# Medii de programare în Inteligența Artificială

## Sisteme Expert

**Lect. univ. dr. Mihaela Colhon**

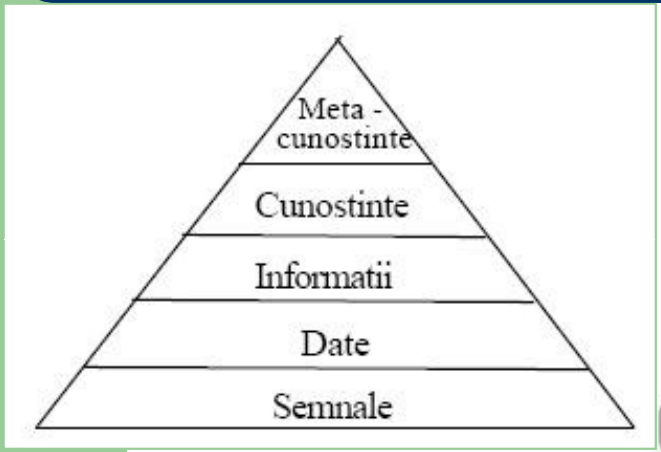
# Sisteme bazate pe cunoștințe

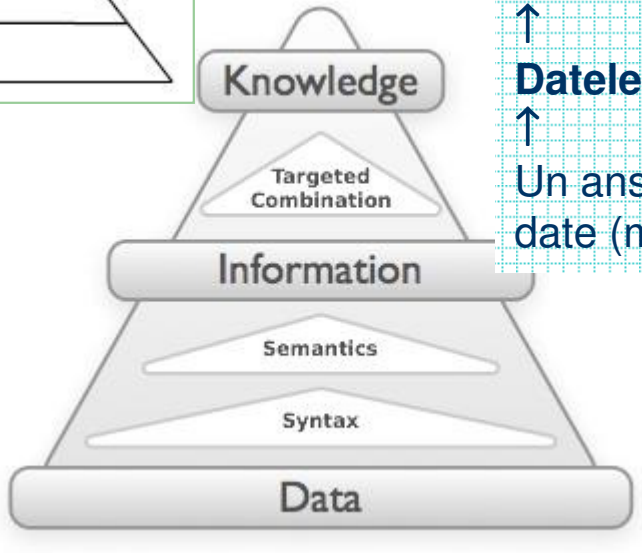
Un sistem de inteligență artificială, sau sistem bazat pe cunoștințe, este format din două componente fundamentale:

- **baza de cunoștințe** care conține cunoștințele specifice domeniului problemei și eventual, cunoștințe generale
- **motorul de inferență** care procesează cunoștințele pentru rezolvarea problemei.

**Definiție** *În inteligența artificială cunoștințele sunt văzute ca mulțimea de fapte și principii acumulate de oameni sau, mai general, nivelul cunoașterii umane la un anumit moment*.

# Clasificarea cunoasterii dupa gradul de generalitate




Cunostintele relative la alte cunostinte se numesc **meta-cunostinte**.

↑

**Informatiile** pot fi combinate in entitati de cunoastere mai complexe, numite **cunostinte**.

↑

**Datele** sunt materia prima a informatiilor.

↑

Un ansamblu de **semnale** alcatuiesc setul de date (mesaje).

# Date. Informații. Cunoștințe

- **Data**: orice simbol sau semnal care fără a fi încadrat într-un anume context nu are niciun înțeles.

  Exemple: yes, galben, 112221, etc.
- Datele pentru a avea sens sunt organizate sau prelucrate în structuri mai complexe numite **informații**.

  Exemple: numere de telefon - 0351111222, 0251223456, etc.;

  culori – roșu, galben, albastru, etc.
- **Cunoștințe**: sunt substituite unori reguli necesare înțelegerii și interpretării informațiilor.

  Exemple: dacă *temperatura e mai mare decât 20°* atunci se considera ca *vremea e calduroasă*; daca *nota este mai mare de 5* atunci se consideră ca e *notă de trecere*.

# Tipuri de cunoștințe

În cadrul acestor sisteme sunt codificate mai multe tipuri de cunoștințe:

- **cunostinte procedurale** (*cum* sa rezolvam o problema): reguli, strategii, agende, proceduri si functii;
- **cunostinte declarative** (*ce* este cunoscut despre problema): concepte, obiecte, fapte;
- **cunostinte bazate pe mostenire** (descriu organizarea cunostintelor): pot fi de doua tipuri:
  - **ne-structurate**: retele semantice
  - **structurate**: cadre, reprezentari orientate obiect
- **meta-cunostinte** (*cunostinte despre cunostinte*): informatia de care este nevoie pentru rezolvarea unei probleme;
- **euristicile** (*reguli-de-aur* care ghideaza rationamentul): criterii, metode sau principii pentru a decide care dintre mai multe posibile alternative de derulare a actiunii promite cea mai eficace cale de rezolvare a problemei

# Inferența în programe IA

**Definiție**. *Se numește metodă de inferență, sau pe scurt inferență, procedura de obținere la un moment dat, a noi elemente (fapte) pe baza cunoștințelor sistemului*.

Fiecare model de reprezentare a cunoștințelor are metode de inferență specifice. Pentru a putea ajunge la solutia unei probleme este necesar, de cele mai multe ori, o aplicare repetata a metodei de inferență.

**Definitie**. *Se numește strategie de control procesul de aplicare repetată a metodei de inferență pentru a ajunge la soluție, de preferinta cât mai repede*.

Metoda de inferență împreună cu strategia de control formează nucleul motorului de inferență al unui sistem bazat pe cunostinte.

