



UNIVERSITATEA BABEȘ-BOLYAI
Facultatea de Matematică și Informatică



INTELIGENȚĂ ARTIFICIALĂ

Sisteme inteligente

Sisteme bazate pe reguli în medii certe

Laura Dioșan

Sumar

A. Scurtă introducere în Inteligența Artificială (IA)

B. Rezolvarea problemelor prin căutare

- Definirea problemelor de căutare
- Strategii de căutare
 - Strategii de căutare neinformate
 - Strategii de căutare informate
 - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
 - Strategii de căutare adversială

C. Sisteme inteligente

- Sisteme care învață singure
 - Arbori de decizie
 - Rețele neuronale artificiale
 - Mașini cu suport vectorial
 - Algoritmi evolutivi
- Sisteme hibride
- Sisteme bazate pe reguli în medii certe
- Sisteme bazate pe reguli în medii incerte (Bayes, factori de certitudine, Fuzzy)

Materiale de citit și legături utile

- ❑ capitolul III din *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- ❑ capitolul 4 și 5 din *H.F. Pop, G. Șerban, Inteligență artificială, Cluj Napoca, 2004*
- ❑ capitolul 2 din *Adrian A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001*
- ❑ capitolul 6 și 7 din *C. Groșan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*

Conținut

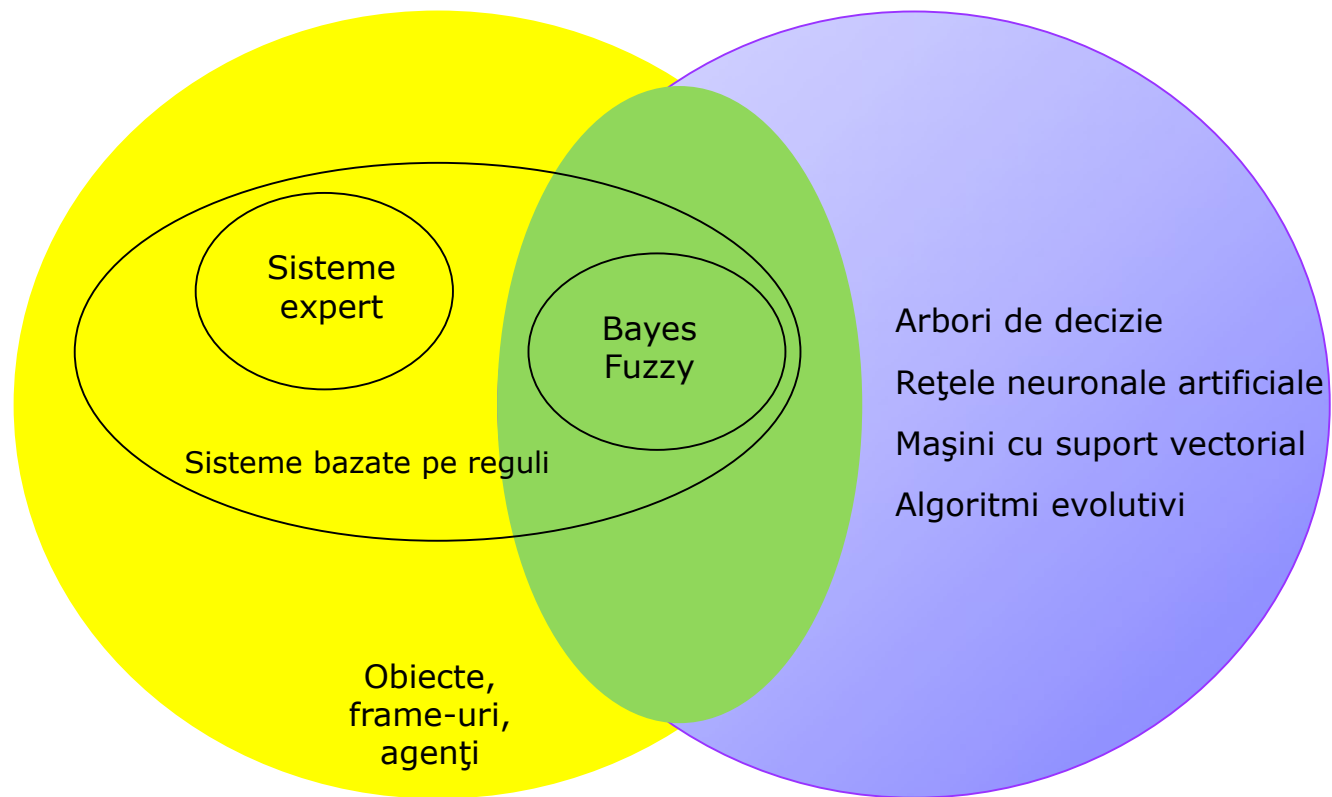
- Sisteme inteligente

- Sisteme bazate pe cunoștințe

- Sisteme bazate pe logică

- Sisteme bazate pe reguli în medii certe

Sisteme inteligente



Sisteme bazate pe cunoștințe (SBC)

Inteligență computațională

Sisteme inteligente – sisteme bazate pe cunoștințe (SBC)

- sistemele computaționale – alcătuite din 2 module principale (roluri):
 - Domeniul de cunoștințe (baza de cunoștințe – BC – *knowledge base*)
 - Informațiile specifice despre un domeniu
 - Modulul de control (MC – *inference engine*)
 - Regulile prin care se pot obține informații noi
 - Algoritmi independenți de domeniu

Sisteme inteligente – SBC

Baza de cunoștințe (BC)

- Conținut
- Tipologie
- Modalități de reprezentare a cunoștințelor
- Modalități de stocare a cunoștințelor

Sisteme inteligente – SBC

