

STUDIU DE CAZ ASUPRA UNUI SISTEM EXPERT

Ionuț Chirilă

Galaji, Mazepa II, Str. Arhipelag nr.3, Bl.Gorun, Ap.65

Rezumat: Ce este un sistem expert? Un sistem de inteligență artificială care se încadrează în tipul sistemelor bazate pe cunoștințe. Un sistem trebuie să folosească cunoștințele pentru a genera toate soluțiile taskului pentru care a fost proiectat.

Ce este un sistem inteligent de diagnoză? Un caz particular de sistem expert.

Ce este o regulă? Regulile sunt folosite pentru a reprezenta cunoștințe euristice, asigurând realizarea unui set de acțiuni în condiții date, verificabile. Dezvoltarea unui sistem expert constă în definirea unui set de reguli care vor conduce la soluții. O regulă este alcătuită din antecedent (parte condițională, LHS-left hand side - partea stângă) și consecvent (parte de acțiuni, RHS-right hand și de - partea dreaptă). Antecedentul conține un set de condiții (elemente condiționale) care trebuie să fie satisfăcute pentru ca regula să fie aplicabilă. În CLIPS, partea condițională a unei reguli este satisfăcută dacă există sau nu anumite fapte în lista de fapte sau o anumită instanță a unei clase definite de utilizator sau este îndeplinită o condiție testată explicit. O condiție din antecedentul unei reguli se numește „pattern”. Procesul prin care se verifică faptele și instanțele care satisfac o anumită condiție (pattern) se numește „pattern-matching” (verificare). CLIPS asigură un mecanism, numit motor de inferență, care verifică automat condițiile regulilor în conformitate cu starea actuală a listei de fapte și a listei de instanțe, determinând regulile aplicabile la un moment dat, care sunt trecute într-o listă numită agendă. Consecventul unei reguli este un set de acțiuni care sunt executate la aplicarea regulii. Acțiunile sunt realizate atunci când motorul de inferență execută o regulă din cele aplicabile. Dacă sunt aplicabile mai multe reguli la un moment dat, motorul de inferență folosește o strategie de rezolvare a conflictelor care alege o singură regulă. Acest proces se termină atunci când nu mai există nici o regulă în agendă sau când s-a specificat explicit, printr-o comandă, oprirea inferenței (agenda conține toate regulile aplicabile la un moment dat). Se observă cu ochiul liber similitudinea cu o structură IF-THEN dintr-un limbaj de programare.

Dar un fapt? Forma de bază de reprezentare în CLIPS pentru cunoștințe. Faptele sunt organizate în liste de fapte. Un fapt poate fi specificat prin index sau prin adresă. Când un fapt este adăugat la lista de fapte sau modificat, acesta primește un index unic în lista de fapte. Indexul de fapte începe de la zero și se incrementează cu unu la fiecare nou fapt adăugat sau modificat. Când se apelează comenzile reset sau clear indexul de fapt este pus la zero.

Operațiile ce se pot efectua asupra faptelor sunt:

1. adăugarea de noi fapte la lista de fapte;
2. ștergerea de fapte din lista de fapte;
3. modificarea unui fapt;
4. duplicarea unui fapt.

Faptele sunt de trei categorii:

- ordonate, constau într-un simbol urmat de o secvență de zero sau mai multe câmpuri separate de spații și delimitate de paranteze “()”;
- neordonate, constau dintr-o structură abstractă (similară cu structura din C) în care fiecare câmp are un nume;
- fapte inițiale cu ajutorul constructorului deffacts se pot introduce în lista de fapte o colecție de fapte folosind comanda reset.

Cuvinte cheie: sistem, expert, fapt, regulă, inferență, bază, cunoștințe.

Abstract: What is an expert? An artificial intelligence system that fits the type of systems based on knowledge. A system must use knowledge to generate all solutions for which task it was designed. What is an intelligent diagnosis system? A particular case of the expert system.

What is right? The rules are used to represent knowledge Heuristic, providing a set of actions on data, verifiable. Developing an expert system consists of defining a set of rules that will lead to solutions. A rule consists of history (part conditional, LHS left-hand side - the left) and consistent (the actions RHS right-hand side - the right). History contains a set of conditions (conditional elements) that must be satisfied for the rule is applicable.

But a fact? The basic form of representation for the Clips knowledge. The facts are organized into lists of facts. A fact can be specified by index or address. When one is added to the list of facts or receives as a unique index in the list of facts. Index of facts starting from scratch and incrementează with every new one is added or changed. When calling reset or clear orders index is set to zero. Operations that can make the facts are: * adding new facts to the list of facts; * removal of facts from the facts, that an amendment * * duplication of fact.

