

INTELIGENȚĂ ARTIFICIALĂ 2012 - 2013

CURS 1

Capitolul 1: Introducere în I.A.

I.A. este un domeniu al Informaticii care are ca scop dezvoltarea unor mașini, calculatoare, "inteligente", adică care să fie capabile să rezolve probleme pentru care metodele clasice de rezolvare, de calcul pur și simplu, nu sunt eficiente și/sau aplicabile.

O astfel de definiție pune mai multe întrebări decât oferă răspunsuri. Ce înțelegem prin metode convenționale?

O metodă convențională de rezolvare a unei probleme presupune respectarea unui algoritm, adică se urmărește o secvență fixă de pași, chiar dacă această secvență include și salturi, dar și acestea sunt prestabilite.

Să luăm ca exemplu rezolvarea ecuației de gradul doi: $ax^2+bx+c=0$, x necunoscută, a , b , c constante. De câteva secole formula de rezolvare este :

$$x_1, x_2 = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Un program va cere operatorului, de la consolă, valorile constantelor a , b , c , apoi va calcula după formulă valorile numerice ale rădăcinilor și în final va afișa aceste valori pe display. Un astfel de program nu se încadrează în ceea ce se numește Inteligență Artificială, oricare ar fi viteza sau precizia de calcul. Totuși nu trebuie înțeles din acest exemplu că opusul inteligenței artificiale ar fi "prostia naturală". Programul de mai sus constă în derularea parametrizată a unei secvențe fixe de operații elementare: adunare, scădere, înmulțire, extragere rădăcină pătrată.

I.A. este sinonimă unei abordări nealgoritmice. Ea se aplică acolo unde spațiul cazurilor și soluțiilor posibile este atât de vast încât nu se poate imagina o secvență fixă de operații elementare care să permită găsirea soluției problemei. Putem da un exemplu istoric: problema jocului de șah. Numărul situațiilor posibile este de ordinul 10^{120} , ceea ce dă o imagine despre ceea ce ar însemna o abordare algoritmică și ce șanse de câștigare a jocului ar avea un astfel de program.

Ceea ce se dorește în Inteligența Artificială este să se programeze o deprindere de a "gândi", "a raționa" și deci de a permite programului să dezvolte automat algoritmul adecvat diferitelor situații întâlnite. Termenul de "raționament" implică două concepte fundamentale.

- reprezentarea cunoștințelor relative la domeniul de aplicații avut în vedere. Termenul "reprezentare" subînțelege și aspectul legat de "achiziția" cunoștințelor (percepție sau învățare)

- exploatarea cunoștințelor pentru a lua decizii (secvențe) adecvate la momentul oportun.

Putem include în domeniul I.A. următoarele subdomenii:

- rezolvarea problemelor,
- achiziția și interpretarea sunetelor și imaginilor,
- demonstrația teoremelor,
- programare automată,
- generarea de planuri de activități,
- jocuri,
- interpretarea limbajului natural,
- sisteme expert.

Chiar dacă la o primă vedere această enumerare poate părea ciudată totuși subdomeniile de mai sus au ca puncte comune multiple aspecte legate de reprezentarea și exploatarea cunoștințelor relative la plicațiile abordate.

Capitolul 2: Metode de rezolvare a problemelor

Problemele complexe, de exemplu conducerea mișcării unui robot pe traiectorie fără obstacole sau conducerea adaptivă a unui bioreactor, nu pot fi rezolvate fără a face apel la tehnici de I.A. Vom prezenta în continuare o clasă particulară de astfel de tehnici.

2.1 Spațiul stărilor

Atunci când se dorește rezolvarea unei probleme complexe trebuie avute în vedere un număr de operații diferite:

- definirea **spațiului stărilor** pentru problema respectivă, adică un ansamblu care cuprinde toate stările (sau cazurile, sau configurările) posibile pentru domeniul de aplicabilitate investigat.
- specificarea uneia sau a mai multor stări de la care poate pleca procesul de rezolvare a problemei. Această stare va fi **starea inițială** a problemei.
- specificarea uneia sau mai multor stări care constituie soluția problemei puse. Acestea vor fi **stările finale** ale problemei.
- definirea unui **ansamblu de operatori** care să descrie acțiunile care pot fi efectuate în procesul de rezolvare în vederea atingerii uneia dintre stările finale plecând de la starea inițială.