Baza de cunoștințe (BC)

□ Conținut

- Informații (exprimate într-o anumită reprezentare – ex. propoziții) despre mediu
- informații necesare pentru înțelegerea, formularea și rezolvarea problemelor
- mulțime de propoziții (exprimate/reprezentate într-un limbaj formal) care descriu mediul
 - reprezentare ușor interpretabilă de către calculator → limbaj de reprezentare a cunoștințelor
 - mecanismul de obținere a unor propoziții noi pe baza celor vechi → inferență/raționare

□ Tipologie

- cunoștințe exacte (perfecte)
- cunoștințe imperfecte (nesigure, incerte)
 - Inexacte
 - Incomplete
 - Incomensurabile

□ Modalități de reprezentare a cunoștințelor

- Logica formală (limbaje formale)
- Reguli
- Rețele semantice

Sisteme inteligente – SBC

Baza de cunoștințe (BC)

- Modalități de reprezentare a cunoștințelor
 - Logica formală (limbaje formale)
 - Definiție
 - Știința principiilor formale de raționament
 - Componente
 - Sintaxă – simbolurile atomice folosite de către limbaj și regulile de construcție a expresiilor (structurilor/propozițiilor) limbajului
 - Semantică – asociază un înțeles simbolurilor și o valoare de adevăr (Adevărat sau Fals) regulilor (propozițiilor) limbajului
 - Metodă de inferență sintactică – regulile necesare determinării unei submulțimi de expresii logice → teoreme (folosite pentru obținerea de noi expresii)
 - Tipologie
 - În funcție de numărul valorilor de adevăr:
 - logică duală
 - logică polivalentă
 - În funcție de tipul elementelor de bază:
 - clasică → primitivele = propoziții (predicate)
 - probabilistică → primitivele = variabile aleatoare
 - În funcție de obiectul de lucru:
 - logica propozițională → se lucrează doar cu propoziții declarative, iar obiectele descrise sunt fixe sau unice (Ionică este student)
 - logica predicatelor de ordin I → se lucrează cu propoziții declarative, cu predicate și cuantificări, iar obiectele descrise pot fi unice sau variabile asociate unui obiect unic (Toți studenții sunt prezenți)
 - Reguli
 - Rețele semantice