Facts are three categories: a) ordered, consisting of a symbol followed by a sequence of zero or more fields separated by spaces and bounded by parentheses “()”. B) neordonate consist of an abstract structure (similar to structure of C) in which each field has a name. c) initial facts using manufacturer deffacts may be placed in the collection of facts for facts using the command reset.

Keywords: system, expert, fact, rule, inference basic knowledge.

1. Analiza aplicației

Baza de fapte care vor fi supuse evaluării și clasificării prin metoda interviului arată ca mai jos. Observăm că fiecare fișă a unui bolnav conține cele trei atribute cerere după care se va face căutarea, precum și unele atribute ne semnificative, care nu au nici un cuvânt de spus în detectarea soluției.

Folosind ghidul de recunoaștere al raționamentului aplicat, explicat în subcapitolele de mai sus, observăm ușor că:

- sistemul facilitează clasificarea, selecția și apoi diagnoza;
- faptele din bază sunt destul de transparente încă de la început;
- procentajul de reguli aplicabil într-un caz este relativ scăzut;
- cererea este bine cunoscută de la început;
- soluția este detectată prin inducție;
- modelul folosit este cel al interviului cu utilizatorul pentru captare;

Observații. Regulile de control și, implicit, soluțiile nu acoperă 100% baza de fapte (există teste pentru care sistemul nu găsește nici o soluție). Aceasta din cauză că am încercat păstrarea legăturilor cu lumea reală, iar evaluarea prin clasificare/diagnoză nu presupune că pentru orice șablon există soluții. Aceste legături cu lumea reală se vor observa din testele selectate, unde:

- în cele mai multe cazuri, progresul bolnavului este LENT;
- simptomele sunt, în general, hal (halucinații) sau voci, specifice unor oameni care suferă de tulburări psihice;
- în cele mai multe cazuri, datorită naturii afecțiunilor, ei au mai fost internați;
- locul anterior internării este, în general, spitalul;
- analizele efectuate pe bolnavi sunt, în general, proaste sau medii.

(deffacts BOLNAVI::lista-bolnavi

(bolnav (name Ionescu_R) (adresa str. Domnească nr. 10) (telefon 0236412345) (diagnostic schizofrenie)(medicament Prozac) (loc sanatoriu))

(bolnav (name Popescu_P) (adresa bd. Timișoara 14) (telefon 0215412345) (diagnostic schizofrenie) (medicament Prozac) (loc spital))

(bolnav (name Voinescu_I) (adresa str. Dreptății 24) (telefon 0745412345) (diagnostic schizoid)(medicament DURO) (loc sanatoriu))

Pentru a veni în ajutorul utilizatorului, am realizat ask-question în așa fel încât, dacă el introduce un alt răspuns (fie litere mari sau mici) decât cele din lista răspunsuri permise, întrebarea să se repete, el fiind astfel atenționat. Se observă mai jos.

(deffunction MAIN::ask-question (?question ?allowed-values)

(printout t ?question)

(bind ?answer (read))

(if (lexemep ?answer) then (bind ?answer (lowcase ?answer)))

(while (not (member ?answer ?allowed-values)) do

(printout t ?question)

(bind ?answer (read))

(if (lexemep ?answer) then (bind ?answer (lowcase ?answer))))

?answer)

Pentru stadiul de plecare, am implementat regula „start” cu grad de încredere 10,000, dând posibilitatea ca același fapt să fie adăugat cu un nou indice prin set-fact-duplication TRUE. Pe stiva „focus”, am adăugat modulele: ÎNTREBARI, ALEGE-BOLNAVI și BOLNAVI (cu faptele de plecare care trebuie sortate).

De asemenea, aici am implementat și regula de combinare a certitudinilor prin care, pentru aceeași concluzie a două reguli diferite, probabilitatea finală este obținută prin metoda „suma probabilității”:

```
(defrule MAIN::combine-certainties ""  
(declare (salience 100)  
  (auto-focus TRUE))  
?rem1 <- (attribute (name ?rel) (value ?val) (certainty ?per1))  
?rem2 <- (attribute (name ?rel) (value ?val) (certainty ?per2))  
(test (neq ?rem1 ?rem2))  
=>  
(retract ?rem1)  
(modify ?rem2 (certainty (/ (- (* 100 (+ ?per1 ?per2)) (* ?per1 ?per2)) 100))))
```

Observații: Această metodă intervine în următorul test:

- lent;
- da;
- spital;
- voci;
- da;
- proaste;
- oscilant.