Problema pusă este deci rezolvată aplicând operatorii și deplasându-se în spațiul stărilor pe un drum care pleacă dintr-o stare inițială și duce într-o stare finală. Acest proces de căutare (search) este fundamental în rezolvarea problemelor. Alegerea operatorilor se face în funcție de o **strategie de control**, despre care se va vorbi mai târziu.

Dacă în continuare vom utiliza termenii de operatori sau acțiuni sau reguli (de producție) atunci ne vom referi practic la același lucru.

2.2 Sisteme de producție

2.2.1 Definiții

Pentru că am amintit de căutare (search) care este un element cheie în majoritatea proceselor care utilizează I.A. este necesar să descriem într-o manieră unitară această noțiune. Un concept unificator în acest sens este conceptul de "sistem de producție" care, în sensul din I.A., constă în:

➤ o **structură de date** descriind starea curentă a problemei. Această structură poate fi una simplă: câteva numere reale, sau dacă problema o cere poate fi foarte complicată, de exemplu o bază de date relațională.

➤ un **ansamblu de reguli (de producție)**. Aceste reguli se aplică asupra structurii de date pentru a le modifica, a le face să evolueze. "Producția" constă în a "fabrica" noi structuri de date prin aplicarea regulilor disponibile. O regulă de producție este definită prin două elemente:

- o **precondiție** definită pe structura de date care trebuie verificată pentru ca regula să se poată aplica pe acea structură de date.
- o **descriere a modului de modificare** a structurii de date atunci când fiind aplicabil regula s-a aplicat. Pentru fiecare stare a structurii de date există deci un subansamblu de reguli, din ansamblul total al regulilor, care sunt aplicabile.

➤ o strategie de control care, la fiecare moment, poate conduce la alegerea regulii aplicabile care trebuie aplicată pe structura de date pentru a o modifica, în varianta fericită, de la starea inițială către starea finală.

Pentru exemplificare luăm un exemplu clasic în I.A.. Fie situația din figura 2.1.

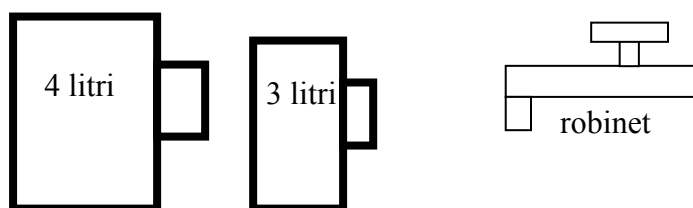


Figura 2.1 Exemplu de aplicație simplă care poate fi rezolvată prin tehnici de I.A.

Avem la dispoziție o sursă de apă (robinet), și două recipiente de 4 și respectiv 3 litri, inițial goale. Se dorește ca în final recipientul de 4 litri să conțină exact 2 litri.

Structura de date este simplă: este alcătuită din două numere reale care corespund respectiv conținutului instantaneu de apă din cei doi recipiente. Starea inițială este deci (0,0), în timp ce stările finale corespund la (2,x) unde $0 \leq x \leq 3$.

Regulile de producție aplicabile sunt:

- R1: umplerea unui recipient la fântână,
- R2: golirea unui recipient (lângă fântână, pe jos),
- R3: golirea unui recipient în celălalt.

Precondițiile sunt:

- pentru R1: recipientul să nu fie plin,
- pentru R2: recipientul să nu fie gol,
- pentru R3: primul recipient să nu fie gol (cel care se golește), iar al doilea (în care se golește) să nu fie plin.

Modul de modificare a bazei de date ca urmare a aplicării fiecărei reguli este ușor de definit:

- pentru R1: recipientul în cauză devine plin, celălalt recipient rămâne neschimbat,
- pentru R2: recipientul în cauză devine gol, celălalt recipient rămâne neschimbat,
- pentru R3: primul recipient își crește cantitatea cu cât se afla în celălalt recipient (dar nu mai mult decât plin – trebuie decis dacă vărsarea se oprește la plin sau se continuă până se golește), al doilea recipient își scade cantitatea cu cât se varsă (trebuie decis dacă vărsarea se oprește la plin sau se continuă până se golește).