# Sisteme Expert. Definire

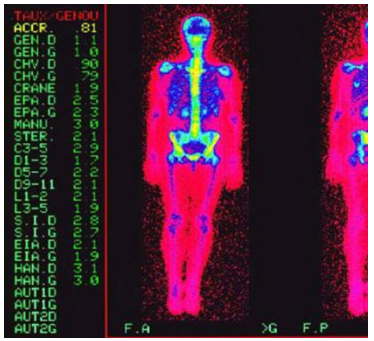
*Un sistem expert este un produs software care poate raționa ca un expert uman în domeniu. Deci, un sistem expert trebuie sa modeleze atât cunoștințele expertului cât și modul în care acesta raționează.*

N. Țăndăreanu:- SISTEME EXPERT. Reprezentarea cunoștințelor și inferență (2001)

*Un sistem expert este un sistem care **emulează** abilitatea de a lua decizii a expertului uman. Termenul «emulează» înseamnă că sistemul este menit să acţioneze în toate privinţele ca expertul uman. Emularea este ceva mai mult decât **simularea**, care cere doar să se acţioneze prin imitarea condiţiilor realizate. Sistemele expert acţionează foarte bine în **domenii bine delimitate**.*

J. Giarratano, G. Riley:- Expert Systems. Principles & programming, 2nd edition (1993)

# Sisteme Expert




Deoarece regulile de raționament sunt ușor de implementat în programe de calculator, sistemele expert pot fi ușor dezvoltate folosind computere.

Domeniul in care astfel de programe sunt folosite cu mare succes este **medicina** (cu precadere sunt utlizate în diagnoză).

Avantajele unui sistem expert versus un doctor:

- SE pot procesa o baza de cunoștințe/date mult mai mare
- SE nu pot uita sau greși în aplicarea regulilor de raționament
- SE este disponibil (aproape) permanent și nu iese la pensie
- Un SE poate accesa și prelucra cunoștințe mai complexe decât un doctor obișnuit

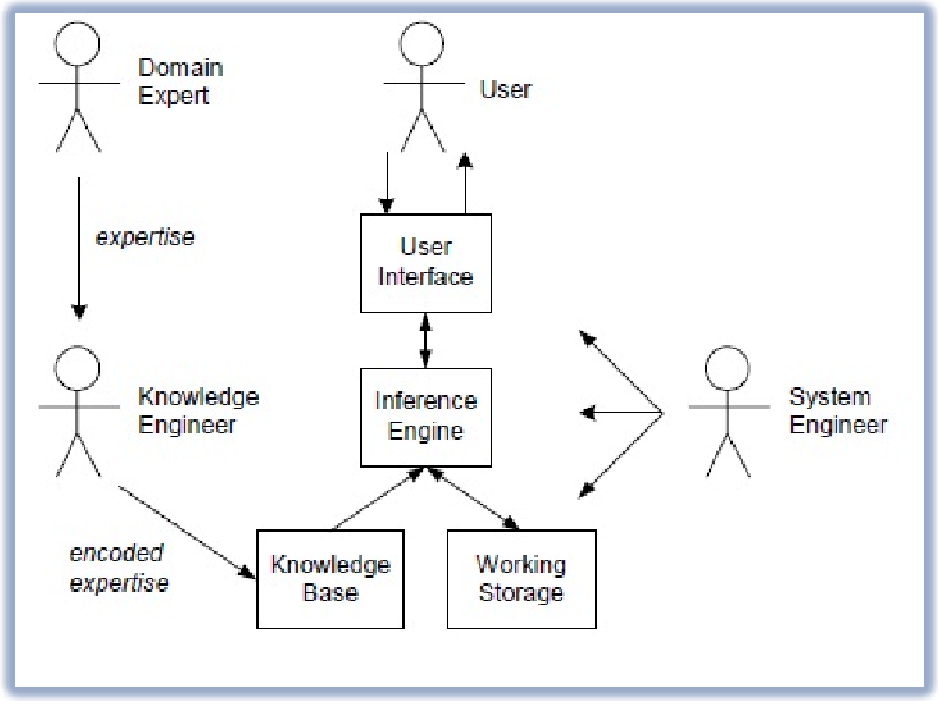
# Sisteme Expert vs. Sisteme Bazate pe Cunostinte

Termenul de "sistem expert" se utilizează cu un înţeles identic termenului de "*sistem bazat pe cunoştinţe*" sau *Knowledge-Based Expert Systems*, deşi în literatura de specialitate sunt menţionate deosebiri între aceste sisteme din punct de vedere al detaliilor constructive şi al funcţionalităţii.

In principal un sistem expert (SE) este un program de calculator care utilizeaza *cunostinte* pe baza carora obtine deductii folosind un set de *proceduri de rationament*.

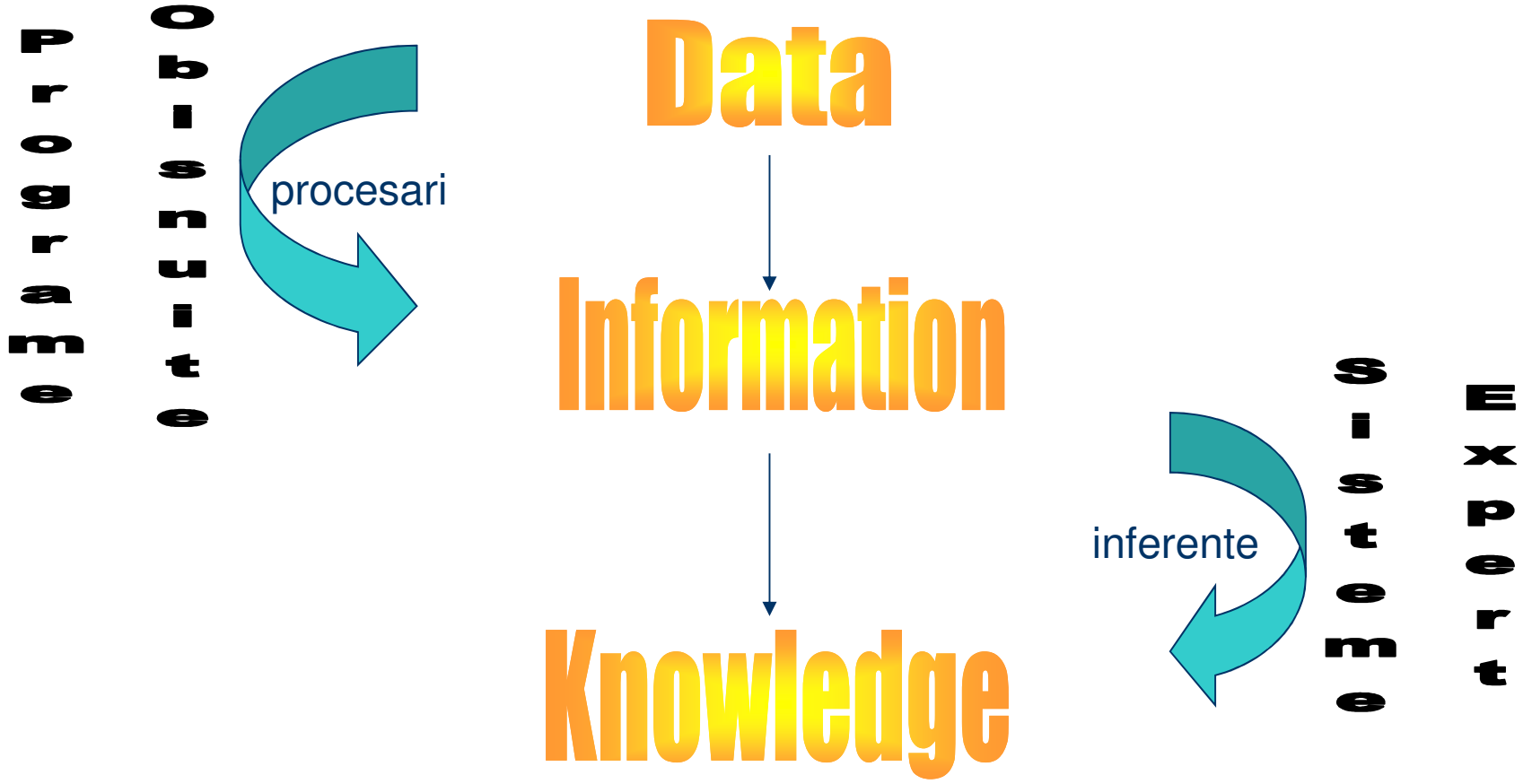
In functie de modul in care se reprezinta rationamentul, putem avea SE bazate de reguli (cele mai folosite), SE cu frame-uri, etc.