Sisteme inteligente – SBC

Baza de cunoștințe (BC)

□ Modalități de reprezentare a cunoștințelor

- Logica formală (limbaje formale)
- Reguli
 - Euristicii speciale care generează informații (cunoștințe)
 - O modalitate de exprimare (reprezentare) a cunoștințelor
 - Ex. *dacă Ionică lucrează la Facebook, atunci el câștigă mulți bani și are puțin timp liber*
 - Interdependențele între reguli → rețea de inferență
 - Fac legătura între cauză și efect - memorate în calculator sub forma unor structuri de control
IF cauză THEN efect

■ Rețele semantice

- Grafuri orientate cu noduri care conțin concepte și arce care reprezintă relații semantice între concepte precum:
 - *Meronymy* (A este meronym al lui B dacă A este o parte a lui B)
 - Ex. Degetul este un meronym al mâinii, roata este un meronym al mașinii
 - *Holonymy* (A este holonym al lui B dacă B este o parte a lui A)
 - Ex. Copacul este un holonym al scoarței
 - *Hyponymy* (A este hyponym al lui B dacă A este un fel de B)
 - Ex. Tractorul este un hyponym al autovehiculului
 - *Hypernymy* (A este hypernym al lui B dacă A este o generalizare al lui B)
 - Ex. Fructul este un hypernym al portocalei
 - *Synonymy* (A este sinonim al lui B dacă A denotă același lucru ca B)
 - Ex. A alergia este sinonim cu a fugi
 - *Antonymy* (A este antonim al lui B dacă A denotă lucruri opuse ca B)
 - Ex. Uscat este antonim cu ud

Sisteme inteligente – SBC

Baza de cunoștințe (BC)

- Modalități de stocare a cunoștințelor
 - Relații
 - Simple → baze de date
 - Ierarhice → ierarhii de concepte (rețele semantice)
 - Logică formală
 - Reguli
 - Logică procedurală
 - Algoritmi

Sisteme inteligente – SBC

Modulul de control (MC)

□ Conținut

- Responsabil cu inferența
- A ajunge la o concluzie plecând de la anumite premise (cunoștințe) și aplicând anumite reguli de inferență
- MC depinde de complexitate și tipul cunoștințelor cu care are de-a face

□ Tipologie

■ În funcție de direcția inferenței:

- MC cu legătură înainte (*forward chaining*)
 - Pornesc de la informația disponibilă (fapte date, condiții) și încearcă să ajungă la o concluzie (fapte derivate)
 - Se bazează pe date (*data driven*)
- MC cu legătură înapoi (*backward chaining*)
 - Pornesc de la o concluzie potențială (ipoteză) și caută evidențe care să o suporte-contrazică (explicații)
 - Se bazează pe scop (*goal driven*)

□ Tehnici de raționare (tehnici de inferență)

■ În medii certe

- bazate pe logică
- bazate pe reguli

■ În medii incerte

- bazate pe teoria probabilităților
- bazate pe teoria posibilității

Inteligență artificială - sisteme bazate pe reguli (în medii certe)

Sisteme inteligente – SBC

Tipologia SBC

- ❑ **Sisteme bazate pe logică (SBL)**
- ❑ **Sisteme bazate pe reguli (SBR)**
- ❑ *Case-based reasoning*
- ❑ *Hypertext manipulating systems*
- ❑ *Data bases and intelligent UI*
- ❑ *Intelligent tutoring systems*

Sisteme inteligente – SBC

Sisteme bazate pe logică (SBL)

- Conținut și obiective
- Arhitectură
- Tipologie
- Tool-uri
- Avantaje și limite

Sisteme inteligente – SBC – SBL

Conținut și obiective

□ Conținut

- explorează o multitudine de cunoștințe date pentru a obține concluzii noi despre activități dificil de examinat, folosind metode specifice logicii formale
- Un sistem logic este alcătuit din:
 - limbaj (sintaxă + semantică)
 - metodă de deducție (inferență)

□ Scopul SBL

- Rezolvarea de probleme cu ajutorul programării declarative
 - descriind ceea ce este adevărat sau nu în rezolvarea problemelor
 - permițând tehnici de raționare automată
- Exemple de probleme rezolvate de SBRL
 - demonstrarea automată a teoremelor

□ De ce se studiază SBL?

