Paradigme de programare

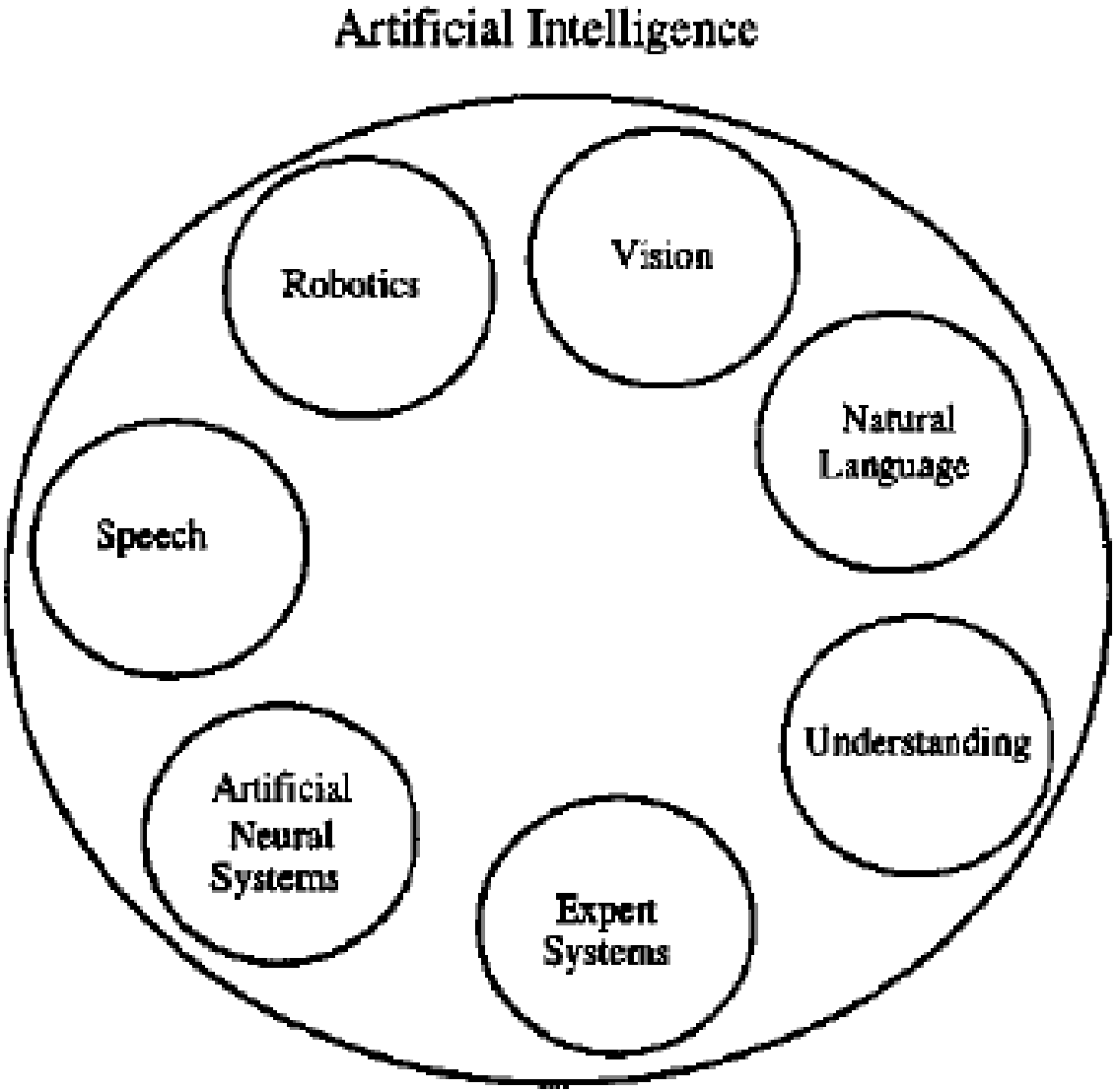
2010-2011, semestrul 2

Curs 9

Cuprins – Programare asociativa in CLIPS

- Sisteme expert
- CLIPS

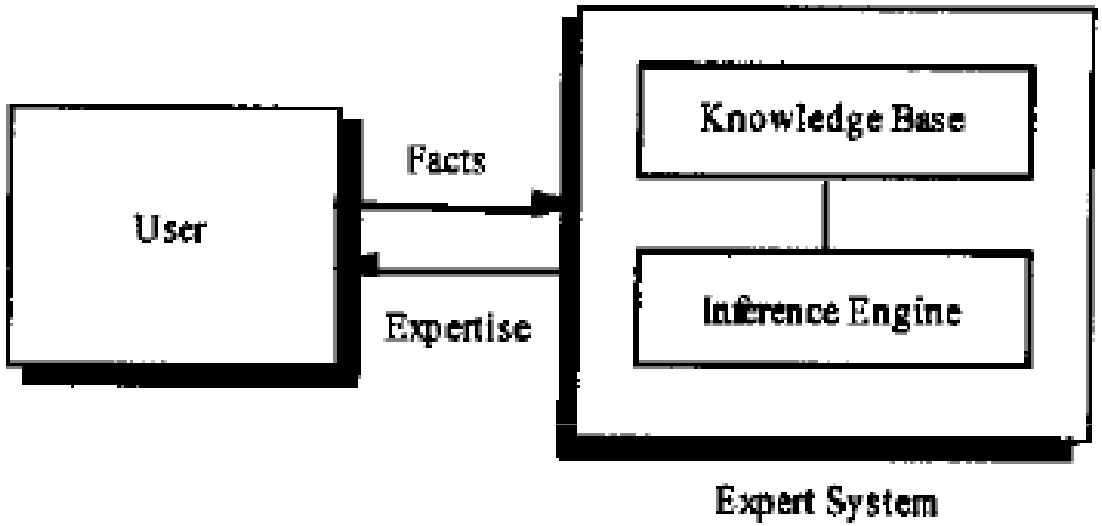
Context


Artificial Intelligence
Robotics
Vision
Natural Language
Speech
Understanding
Artificial Neural Systems
Expert Systems

Sistem expert - definitie

- Edward Feigenbaum: “un program inteligent care foloseste cunostinte si reguli de inferenta pentru a rezolva probleme suficient de dificile incat sa necesite expertiza umana semnifcativa”
- Un sistem expert simuleaza procesul de decizie al unui expert uman
- Restrictionat la un anumit domeniu (dezvoltarea unui rezolvitor generic de probleme inca ne depaseste)

Interactiunea utilizator - SE



- Disponibil pe scara larga, in permanenta, ofera raspunsuri rapide
- Reuneste cunostintele a multipli experti
- Poate explica in detaliu rationamentul care a dus la o concluzie
- Constant si neemotional

Aplicatii ale SE

- Configurare de sisteme
- Diagnostic
- Educatie
- Monitorizare
- Planificare
- Prognoza
- Remediere
- etc