# Componentele unui SE versus utilizatorii umani




© Dennis Merritt „Building Expert System in Prolog”

# Sisteme Expert vs. “the Others”



# Sisteme Expert vs. “the Others”

Cea mai mare deosebire dintre programele convenţionale şi sistemele expert constă în faptul că programele convenţionale operează cu **date**, în timp ce sistemele expert operează cu **cunoştinţe**.

Mijloacele prin care sistemele expert îşi ating obiectivele se bazează pe un set de cunoștințe prealabile și a unor reguli analitice stabilite de experți.

Instrumentele folosite în dezvoltarea sistemelor expert se împart în două categorii principale:

- *limbaje de nivel înalt de prelucrare simbolică*
- *sisteme expert de uz general* sau sisteme cadru.

Cele mai răspândite limbaje de nivel înalt sunt LISP şi PROLOG iar ca sistem cadru vom studia pe cel oferit de mediul de programare GURU.

http://www.agir.ro/univers-ingineresc/sisteme_expert_1249.html

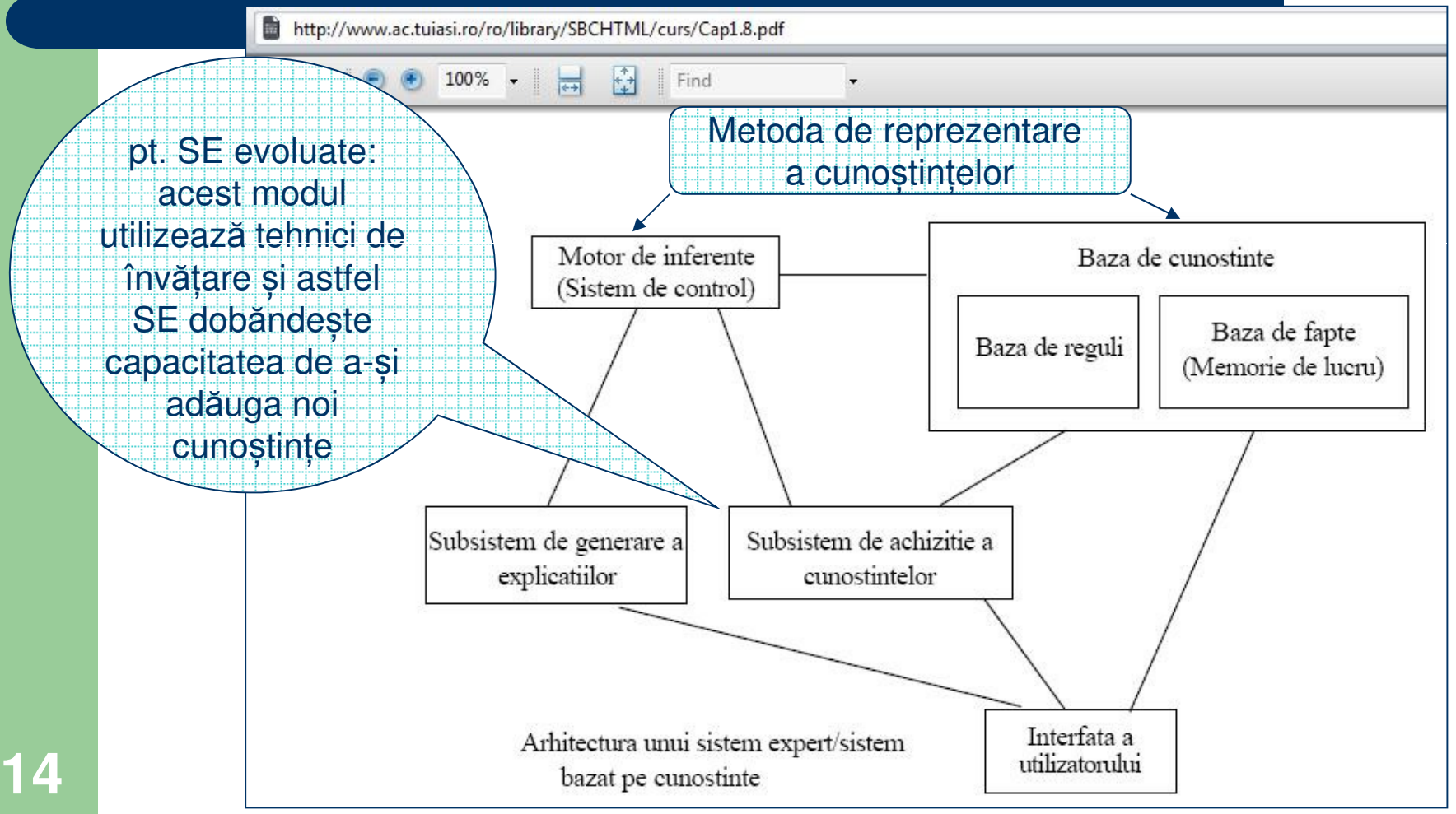
# Sisteme Expert

În prezent, sistemele expert sunt tot mai mult asociate direct cu modurile de raționare și cunoștințele reprezentate. Sistemele expert sunt de fapt dezvoltări în noile câmpuri ale achiziției de cunoștințe ca forme specializate de învățare, în care cunoștințele sunt achiziționate direct de la expert. Alte sisteme expert, ca tehnici de IA, includ explicații, învățare inteligentă, planificare, rezolvarea problemelor distribuite, cercetări ce sunt adresate direct IA-ului.