- Logica formală este precisă și definită

Sisteme inteligente – SBC – SBL

Arhitectură

□ Baza de cunoștințe (BC)

■ Sintaxă

- simbolurile atomice folosite de către limbaj și regulile de construcție a expresiilor (structurilor/propozițiilor) limbajului

■ Semantică

- asociază un înțeles simbolurilor și o valoare de adevăr (Adevărat sau Fals) regulilor (propozițiilor) limbajului

□ Modulul de control (MC)

- Metodă de inferență sintactică – regulile necesare determinării unei submulțimi de expresii logice → teoreme (folosite pentru obținerea de noi expresii)

Sisteme inteligente – SBC – SBL

Tipologie

□ Sisteme bazate pe logica propozițională

- se lucrează doar cu propoziții declarative
- obiectele descrise sunt unice și fixe (*Ionică este student*)

□ Sisteme bazate pe logica predicatelor de ordin I

- se lucrează cu propoziții declarative, cu predicate (*Student(a)*) și cuantificări (variabile cuantificabile \rightarrow *pentru orice a, Student(a) \rightarrow AccesWiFi(a)*)
- obiectele descrise pot fi unice sau dinamice (variabile asociate unui obiect unic – *Toți studenții sunt prezenți*)
- predicatele au argumente simple (*Student(a)*)

□ Sisteme bazate pe logica predicatelor de ordin superior (≥ 2)

- se lucrează cu propoziții declarative, cu predicate și cuantificări (variabile cuantificabile)
- permit variabilelor să reprezinte mai multe relații între obiecte
- predicatele pot avea argumente simple, argumente de tip predicat (*StudentSenator(Student(a))*) sau argumente de tip funcție (*Bursier(a are media peste 9.50)*)

□ Sisteme temporale

- Reprezintă valoarea de adevăr a faptelor de-a lungul timpului (*Ionică este uneori grăbit*)

□ Sisteme modale

- Reprezintă și fapte îndoielnice (*Ionică poate să promoveze examenul*)

Sisteme inteligente – SBC – SBL

Sisteme bazate pe logica propozițională

□ Baza de cunoștințe

■ Poate fi alcătuită din:

- Simboluri (A, B, P, Q, ...)
- Propoziții (formule)
 - definite astfel:
 1. Un simbol
 2. Dacă P este o propoziție, atunci și $\neg P$ este tot propoziție
 3. Dacă P și Q sunt propoziții, atunci $P \wedge Q$, $P \vee Q$, $P \Rightarrow Q$, $P \Leftrightarrow Q$ sunt tot propoziții
 4. Un număr finit de aplicări ale regulilor (1) – (3)
 - Interpretare a unei propoziții \rightarrow stabilirea valorii de adevăr
 - Model \rightarrow interpretare a unei mulțimi de propoziții astfel încât toate propozițiile să fie adevărate

□ Modulul de control

■ realizează inferența

- stabilirea valorii de adevăr a unei propoziții obiectiv pe baza informațiilor din BC

■ în mai multe moduri

- verificarea modelului
 - enumerarea tuturor combinațiilor posibile pentru valorile de adevăr ale simbolurilor și propozițiilor implicate în SBL
- deducția modelului cu ajutorul regulilor de inferență

Sisteme inteligente – SBC – SBL

Sisteme bazate pe logica propozițională – modulul de control

□ Problemă

- Se dă o BC = {P1, P2, ..., Pm} formată din simbolurile {X1, X2, ..., Xn} și o propoziție obiectiv O.
- Se poate deduce O din BC?

