarea:

regulă: dacă *x* este *A*, atunci *y* este *B*
premisă (antecedent): *x* este *A'*
concluzie (consecvent): *y* este *B'*, unde $B' = A' \circ (A \to B)$.

Dacă $A' = A$ și $B' = B$, regula se reduce la Modus Ponens clasic.

Matricea $A \to B$ deseori se notează cu R. Procesul de inferență fuzzy este văzut ca o transformare a unei mulțimi fuzzy într-o altă mulțime fuzzy. R are semnificația unei matrici de posibilități condiționate ale elementelor din A și B:

$$R = \Pi_{B|A} = \begin{vmatrix} a_1 \to b_1 & a_1 \to b_2 & \dots \\ a_2 \to b_1 & \dots & \\ \dots & & \end{vmatrix}$$

Inferența Mamdani cu reguli multiple

Un exemplu de sistem de inferență fuzzy de tip Mamdani este prezentat în figura 8. Pentru a calcula ieșirea acestui sistem când se dau intrările trebuie parcuși următorii 6 pași:

1. Se determină o mulțime de reguli fuzzy;
2. Se realizează fuzzificarea intrărilor utilizând funcțiile de apartenență;
3. Se combină intrările fuzzificate urmând regulile fuzzy pentru stabilirea puterilor de activare ale regulilor;
4. Se calculează consecvenții regulilor prin combinarea puterilor de activare ale regulilor cu funcțiile de apartenență ale ieșirilor;
5. Se combină consecvenții pentru a determina mulțimea de ieșire;
6. Se defuzzifică mulțimea de ieșire, doar dacă se dorește ca ieșirea să fie strictă.
   În cele ce urmează se prezintă o descriere detaliată a acestui proces.

După ce am determinat mulțimea fuzzy indusă de o regulă de inferență, în unele aplicații trebuie obținută o valoare singulară, strictă, pe baza acestei mulțimi. Procesul se numește *defuzzificare*. Cea mai utilizată tehnică de defuzzificare este *metoda centrului de greutate* (sau a *centroidului*):

$$x_{CG} = \frac{\sum_i x_i \cdot \mu_A(x_i)}{\sum_i \mu_A(x_i)}$$

5.4. Sisteme expert fuzzy

- Un centru de service păstrează componente de schimb și repară componente defecte
- Clienții aduc o componentă defectă și primesc o piesă de schimb de același tip
- Componentele defecte sunt reparate și repuse în circuit
- Obiectivul sistemului expert este de a ajuta managerul în luarea deciziilor, astfel încât clienții să rămână mulțumiți

Procesul de dezvoltare al unui sistem expert fuzzy

- Specificarea problemei și definirea variabilelor lingvistice
- Determinarea mulțimilor fuzzy
- Construirea regulilor fuzzy
- Codarea mulțimilor, regulilor și procedurilor de inferență
- Evaluarea și rafinarea sistemului

Str. Italiană nr. 28, sect. 2; sect. 2; București - România
weh@spiruharet.ro weh.spiruharet.ro

Există 4 variabile lingvistice:

- Timpul mediu de așteptare (întârzierea medie) m
- Factorul de utilizare a reparațiilor ρ
  - ρ = număr clienți veniți / număr clienți plecați
- Numărul de angajați s
- Numărul inițial de piese de schimb n
  - Aceasta este ieșirea sistemului
  - Trebuie determinat: $n = f(m, \rho, s)$

6. Concluzii

Alte aplicații din lumea reală

- Subsisteme de autovehicule, transmisie automată, ABS
- Control automat al trenului monorail din Tokyo
- Aparate de aer condiționat
- Motorul de animație Massive
- Aparate de fotografiat
- Prelucrarea imaginilor (de exemplu, detecția muchiilor)
- Recunoașterea modelelor
- Lifturi
- Mașini de spălat vase
- Mașini de spălat, alte electrocasnice
- Filtre de limbaj pe forumuri și camere de discuții
- Jocuri video
- Microcontrolere și microprocesoare (de exemplu, Freescale)