Organizaţiile care desfăşoară activităţi comerciale încep să acorde o atenţie din ce în ce mai mare încercărilor de a ajuta inteligenţa umană, cunoştinţele personalului, experienţa sa cu mijloacele şi tehnicile utilizate de sistemele expert.

Putem spune că, în general, sistemele expert includ *limbajul natural*, *roboţii industriali*, precum şi *computerele inteligente*.

# Arhitectura unui Sistem Expert



# Reprezentarea cunoștințelor

Reprezentarea cunoştinţelor într-un calculator constă în găsirea unor corespondenţe între lumea exterioară şi sistemul simbolic ce permite execuţia raţionamentelor.

Etapele reprezentării cunoaşterii

- Stabilirea sistemului de reprezentare
- Stabilirea sistemului de clasificare ce urmăreşte gruparea cunoștințelor în clase
- Stabilirea sistemului de organizare privitor la gruparea şi ordonarea elementelor şi claselor, precum şi asigurarea efectuării unor operații fundamentale cum sunt: acces, căutare, interferenţă, etc.

Se pot distinge, pentru aplicaţiile de inteligenţă artificială, trei clase de metode de reprezentare:

- metode logice (enunţuri adevărate);
- metode relaţionale (grafuri şi reţele);
- metode procedurale (proceduri).

http://www.ac.tuiasi.ro/ro/library/SBCHTML/curs/Cap1.5.pdf

# Avantajele Sistemelor Expert

1) disponibilitatea crescută, expertiza devenind accesibilă pe orice calculator adecvat. Se spune despre **SE** că determină producția de masă a expertizei: **SE** este disponibil în orice moment și nu este afectat de emotivitate, factori de stres.
2) reducerea costului, explicabilă în corelație cu ideea de mai sus și tinând seama de prețul sistemelor de calcul.
3) Permanența, **SE** neavând o viață limitată.
4) reducerea pericolelor, în sensul că **SE** pot fi folosite în medii periculoase pentru om, atunci când sunt încorporate în sisteme adecvate (de exemplu, în sistemul de comandă al roboților industriali).
5) cresterea calității, prin obținerea unei expertize complete a domeniului de lucru, ceea ce expertul uman nu face întotdeauna.
6) posibilitatea de explicare în detaliu a solutiei obtinute pentru cresterea gradului de încredere a utilizatorului, atunci când **SE** este dotat cu un sistem de generare a explicatiilor.
7) rapiditatea, în sensul că pe sisteme software și hardware adecvate se pot obține, pentru anumite domenii, performante de timp mai bune decât cele ale expertilor umani.
8) SE pot contribui la raspândirea cunostintelor din domeniul de lucru; astfel, un utilizator poate rula programul pentru mai multe probleme, de la simplu la complex și urmărind soluțiile și explicațiile date de **SE** se poate instrui în domeniul respectiv.
9) SE determină păstrarea în siguranta a cunoștințelor; marile firme și-au construit **SE** pentru a fi afectate într-o mai mica masura de plecarea unor experti umani.
10) clasificarea și perfecționarea metodelor de rezolvare a problemelor din domeniul de lucru apare ca un beneficiu indirect, obținut o dată cu construirea unui **SE**.

# Sisteme Expert. Exemple

Primele sisteme expert dezvoltate în domenii aplicative au fost DENDRAL, destinat analizei structurilor moleculare, MYCIN, un sistem expert pentru diagnosticul şi tratamentul infecţiilor sanguine, sistemele EMYCIN, HEADMED, CASNET şi INTERNIST pentru domeniul medical, PROSPECTOR pentru evaluarea prospecţiunilor şi forajelor geologice, sau TEIRESIAS pentru achiziţia inteligentă a cunoaşterii.

# DENDRAL

## The DENDRAL Problem

- *Chemist is presented with an unknown chemical compound*
- *Chemist must determine the molecular structure*
- *Therefore needs to find out which atoms are in the structure*
- *Needs to know how the atoms are connected to form molecules*

Determină structura unei molecule plecând de la informațiile date de un spectograf.

DENDRAL este considerat primul sistem expert fiind prezentat în 1960 la Universitatea Stanford.

http://www.referat.ro/referate/Sisteme_Expert_-_Mycin_4213.html

# MYCIN

Mycin a fost unul dintre primele SE, și modul său de proiectare a influențat masiv designul SE comerciale care i-au urmat.

Mycin a fost dezvoltat la Universitatea Stanford in anii ’70. Rolul său era de a diagnostica și recomada tratamente pentru diferite infectii ale sangelui. Pentru a pune un diagnostic, trebuiau cultivate organismele care infectau pacientul. Din pacate, treaba asta dura cel putin 48 ore, asa că dacă doctorii ar fi asteptat atât, pacientul ar fi putut muri! De aceea doctorii erau nevoiti sa “ghicească” repede problemele ce puteau apărea și să furnizeze un tratament acoperitor pentru toate aceste probleme.

Mycin utilizează cunoștințe incerte și este realizat în LISP.

# MYCIN

## The Decision Process

- *There are four questions in the process of deciding on treatment:*
  - Does the patient have a significant infection?
  - What are the organism(s) involved?
  - What set of drugs might be appropriate to treat the infection?
  - What is the best choice of drug or combination of drugs to treat the infection?