Pentru modulul de întrebări, am folosit regula ask-a-question, la care am adăugat și posibilitatea ca o întrebare să aibă precursori - în construcția arborelui de căutare.

```
(deftemplate INTREBARI::question  
(slot attribute (default ?NONE))  
(slot the-question (default ?NONE))  
(multislot valid-answers (default ?NONE))  
(slot already-asked (default FALSE))  
(multislot precursors (default ?DERIVE)))  
  
(defrule INTREBARI::ask-a-question  
?f <- (question (already-asked FALSE)  
  (precursors)  
  (the-question ?the-question)  
  (attribute ?the-attribute)  
  (valid-answers $?valid-answers))  
=>  
(modify ?f (already-asked TRUE))  
(assert (attribute (name ?the-attribute)  
  (value (ask-question ?the-question ?valid-answers))))
```

Pentru precursori, se testează dacă avem precursor satisfăcut sau nu.

Pentru modulul INTREBĂRI-BOLNAV, am folosit metoda interviului ca mai jos. Întrebarea conține în paranteză și numele variantelor acceptate, pentru a veni în ajutorul utilizatorului.

(defmodule BOLNAV-INTREBĂRI (import ÎNTREBARI ?ALL))

(deffacts BOLNAV-ÎNTREBĂRI::question-attributes

(question (attribute comportament)

*(the-question „Ce fel de comportament are subiectul căutat?
(liniștit/violent/oscilant/imprevizibil)“)*

(valid-answers liniștit violent oscilant imprevizibil))

(question (attribute e-făcut) (the-question “Are analizele făcute? (da/nu) “) (valid-answers da nu))

(question (attribute analize)

(precursors e-facut is da)

(the-question “ Cum i-au ieșit analizele? (bune/medii/nesatisf/proaste) “)

(valid-answers bune medii nesatisf proaste))

(question (attribute simptome)

(the-question “ Ce simptome manifestă subiectul căutat? (hal/voci/dureri/nimic/altele) “)

(valid-answers hal voci dureri nimic altele))

(question (attribute rec)

(the-question “ A mai fost internat? (da/nu) “)

(valid-answers da nu))

(question (attribute prev-loc)

(precursors rec is da)

(the-question “ În ce fel de loc? (spital/sanatoriu/acasă/altul) “)

(valid-answers spital sanatoriu acasă altul))

(question (attribute progres)

(the-question “ Cum progresează subiectul căutat? (lent/satisf/mediu/rapid) “)

(valid-answers lent satisf mediu rapid))

Pentru modulul REGULI, am considerat structura multislot if-then cu încredere default 100. De asemeni, am implementat astfel încât să pot avea clauze conectate cu AND atât în antecedentul regulii, cât și în consecvent.

Exemplu: *(rule (if comportament is violent and prev-loc is spital)*

(then best-loc is sanatoriu with certainty 75))

(rule (if comportament is oscilant and progres is lent)

(then best-loc is spital with certainty 50 and

best-loc is sanatoriu with certainty 40))

Pentru regulile obținute cu o anumită certitudine, combinăm prin minimizare certitudinea concluziei cu cea a regulii.

(defrule RULES::perform-rule-consequent-with-certainty

?f <- (rule (certainty ?c1)

(if

(then ?attribute is ?value with certainty ?c2 \$?rest))

=>

```

(modify ?f (then ?rest))
(assert (attribute (name ?attribute)
                 (value ?value)
                 (certainty (/ (* ?c1 ?c2) 100))))

(defrule RULES::perform-rule-consequent-without-certainty
?f <- (rule (certainty ?c1)
           (if
            (then ?attribute is ?value $?rest))
           (test (or (eq (length$ ?rest) 0)
                    (neq (nth 1 ?rest) with))))
=>
(modify ?f (then ?rest))
(assert (attribute (name ?attribute) (value ?value) (certainty ?c1)))

```

În modulul ALEGE-BOLNAVI, am inclus regulile necesare construirii soluției, adică cele pentru depistarea celor trei atribute după care am efectuat evaluarea faptelor bolnavi: best-medicament, best-diagnostic, best-loc. Așa cum am mai menționat la subcapitolul factorilor de încredere, pentru a fi sigur că soluția este riguroasă, am optat pentru metoda minimizării de a obține certitudinea totală a faptelor soluții.

```

(defmodule ALEGE-BOLNAVI (import RULES ?ALL)
  (import INTREBARI ?ALL)
  (import MAIN ?ALL))
(defrule ALEGE-BOLNAVI::startit => (focus RULES))
(deffacts reguli-bolnavi

```

Iată câteva exemple de reguli de comandă din acest modul:

; Pentru alegerea bolnavilor cu cel mai apropiat loc

```

(rule (if comportament is violent and prev-loc is spital)
      (then best-loc is sanatoriu with certainty 75))

```

```

(rule (if comportament is oscilant and prev-loc is acasă)
      (then best-loc is spital with certainty 50))

```

```

(rule (if comportament is linistit and prev-loc is acasă)
      (then best-loc is acasă with certainty 50))

```

Pentru alegerea bolnavi cu cel mai apropiat medicament

```

(rule (if simptome is hal and progres is lent)
      (then best-medicament is Prozac with certainty 85 and
           best-medicament is DURO with certainty 60))

```

```

(rule (if analize is nesatisf and progres is satisf)
      (then best-medicament is PEBA with certainty 55))

```

```

(rule (if simptome is voci and comportament is oscilant)
      (then best-medicament is DURO with certainty 75 and))

```

; Pt alegerea bolnavi cu cel mai apropiat diagnostic

(rule (if simptome is hal and comportament is oscilant)

(then best-diagnostic is psihotism with certainty 50 and

best-diagnostic is paranoia with certainty 40))

(rule (if comportament is oscilant and simptome is nimic)

(then best-diagnostic is depresie with certainty 55))

Iată cum arată două teste în mediul CLIPS care are ca date de intrare captate prin interviu următoarele răspunsuri:

- lent, nu, hal, nu, oscilant.

Pentru selectarea celui-mai-bun loc de tratare motorul inferențial selectează premisele regulii:

(rule (if comportament is oscilant and progres is lent)

(then best-loc is spital with certainty 50 and

best-loc is sanatoriu with certainty 40))

Pentru selectarea celui-mai-bun medicament, după parcurgerea regulii și suspendarea temporară a celor care nu pot fi evaluate sau eliminarea celor care nu ajută în construirea soluției, rămâne validă:

(rule (if simptome is hal and progres is lent)

(then best-medicament is Prozac with certainty 85 and

best-medicament is DURO with certainty 60))

Pentru selecția celui mai bun diagnostic, avem:

(rule (if simptome is hal and progres is lent)

(then best-diagnostic is schizofrenie with certainty 80))

De aici, prin combinare și folosind minimizarea certitudinilor, se obține un rezultat cu o probabilitate minimă de 50 pentru spital și 40 pentru sanatoriu, care se compară cu înregistrările de fapte din baza inițială. De aici, rezultă pentru bolnavii Sorescu_M și Popescu_P (care au cele trei atribute identice) certitudinea 50%, iar pentru Ionescu_R, care are best-loc sanatoriu, o probabilitate mai scăzută, de 40%.

Procedând analog, prin inducție, interpretăm rezultatul în cazul următorului test: lent, da, spital, maniacale, nu, imprevizibil.

Motorul de inferență va avea de potrivit următorul set de date cu baza fapte:

Best-loc spital 65%

Best-medicament- DURO 40%

Best-diagnostic maniac 50%

Aici, prin comparare cu baza fapte, observăm cu ochiul liber că singurul care corespunde este Stamate_D, cu o probabilitate medie a minimumului de 40%.

În modulul de AFIȘARE, am alocat un grad de încredere scăzut funcției de afișare a rezultatelor (saliency 10), pentru a fi sigur că se va executa ultima.

Deși aplicația a urmărit, în principal, implementarea unui sistem expert bazat pe reguli, se desprinde o serie mai largă de concluzii și observații atât la nivel conceptual, cât și la nivelul structurilor implementate.

- toate componentele sistemului expert constituie subsisteme tratate separat după aceleași principii de proiecție, dar care trebuie să lucreze împreună ca un tot unitar.
- captarea cunoștințelor din domeniul de expertiză = un subset al unui set de cunoștințe (la nivel conceptual asemănător încapsulării din Java);
- totalitatea cunoștințelor expertului uman ® formulare expert uman ® captura inginer cunoștințe ® cunoaștere tehnică, corespunzătoare ® domeniul de definiție sistem expert;
- un sistem expert este capabil să proceseze simultan o cantitate mare de informații față de un expert uman; poate urma timp îndelungat un raționament complicat cu succes; de asemenea, nu pierde informația relevantă;