Procedura care implementează sistemul de producție care să rezolve problema propusă este:

```
PROCEDURE sistem_de_producție
  baza_de_date_curentă := baza_de_date_inițială;
  WHILE baza_de_date_curentă /= stare_finală DO
    selectează o regulă aplicabilă R;
    baza_de_date_curentă := rezultatul_aplicării_regulii_R
  bază_de_date;
END;
END;
```

2.2.2 Strategiile de control

În procedura descrisă mai sus apare pur și simplu instrucțiunea "selectează o regulă aplicabilă R". Rolul strategiei de control este să aleagă efectiv regula aplicabilă și să păstreze în memorie secvența regulilor deja aplicate ca și structurile de date produse de aceste reguli. Apare evident că alegerea corectă, ca să nu spunem optimală, este punctul central al unei astfel de metode de rezolvare a problemelor.

Se pot distinge două tipuri de strategii:

➤ **Strategia irevocabilă** se întâlnește în cazul în care alegerea unei reguli aplicabile nu poate fi pusă în discuție ulterior. Acest tip de strategie dă rezultate foarte bune când dispunem de o **cunoaștere locală** suficientă în ceea ce privește maniera de evoluție spre scopul final indiferent de starea de plecare.

Printre metodele convenționale de optimizare (deci de minimizare a unei funcții de cost) se găsesc și un mare număr de algoritmi bazați pe calculul gradientului (un exemplu de cunoaștere locală), iar căutarea optimului se face în direcția inversă gradientului. Se poate aplica o astfel de metodă într-un sistem de producție cu condiția să dispunem de o funcție de cost (funcție reală definită pe structura de date). Strategia irevocabilă constă atunci în alegerea dintre regulile aplicabile a aceleia care produce o structură de date care corespunde la cea mai mare variație a funcției de cost (în sensul scăderii).

➤ **Strategiile revocabile** permit rediscutarea ulterioară a fiecărei alegeri a unei reguli. Este necesar atunci să se definească un mecanism de punere în discuție ulterioară a regulilor.

O astfel de strategie este și aceea care se numește și "**backtracking**" în sensul că de fiecare dată când se face o alegere a unei reguli aplicabile se definește și un punct de tip "backtracking". Dacă în cele ce urmează prin secvența de operații aleasă strategia de control nu ajunge să atingă scopul propus, atunci strategia se întoarce în ultimul punct "backtracking" stabilit. Se încearcă ca plecând de aici să se aleagă o altă regulă. Dacă această posibilitate nu există se face o nouă revenire cu încă un punct "backtracking" înapoi. Și așa mai departe.

O strategie revocabilă de tip "căutare în graf" (**graph search**) dacă se generează simultan mai multe secvențe de aplicare de reguli și la sfârșit se alege secvența optimă. Vom reveni asupra acestui subiect.

2.2.3 Tipuri de sisteme de producție

2.2.3.1 Sistemele direct și invers

Până acum am vorbit de evoluția unui sistem de producție de la starea inițială către starea finală. Un astfel de sistem se numește **direct**, iar regulile de producție de **tip F (forward)**. Ne putem imagina și o manieră de rezolvare inversă a problemei, plecând de la starea finală și o evoluție cu regului de tip B (**backward**) către starea inițială. Sistemul de producție este atunci de tip invers.

2.2.3.2 Sisteme decompozabile

În anumite condiții un sistem de producție poate fi **decompozabil**. Aceasta semnifică că structura de date inițială poate fi partiționată în mai multe componente care pot fi tratate în paralel. Partiționarea structurii de date și deci a scopului final de atins, poate în general să simplifice mult efortul implicat de strategia de control.

2.3 Grafuri

Vom examina aici o manieră de reprezentare a spațiului stărilor a unei probleme.

Un graf este o reprezentare care comportă noduri și arce orientate. Nodurile reprezintă diferitele structuri de date care pot fi generate prin sistemul de producție, iar arcele reprezintă regulile aplicabile diferitelor structuri de date. Un exemplu este dat în figura 2.2.

RI reguli aplicabile, SDJ structuri de date. SD1 este starea inițială, în această stare sunt aplicabile două reguli R1 și R2.

Dacă există un arc orientat de la nodul n_i către nodul n_j atunci n_i este **predecesorul** (ascendentul) lui n_j , iar n_j este **succesorul** (descententul) lui n_i .