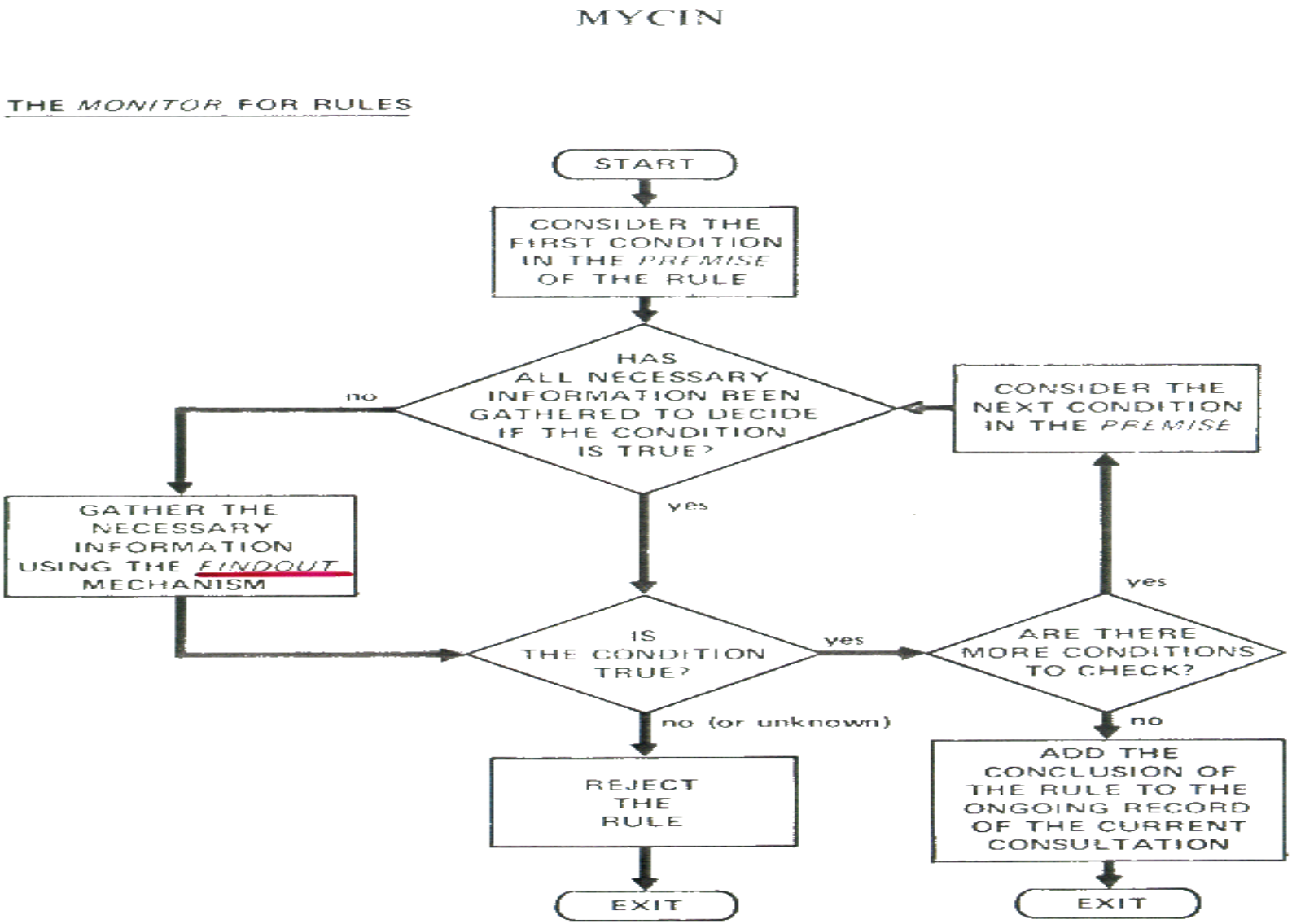

# Exemplu de regula MYCIN

RULE060

IF: THE IDENTITY OF THE ORGANISM IS BACTEROIDES
THEN: I RECOMMEND THERAPY CHOSEN FROM AMONG THE FOLLOWING DRUGS:

| | |
|---|---|
| 1 - CLINDAMYCIN | (.99) |
| 2 - CHLORAMPHENICOL | (.99) |
| 3 - ERYTHROMYCIN | (.57) |
| 4 - TETRACYCLINE | (.28) |
| 5 - CARBENICILLIN | (.27) |

# Arborele de Decizie MYCIN



# Domenii de aplicabilitate pentru SE

| FUNCȚIA | SPECIFICAȚIA |
|---|---|
| CONTROL Și MONITORIZARE | Controlul inteligent al sistemelor (automat) |
| DEPANARE Și REPARARE | Recomandă corecții ale deficiențelor funcționării sistemelor. |
| DIAGNOSTIC Și ÎNTREȚINERE | Localizează erorile de funcționare și recomandă corecțiile necesare. Îmbunătățirea performanțelor celor care învață folosind strategiile CAD (Computer Asisted Instruction) |
| INTERPRETARE/ PLANIFICARE | Identificarea unor situații pe baza semnalelor de la senzori. |
| PREDICȚIE | Inferarea unor situații probabile pe baza informațiilor deja cunoscute. |
| SIMULARE | Deducerea consecințelor acțiunilor sau evenimentelor generate de sistemul însuși. |
| CLASIFICARE (TAXONOMIE) | Organizarea entităților (obiectelor) pe clase și categorii. |
| SELECȚIE | Identificarea celei mai bune alternative dintr-o listă de posibilități. |

# Metoda regulilor de producție

*Regulile de producţie sunt enunţuri condiţionale uşor de înţeles şi de descris, care specifică o acţiune care va avea loc sub rezerva adevărului condiţiilor*. În vocabularul inginerilor de cunoştinţe, ele se mai numesc relaţii **cauză** – **efect**, **premisă** – **concluzie**, **ipoteză** – **acţiune**, **condiţie** – **acţiune**, **test** – **rezultat**, **IF** – **THEN** sau **IF-THEN**– **ELSE**.

Regulile de forma:

*Dacă premisă atunci concluzie.*

reprezintă reguli **deductive**, iar regulile de forma:

*Concluzie dacă premisă.*

reprezintă reguli **inductive**.

# Reguli de producție

Regulile de producţie **diferă** de instrucţiunile **IF-THEN-ELSE** din limbajele de programare imperative, deoarece ele sunt relativ independente una de alta şi sunt bazate pe euristici, pe raţionamentul izvorât din experienţă şi nu pe algoritmi. Ele pot încorpora chiar şi incertitudine, o valoare numerică atribuită de expert, rezultată din experienţă sa în soluţionarea problemelor.