Reprezentarea cunostintelor intr-un SE

- Un inginer de cunostinte intervieveaza experti umani si transfera informatia in SE
- Metoda comuna: reguli daca-atunci
- Adesea aceste reguli prezinta un grad de incredere
- Se reprezinta cu precadere euristici, nu inlantuiri cauzale complexe
- Cunostintele nu se pot generaliza prin analogie cu alte domenii (SE nu cunoaste nimic despre alte domenii)

Cand este adecvat un SE

- Probleme care nu au o solutie algoritmica
- Cunostintele expertului tind sa fie nesigure si euristice
- Vom folosi inferente pentru a atinge o solutie “rezonabila”
- Daca sistemul expert necesita o structura de control foarte rigida, probabil poate fi recodificat ca algoritm

CLIPS

- Limbaj multiparadigma
  - Programare bazata pe reguli
  - Programare procedurala
  - Programare orientata obiect
- Dezvoltat de NASA in anii ’80
- Folosit pentru dezvoltarea de sisteme expert

Determinarea drumului minim intre 2 noduri din graf – o solutie declarativa

G = (V,E) – un graf orientat

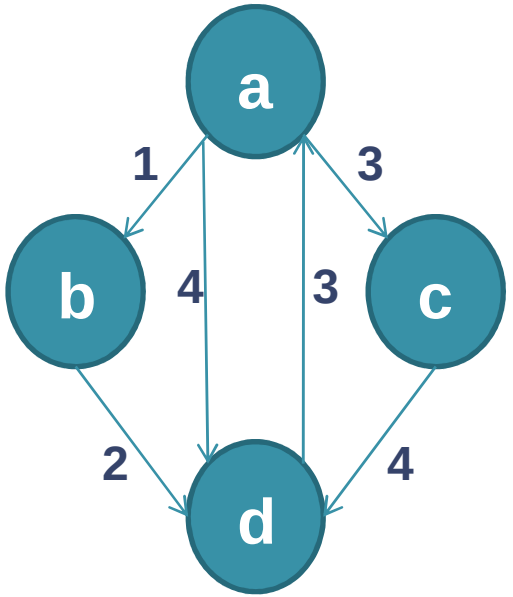
w:E->R_+ - o functie care asociaza unei muchii un cost