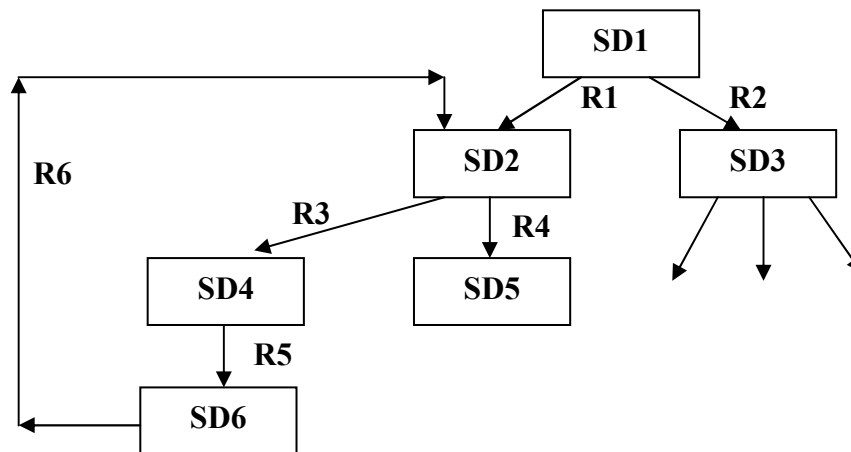


Figura 2.2 Reprezentarea unui sistem de producție printr-un graf

Un caz particular foarte important este acela în care fiecare nod nu are decât un singur ascendent (nodul inițial nu are nici unul). Avem de a face atunci cu un **arbore**. În figura 2.3 nu avem un arbore pentru că SD2 are doi ascendenți SD1 și SD6. Dacă SD6 este o stare finală acceptabilă pentru problemă atunci o soluție a problemei este dată de secvența R1, R3, R5.

De exemplu, fie exemplul cu cei doi recipiente de 4L și 3L atunci graful de stări al problemei este următorul: cu (0,0) starea inițială și cu (2,x) starea finală, unde "x" semnifică că este indiferentă cantitatea de apă din al doilea recipient.

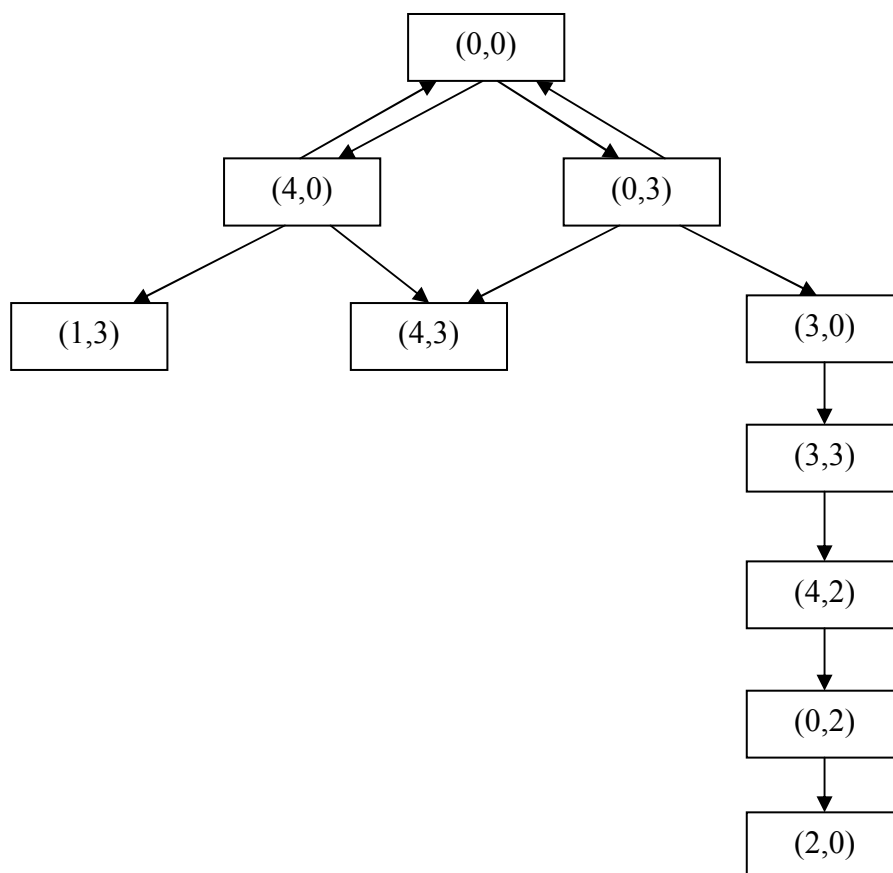


Figura 2.3 Reprezentarea sistemului de producție asociat aplicației luate ca exemplu