În cadrul metodei regulilor de producţie cunoştinţele sunt reprezentate prin două tipuri de entităţi: **reguli** şi **fapte**.

**Regulile** sunt cunoştinţe reprezentate de implicaţii şi exprimă cunoştinţe generale referitoare la domeniul problemei de rezolvat. De exemplu:

***Dacă*** *rata inflaţiei creşte şi cererea de credite creşte* ***atunci*** *rata dobânzii creşte.*

**Faptele** sunt aserţiuni unitare şi reprezintă cunoştinţele specifice care descriu un caz particular, de exemplu, o instanţă a problemei de rezolvat. De exemplu:

*MPIA este o disciplină studiată de studenţii anului III.*

# Sisteme Expert Bazate pe Reguli

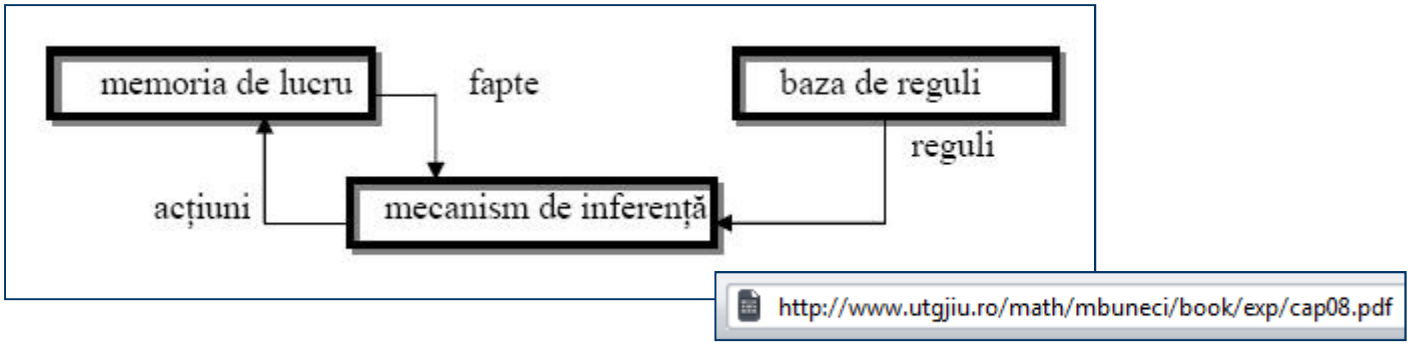
Mijloacele prin care **SE** îşi ating obiectivele se bazează pe mulţimi de fapte şi reguli euristice care au scopul de a prelucra cunoştinţelor. Un **SE** bazat pe reguli are trei componente:

- ***baza de cunoştinţe*** (sau ***baza de reguli***) care descrie domeniul de expertiză în care se aplică sistemul expert; reprezentarea cunoştinţelor se face utilizând ***reguli*** de forma:

  **IF** condiţie$_1$ şi … şi condiţie$_k$ **THEN** acţiune

  (condiţie$_1$ şi … şi condiţie$_k$ se numesc premisele regulii)
- ***baza de fapte*** a sistemului; trebuie făcută distincţie între ***faptele iniţiale***, care reprezintă datele iniţiale ale problemei de rezolvat şi ***faptele dinamice***, care sunt deduse pe măsura derulării procesului de rezolvare a problemei.
- ***mecanismul de inferenţă*** care reprezintă componenta de control şi execuţie dintr-un sistem bazat pe reguli; acest mecanism stabileşte ordinea de selecţie a regulilor din baza de cunoştinţe şi aplicarea lor; efectul acestei declanşări este modificare memoriei de lucru.

# Sisteme Expert Bazate pe Reguli. Arhitectura

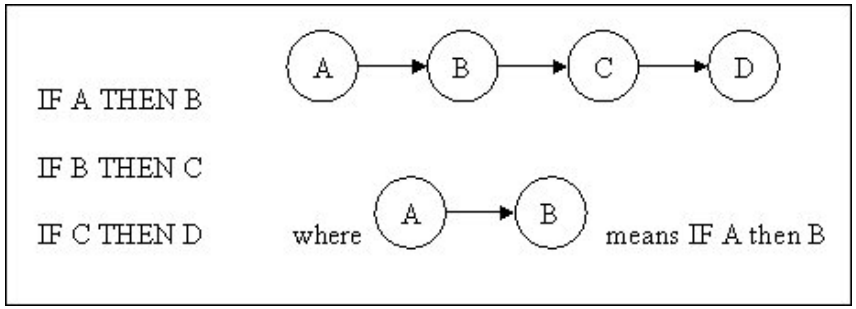


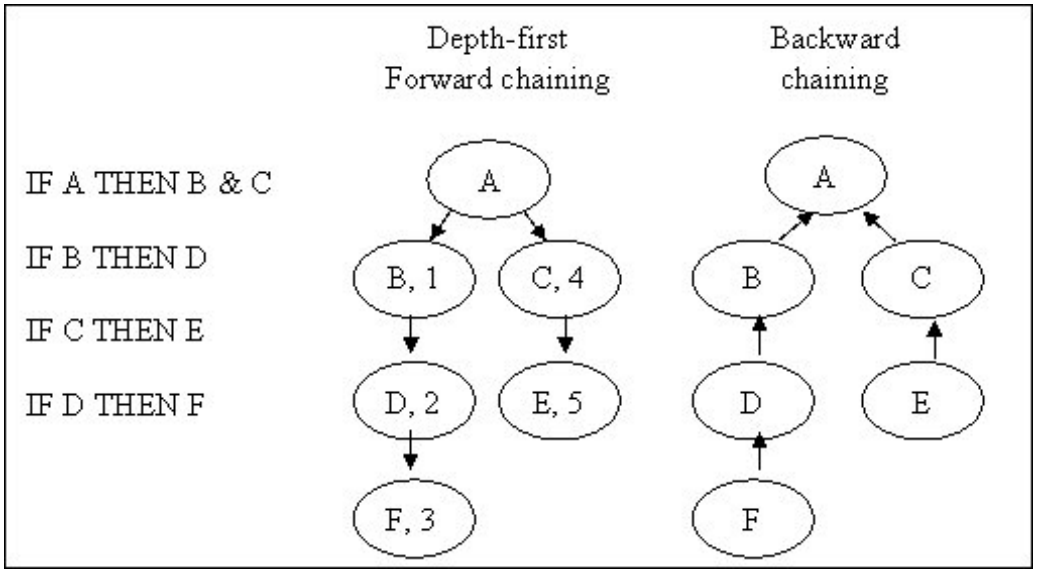

Mecanismul de inferenţă poate folosi trei moduri de raţionament:

- ***forward***, dirijat de fapte (***inferenţă cu înlănţuire înainte***)
- ***backward***, dirijat de scop (***inferenţă cu înlănţuire înapoi***)
- ***mixt***

http://www.cobblestoneconcepts.com/ucgis2summer2002/choi/choi.htm

# Inferența Forward și Backward



# Avantajele și Limitările Regulilor

- Intr-un SE bazat pe reguli, cunoştinţele sunt salvate în reguli. De aceea, regulile trebuie să deţină o importantă parte a informaţiilor sistemului în aşa fel încât să poată modela o problemă complexă.
- Structura de control a unei reguli trebuie să imite câteva strategii umane de rezolvare a problemelor, şi astfel, regulile salvate într-un SE să poată fi utilizate în rezolvarea problemelor.

Totuşi, există câteva limitări ale sistemelor bazate pe reguli. Câteodată este dificil să obţinem un set corect de reguli. Există câteva cauze.

Deşi o regulă presupune faptul că există un corp general acceptat de informaţie explicită într-un domeniu, există multe domenii în lumea reală care nu au modele cauzale şi nici reguli explicite general acceptate. Pe scurt, **nu toate lucrurile/evenimentele pot fi controlate**!

# SE în PROLOG

Prolog-ul are implementat un raționament de tip **backward** sau **goal-driven** care se pretează multor SE. Fiecare regula (clauză) în Prolog are un scop iar acesta derivă din unul sau mai multe sub-scopuri.

Regulile unui SE se scriu de maniera următoare:

IF $premisa_1$ AND $premisa_2$ ... THEN concluzie

În Prolog, o astfel de regulă se scrie astfel:

concluzie :- $premisa_1$, $premisa_2$, ...

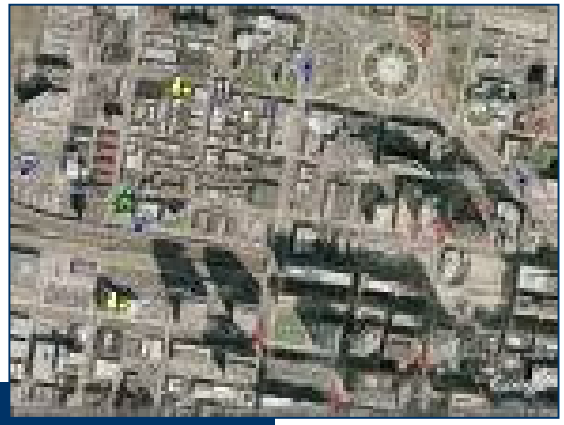
Baza de cunoștințe a oricărui SE se traduce în termeni Prolog în baza de fapte.

# Aplicație

Sistem Expert pentru Interpretarea Fotografiilor Spațiale

http://www.cobblestoneconcepts.com/ucgis2summer2002/choi/choi.htm

# Interpretarea Fotografiilor Spațiale



Există patru elemente de bază pentru interpretarea fotografiilor aeriene: *tonul*, *textura*, *modelul*, *forma*, *umbra*, *mărimea* şi *situaţia* (Avery şi Berlin, 1992). *Tonul* reprezintă densitatea strălucirii. *Textura* este frecvenţa schimbărilor de ton în cadrul fotografiilor. *Modelul* este aranjamentul spaţial al obiectelor. *Forma* este forma generală a unui obiect. *Umbra* dezvăluie forma şi înălţimea unui obiect. *Mărimea* unui obiect este legată de cea a celorlalte. *Situaţia* este poziţia unui obiect în relaţie cu cele din vecinătatea apropiată.

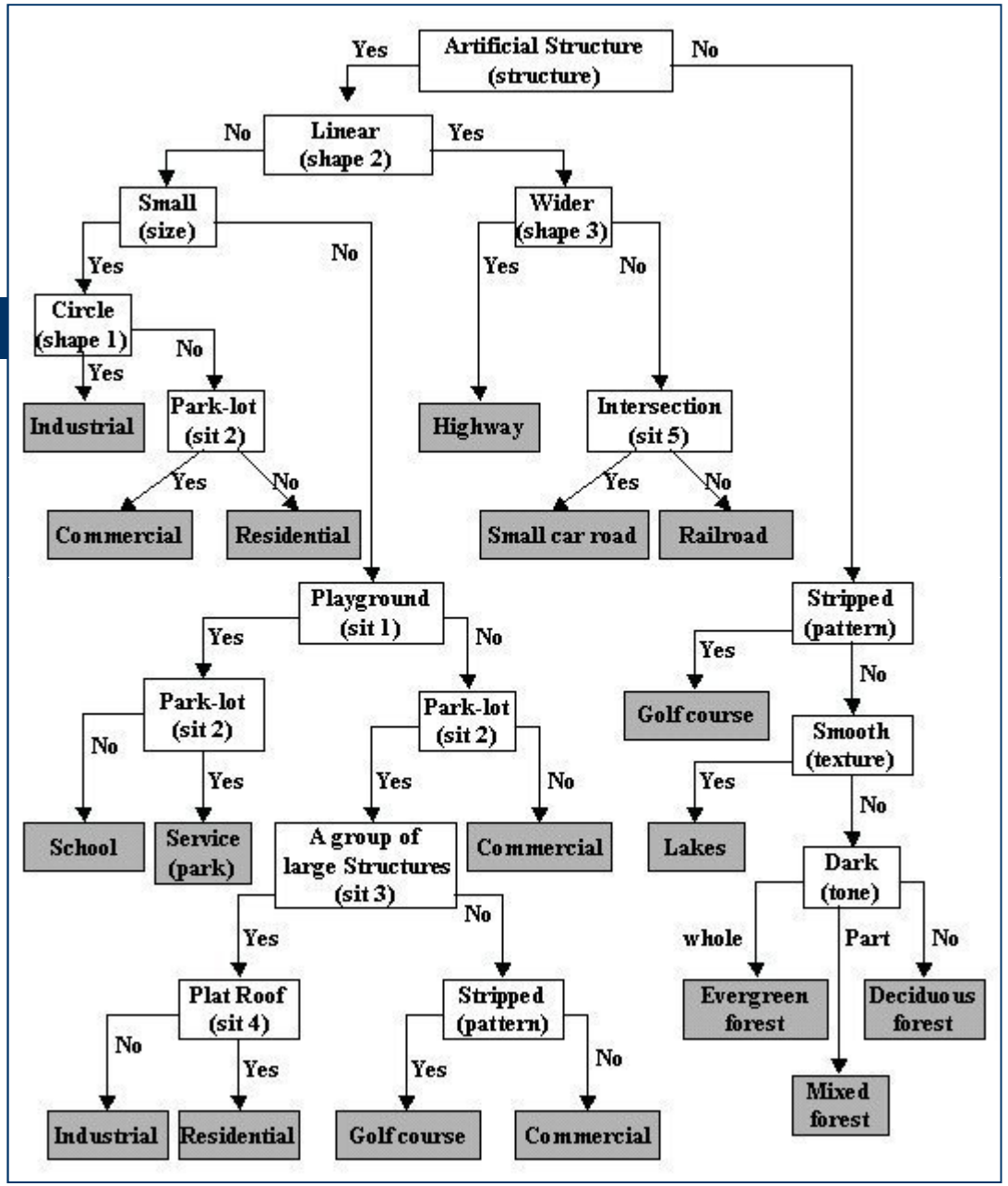
Din cele şapte elemente ale interpretării unei fotografii aeriene, şase elemente sunt folosite în această aplicaţie: *tonul*, *textura*, *modelul*, *mărimea*, *forma* şi *situaţia*. Umbra nu poate fi folosită în interpretare datorită scării mici a fotografiei aeriene.