1. Cale initiala = cale de cost 0, care contine numai nodul de start
2. Calea x..y,z extinde calea x..y daca (y,z) **<-** E si z <-/- x..y. Costul caii x..y,z = costul caii x..y + w(y,z)
3. Calea x..y este utila daca nu se cunoaste la momentul respectiv o alta cale de cost mai mic de la x la y
4. Cale minima = cale utila in momentul in care nicio cale utila nu mai poate fi extinsa

Reprezentarea datelor in CLIPS

- Fapte (descriu atributele semnificative ale obiectelor din problema)

```
(deffacts edges
   (edge (from a) (to b) (cost 1))
   (edge (from a) (to c) (cost 3))
   (edge (from a) (to d) (cost 4))
   (edge (from b) (to d) (cost 2))
   (edge (from c) (to d) (cost 4))
   (edge (from d) (to a) (cost 3)))
```



Template-uri

- Faptele sunt instante ale unor template-uri
- Pentru fapte gen (edge (from a) (to b) (cost 1)):

(deftemplate edge (slot from) (slot to) (slot cost))

- Nu este obligatoriu sa specificam tipul variabilelor din sloturi. In cazul in care nu specificam, tiparea este dinamica.

Mai multe template-uri

```
(deftemplate required-path (slot start) (slot stop))
(deftemplate path (multislot nodes) (slot cost))
```

- multislot – poate contine un numar variabil de valori, nu neaparat de acelasi tip
- Daca un template se reduce la un singur multislot, el nu mai trebuie declarat explicit (ex: (om “gigi” 16 “oradea”))

Reguli

(defrule nume-regula
  $pattern_1$
  ...
  $pattern_n$
=>
  $actiune_1$
  ...
  $actiune_m$)

- Patternurile sunt fapte parametrizate, eventual incomplete

Reguli - exemple

```
(defrule initial-path
   (required-path (start ?x))
=>
   (assert (path (nodes ?x) (cost 0))))
```

- Patternurile controleaza aplicabilitatea regulii (o regula este aplicabila atunci cand se pot gasi in baza de fapte fapte care sa se potriveasca cu fiecare pattern al regulii)
- Pot exista reguli fara niciun pattern

Principiul refractiei

O regula se aplica o singura data pe un acelasi set de fapte cu un acelasi set de legari ale variabilelor din patternuri la valori.

Regiunea variabilelor din reguli

- Variabilele folosite intr-un $pattern_i$ al unei reguli R sunt vizibile in orice $pattern_j$, j>i, din R

Ex:

(f1 ?x)