□ Verificarea modelului

■ Etape

- Se construiește tabelul corespunzător tuturor combinațiilor posibile pentru valorile de adevăr ale simbolurilor
- Se determină dacă toate modelele BC sunt și modele ale lui O
 - Modelele BC – acele linii în care toate propozițiile din BC sunt *adevărate*

■ Exemplu

- P = *Afară este foarte cald*
- Q = *Afară este umezeală*
- R = *Afară plouă*
- BC = {P ∧ Q ⇒ R, Q ⇒ P, Q}
- R = *Va ploua?*

P	Q	R	P ∧ Q ⇒ R	Q ⇒ P	Q	BC	R	BC ⇒ R
T	T	T	T	T	T	T	T	T
T	T	F	F	T	T	F	F	T
T	F	T	T	T	F	F	T	T
T	F	F	T	T	F	F	F	T
F	T	T	T	F	T	F	T	T
F	T	F	T	F	T	F	F	T
F	F	T	T	T	F	F	T	T
F	F	F	T	T	F	F	F	T

■ Dificultăți

- Nr tuturor combinațiilor crește exponențial cu n → timp mare de calcul
- Soluția: deducerea prin folosirea regulilor de inferență

Sisteme inteligente – SBC – SBL

Sisteme bazate pe logica propozițională – modulul de control

□ Problemă

- Se dă o BC = $\{P_1, P_2, \dots, P_m\}$ formată din simbolurile $\{X_1, X_2, \dots, X_n\}$ și o propoziție obiectiv O.
- Se poate deduce O din BC?

□ Deducția modelului cu ajutorul regulilor de inferență

■ Etape

- Construirea unei demonstrații a valorii de adevăr a propoziției obiectiv pe baza:
 - propozițiilor
 - originale din BC
 - derivate
 - regulilor de inferență

Regulă de inferență	Premisă	Propoziția derivată
Modus ponens	$A, A \Rightarrow B$	B
Și introductiv	A, B	$A \wedge B$
Și eliminativ	$A \wedge B$	A
Negație dublă	$\neg \neg A$	A
Rezoluție unitară	$A \vee B, \neg B$	A
Rezoluție	$A \vee B, \neg B \vee C$	$A \vee C$

■ Exemplu

□ Problemă

- P = Afară este foarte cald
- Q = Afară este umezeală
- R = Afară plouă
- $BC = \{P \wedge Q \Rightarrow R, Q \Rightarrow P, Q\}$
- $R = \text{Va ploua?}$

Soluție

- | | | |
|----|------------------------------|----------------------|
| 1. | Q | Premisă |
| 2. | $Q \Rightarrow P$ | Premisă |
| 3. | P | Modus Ponens (1,2) |
| 4. | $(P \wedge Q) \Rightarrow R$ | Premisă |
| 5. | $P \wedge Q$ | Și introductiv (1,3) |
| 6. | R | Modus Ponens (4,5) |

Sisteme inteligente – SBC

Tipologia SBC

- **Sisteme bazate pe logică (SBL)**
- **Sisteme bazate pe reguli (SBR)**
- *Case-based reasoning*
- *Hypertext manipulating systems*
- *Data bases and intelligent UI*
- *Intelligent tutoring systems*

Sisteme inteligente – SBC

- Sisteme bazate pe reguli (SBR)
 - Conținut și obiective
 - Proiectare
 - Arhitectură
 - Tool-uri și exemple
 - Avantaje și limite

Sisteme inteligente – SBC – SBR

Conținut și obiective

□ Conținut

- explorează o multitudine de cunoștințe date pentru a obține concluzii noi despre activități dificil de examinat, folosind metode asemănătoare cu experții umani
- pot avea succes la problemele fără soluție algoritmică deterministă
- încearcă să imite un expert uman (într-un anumit domeniu)
- SBR nu înlocuiesc experiența umană, dar îi largesc sfera disponibilității permițând ne-expertilor să lucreze mai bine → **Sisteme expert (SE)**