# Raționamentul spațial

- *Situaţia* este folosită pentru a separa utilităţile urbane de alte elemente precum lacuri sau păduri.
- Dacă sunt structuri artificiale (clădiri sau case) această *aria* și *forma* sunt folosite pentru delimitarea spaţiilor comerciale și industriale de celelalte structuri.
- *Aria* cu păduri este separată de ton.
- Cele mai dificile categorii de interpretare sunt căile de comunicație rutiere şi feroviare datorită scării mici a fotografiilor aeriene. Căile feroviare pot fi deosebite de drumurile mici între zonele rezidenţiale prin faptul că primele au mai puţine intersecţii decât drumurile.

http://www.cobblestoneconcepts.com/ucgis2summer2002/choi/choi.htm

# Arborele de decizie



# Cunoștințele primare ale sitemului

Vom nota cu IP = {$i_1$ , ……, $i_{13}$} o mulţime de simboluri, fiecare simbol reprezentând o întrebare care se poate pune utilizatorului. Răspunsurile utilizatorului la întrebările primite reprezintă cunoștințele primare ale sistemului.

- i1: (STRUCTURE):Is there artificial structure?
- i2: (SIZE): Is the structure small?
- i3: (SHAPE1): Is the structure circle?
- i4: (SITUATION1): Is the playground adjacent to the structure?
- i5: (SITUATION2): Is there large parking lots adjacent to the structure?
- i6:(SITUATION3): Is there some large structures near to the structure?
- i7: (SITUATION4): Does the structure have flat roof?
- i8: (SHAPE2): Does it looks like long linear feature?
- i9: (SHAPE3): Is the line feature wider then the other line features?
- i10: (SITUATION5): Is there many intersections/structures along the line feature?
- i11: (PATTERN): Is it stripped irregularly with dark and white tone?
- i12: (TEXTURE): Is the surface of the feature smooth?
- i13: (TONE): Is the tone darker than other similar features?

# Cunoștințele finale ale sistemului

Vom nota cu PR = {$p_1$ , ……, $p_{13}$} o mulţime de simboluri, fiecare reprezentând o problemă pe care o poate rezolva sistemul.

- p1: It is a industrial area.
- p2: It is a commercial area.
- p3: It is a residential area.
- p4: It is a school.
- p5: It is a park.
- p6: It is a golf course.
- p7: It is a evergreen forest.
- p8: It is a deciduous forest.
- p9: It is a mixed forest.
- p10: It is a lake.
- p11: It is railroad.
- p12: It is a small car road.
- p13: It is a highway.

# Implementarea

Limbajul de programare utilizat pentru implementarea aplicației este Prolog iar interfața grafică a fost construită folosind pachetul grafic XPCE.

Utilizatorul, pe baza unui dialog cu expertul, va putea să identifice categoriile spațiale ce apar în imagine. Dialogul dintre sistemul expert și utilizator se face de maniera următoare:

- sistemul generează mai multe întrebări pentru utilizator; procesul de generare al întrebărilor este “dictat” atât de arborele de decizie cât și de motorul de inferențe
- utilizatorul răspunde la întrebările sistemului apăsând butonul corespunzător răspunsului dorit.
- Rezultatul raționamentului este afișat tot într-o fereastră de dialog acesta fiind însoțit și de o imagine sugestivă.

# Afișarea soluției


Sist_diag - Shortcut.lnk

# Implementarea în Prolog

- **rule**([$x_1$, ..., $x_n$], $y$) reprezinta regula IF $x_1 \wedge ... \wedge x_n$ THEN y
- **lista_probleme**([$p_1$, ... , $p_n$]) specifică mulțimea problemelor pe care le poate diagnostica sistemul
- **problema**($p_i$, *Text*) specifică textul care va fi afisat de sistem ca răspuns în cazul în care problema $p_i$ a fost diagnosticată de sistem
- **intrebare**($i_j$, *Text*) specifică faptul că $i_j$ reprezintă simbolul atașat unei intrebări care poate fi pusă utilizatorului, iar *Text* este conținutul intrebării
- **intrebare_raspuns**($s_j$, $i_k$, *da*) sau **intrebare_raspuns**($s_j$, $i_k$, *nu*) specifică faptul că $s_j$ este o cunoștință primară obținută prin raspunsul *da*, respectiv *nu* la întrebarea $i_k$ adresata utilizatorului
- **si**([$d_1$, ..., $d_n$]) reprezinta lista simbolurilor intermediare.

# Vă mulțumesc!