(f2 ?x) ;; acelasi ?x

- Regiunea variabilelor definite in R nu depaseste pe R

Mai multe reguli

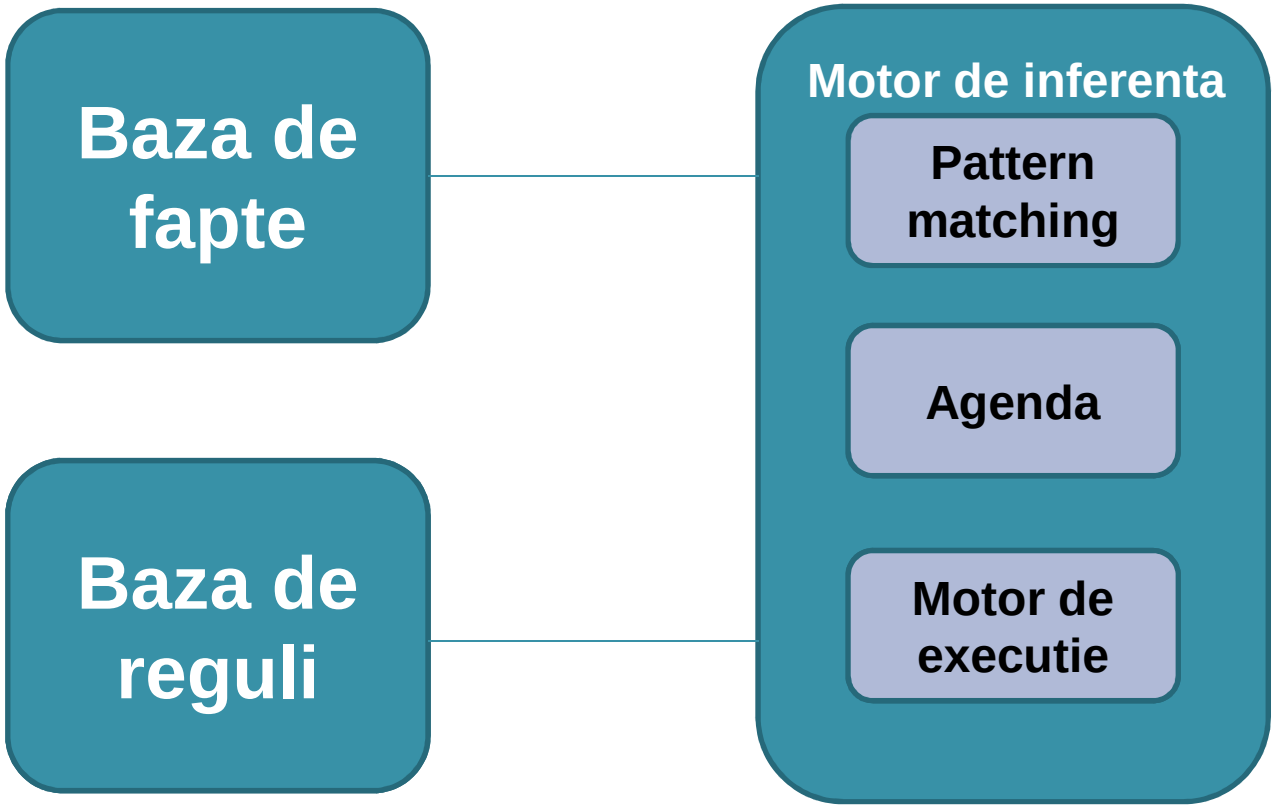
```
(defrule extend-path
   (path (nodes $?n ?x) (cost ?c))
   (edge (from ?x) (to ?z&~?x&:(not (member ?z $?n)))
   (cost ?c1))
=>
   (assert (path (nodes $?n ?x ?z) (cost (+ ?c ?c1)))))

(defrule discard-useless-path
   (declare (salience 1))
   (path (nodes $?n ?x) (cost ?c))
   ?f <- (path (nodes $?n1 ?x) (cost ?c1))
   (test (> ?c1 ?c))
=>
   (retract ?f))
```

Mai multe reguli

```
(defrule min-path
   (declare (salience -1))
   (required-path (stop ?y))
   (path (nodes $?n ?y) (cost ?c))
=>
   (printout t “min path: ” (create$ $?n ?y) “ with cost ”
   ?c crlf))
```

Schema a executiei unui program CLIPS



Agenda CLIPS

- CLIPS: controlul fluxului este prin date
- o regula este facuta aplicabila de
  - Diverse combinatii de fapte
  - Diverse legari ale variabilelor in cadrul acelorasi patternuri
  - Cand, ca urmare a pattern match-ului, toate patternurile unei reguli au putut fi potrivite cu un set de fapte, pentru aceasta regula si aceasta potrivire se creeaza o inregistrare de activare
- Agenda = structura de date care inmagazineaza toate inregistrarile de activare ale regulilor aplicabile la un moment dat

Inregistrari de activare

- Exemplu la calculator

Algoritm de functionare a agendei CLIPS

Agenda_alg (Reguli,Fapte) {

Agenda <- {inregistrari activare rezultate din pattern matching}

cat timp Agenda != vida {

sorteaza Agenda descrescator dupa salience

inregistrare = cap(Agenda)

Agenda = Agenda - {inregistrare}

executa regula corespunzatoare inregistrarii

daca regula=terminala intrerupe

Agenda <- {inregistrari activare rezultate din pattern matching}

}

Modalitati de terminare a executiei programului

- (run <limita>)
- La intalnirea unei reguli terminale
- Nu mai exista nicio regula aplicabila (Agenda este vida)

Recursivitate in CLIPS

- Exemplu la calculator