□ Scopul SBR

- Rezolvarea acelor tipuri de probleme care, de obicei, necesită experți umani prin
 - Transferul expertizei de la un expert la un sistem computațional și
 - Apoi la alți oameni (ne-experti)
- Exemple de probleme rezolvate de SBR → Probleme de recomandare/consultare
 - Consultant medical – aplicație care înlocuiește medicul (dându-se simptomele, SE sugerează un diagnostic și un tratament)
 - Detector al problemelor de funcționare ale unei mașini
 - Detector de probleme în sistemele de operare - Microsoft Windows troubleshooting
 - Consultant financiar

□ De ce se studiază SBR?

- Pentru a înțelege metodele umane de raționare
- Experții umani au nevoie de vacanțe, pot pleca la alte companii, se pot îmbolnăvi, cer măririi de salar, etc.
- Au foarte mare succes comercial

Sisteme inteligente – SBC – SBR

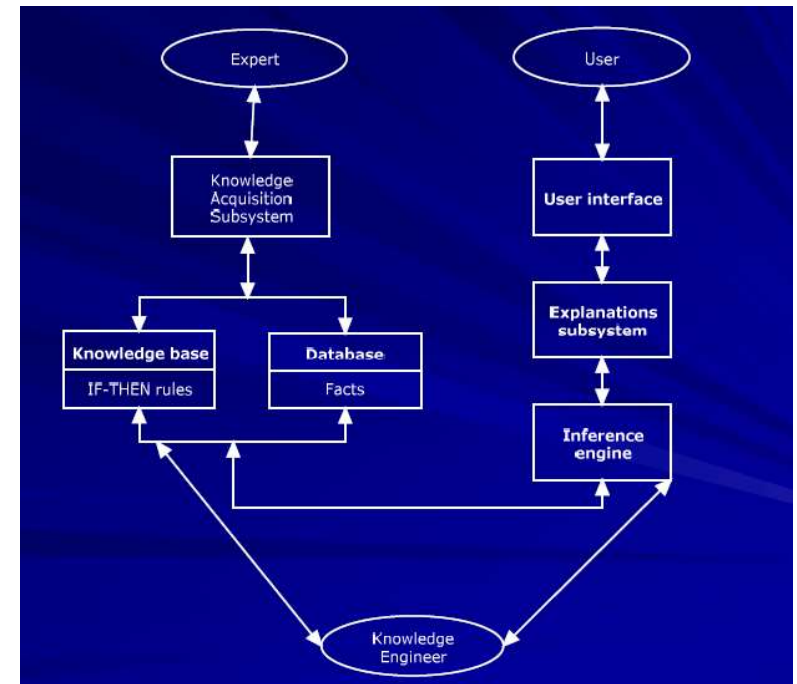
Proiectare

- Achiziționarea informațiilor (cunoștințelor)
- Reprezentarea cunoștințelor
- Inferența cunoștințelor
- Transmiterea către utilizator a cunoștințelor

Sisteme inteligente – SBC – SBR

Arhitectură

- Baza de cunoștințe (BC)
 - Informațiile specifice despre un domeniu
- Modulul de control (MC)
 - Regulile prin care se pot obține informații noi
- Interfața cu utilizatorul
 - permite dialogul cu utilizatorii în timpul sesiunilor de consultare, precum și accesul acestora la faptele și cunoștințele din BC pentru adăugare sau actualizare
- Modulul de îmbogățire a cunoașterii
 - ajută utilizatorul expert să introducă în bază noi cunoștințe într-o formă acceptată de sistem sau să actualizeze baza de cunoștințe.
- Modulul explicativ
 - are rolul de a explica utilizatorilor atât cunoștințele de care dispune sistemul, cât și raționamentele sale pentru obținerea soluțiilor în cadrul sesiunilor de consultare. Explicațiile într-un astfel de sistem, atunci când sunt proiectate corespunzător, îmbunătățesc la rândul lor modul în care utilizatorul percepe și acceptă sistemul