- e. raționamentul „forward” diferă de raționamentul „backward” prin trei caracteristici:
- „forward” pornește de la premisele regulilor la concluzii; „backward” începe de la cerere ® concluzii reguli;
 - „forward” preia regulile în ordinea naturală în care apar în baza de cunoștințe; „backward” ia în considerare numai regulile a căror concluzie coincide cu cererea;
 - „forward” nu suspendă niciuna din regulile pe care nu o poate evalua.
- f. folosind redefinirea regulilor un sistem de înlănțuire „forward” poate fi implementat printr-un sistem de înlănțuire “backward” și invers; în schimb, are de suferit eficiența;
- g. există asemănări între investigarea înapoi și metoda căutării cu revenire („backtraking”) din limbajele procedurale tradiționale;
- h. căutarea în arborele „backward” este determinată de consecvenții regulilor, în timp ce căutarea în arborele „forward” este determinată de antecedentii regulilor;
- i. un sistem expert bazat pe reguli prezintă avantaje precum: modul cum lucrează motorul de inferență este mult mai ușor de observat, este ușor de modificat, iar deciziile sunt mai ușor de monitorizat;
- j. implementarea sistemului expert folosind module (pentru aplicația în discuție modul întrebări, modul reguli, modul bolnavi etc.) înseamnă: urmărirea execuției mai ușor, posibilitatea corectării mai rapide a unei erori de implementare, modificări mai facile de structură;
- k. acum, să ne închipuim o a doua abordare a problemei; astfel, avem în cerere tot cele trei atribute, dar baza de fapte cu bolnavi conține mai multe, în care toate sunt relevante în stabilirea soluției.

De exemplu, așa ar arăta un fragment al bazei:

(defacts BOLNAVI::lista-bolnavi

(bolnav (name Ionescu_R) (diagnostic schizofrenie)(medicament Prozac) (loc sanatoriu)(simptome hal)(analize proaste)(comportament violent))

(bolnav (name Popescu_P) (diagnostic schizofrenie) (medicament Prozac) (loc spital)(simptome voci)(analize proaste)(comportament oscilant))

(bolnav (name Voinescu_I) (diagnostic schizoid)(medicament DURO) (loc sanatoriu)(simptome voci)(analize medii)(comportament oscilant))

În acest caz, și interviul trebuie puțin modificat. Mai jos un exemplu posibil:

(question (attribute comportament)

(the-question “Ce fel de comportament are subiectul cautat? “)

(valid-answers linistit violent oscilant imprevizibil))

(question (attribute e-facut) (the-question “Are analizele facute? “) (valid-answers da nu))

(question (attribute analize)

(precursors e-facut is da)

(the-question “ Cum i-au iesit analizele?”)

(valid-answers bune medii nesatisf proaste))

(question (attribute simptome)

(the-question “ Ce simptome manifesta subiectul cautat?”)

(valid-answers hal voci dureri nimic))

(question (attribute rec)

(the-question “ A mai fost internat? “)

(valid-answers da nu))

(question (attribute lua-medicament)

(the-question “ Ce medicament lua subiectul cautat: DURO, ANTI, Prozac, PEBA, altul? “)

(valid-answers DURO ANTI Prozac PEBA altul)

(question (attribute avea-diagnostic)

(the-question “ Ce diagnostic avea subiectul cautat: schizofrenie, paranoia, schizoid, psihotism, altul? “)

(valid-answers schizofrenie schizoid paranoia psihotism altul)

(question (attribute prev-loc)

(precursors rec is da)

(the-question “ In ce fel de loc? “)

(valid-answers spital sanatoriu acasa alt)

(question (attribute progres)

(the-question “ Cum progreseaza subiectul cautat? “)

(valid-answers lent satisf mediu rapid))

Regulile în bună măsură rămân aceleași, cu modificările necesare. Apar atributele comportament, analize și simptome și precursorul lua-medicament.

Pentru aceleași teste, rezultatele pierd semnificativ din certitudine, datorită tocmai faptului că, deși acum un fapt bolnav este descris complet de șase factori, căutarea se realizează după numai trei. Astfel, deși certitudinea finală poate fi aceeași cu cea din cazul corespunzător al primei abordări, celelalte trei atribute putând infirma anumite concluzii și chiar reguli.

2. Concluzie

Un sistem expert poate reprezenta o componentă dinamică a unei aplicații care să culeagă informații prin intermediul metodei interviului. Față de algoritmi din programarea tradițională care lucrează cu date omogene, baza de cunoștințe a sistemului acceptă structuri eterogene ale informației, modelul real având de câștigat.

Bibliografie

1. **NOVAC UDUDEC, C.:** Inginerie Software, Editura Tehnică, București, 1999.
2. **Georgescu, I.:** Elemente de inteligență artificială, Ed. Academiei, București, 1985.
3. **NOVAC, C.:** Inteligență artificială și sisteme expert, Note de curs.
4. **TATAR, D.:** Inteligența artificială: demonstrarea automată a sistemelor, prelucrarea limbajului natural, Editura Albastră, Cluj-Napoca, 2001.