Sisteme inteligente – SBC – SBR

Arhitectură → baza de cunoștințe

□ Conține

- Informațiile specifice despre un domeniu sub forma unor
 - fapte – afirmații corecte
 - reguli - euristici speciale care generează informații (cunoștințe)

□ Rol

- stocarea tuturor elementelor cunoașterii (fapte, reguli, metode de rezolvare, euristici) specifice domeniului de aplicație, preluate de la experții umani sau din alte surse

Sisteme inteligente – SBC – SBR

Arhitectură → baza de cunoștințe

□ Fapte

■ Definiție

- Afirmatii necondiționate corecte (propoziții)
- Memorate în calculator sub forma unor structuri de date

■ Exemplu

- *Ionică lucrează la Facebook*

■ Tipologie

- În funcție de persistență (ritmul de modificare)
 - Fapte statice – aprox. permanente (*Ionică lucrează la Facebook*)
 - Fapte tranzitive – specifice unei instanțe/rulări (*Ionică este în pauza de masă*)
- În funcție de modul de generare
 - Fapte date (*Ionică participă la ședință*)
 - Fapte derivate – rezultate prin aplicarea unor reguli (*Dacă Ionică este PM, atunci el trebuie să conducă ședința*)

Sisteme inteligente – SBC – SBR

Arhitectură → baza de cunoștințe

□ Reguli

■ Definiție

- Euristici speciale care generează informații (cunoștințe)
- O modalitate de exprimare (reprezentare) a cunoștințelor
- Interdependențele între reguli → rețea de inferență
- Fac legătura între cauză și efect - memorate în calculator sub forma unor structuri de control
IF cauză THEN efect
 - Deducție – cauză + regulă → efect
 - Abducție – efect + regulă → cauză
 - Inducție – cauză + efect → regulă

■ Exemplu

- O cauză și mai multe consecințe (combinat cu ȘI)
 - *DACĂ Ionică lucrează la Facebook, ATUNCI el câștigă mulți bani ȘI are puțin timp liber*
- O cauză și mai multe consecințe (combinat cu SAU)
 - *DACĂ anotimpul este iarnă ATUNCI vremea este rece SAU este zăpadă*
- Mai multe cauze/antecedente (combinat cu ȘI) și un efect/o consecință
 - *DACĂ anotimpul este iarnă ȘI temperatura este sub 0 grade ȘI bate vântul ATUNCI nu mergem la plimbare*
- Mai multe cauze/antecedente (combinat cu SAU) și un efect/o consecință
 - *DACĂ anotimpul este iarnă SAU temperatura este sub 0 grade SAU bate vântul ATUNCI vremea este rece*
- Mai multe cauze/antecedente (combinat cu ȘI și SAU) și un efect/o consecință
 - *DACĂ anotimpul este iarnă ȘI temperatura este sub 0 grade SAU bate vântul ATUNCI avioanele nu pot ateriza*

Sisteme inteligente – SBC – SBR

Arhitectură → baza de cunoștințe

□ Reguli

■ Tipologie

□ În funcție de gradul de incertitudine

- Reguli sigure – *Dacă ești angajat, atunci primești salar*
- Reguli nesigure – *Dacă este iarnă, temperatura este sub 0 grade*

□ În funcție de ceea ce exprimă

- Relații – ex. *Dacă studentul are media peste 9.50, atunci el primește bursă*
- Recomandări – ex. *Dacă plouă, atunci să luăm umbrela*
- Directive – ex. *Dacă bateria telefonului este gata, atunci trebuie pusă la încărcat*
- Euristici – ex. *Dacă lumina telefonului este stinsă, atunci bateria este plină*

■ Avantajul lucrului cu reguli

- Ușor de înțeles (o formă naturală a cunoștințelor)
- Simplu de explicat
- Simplu de modificat și întreținut

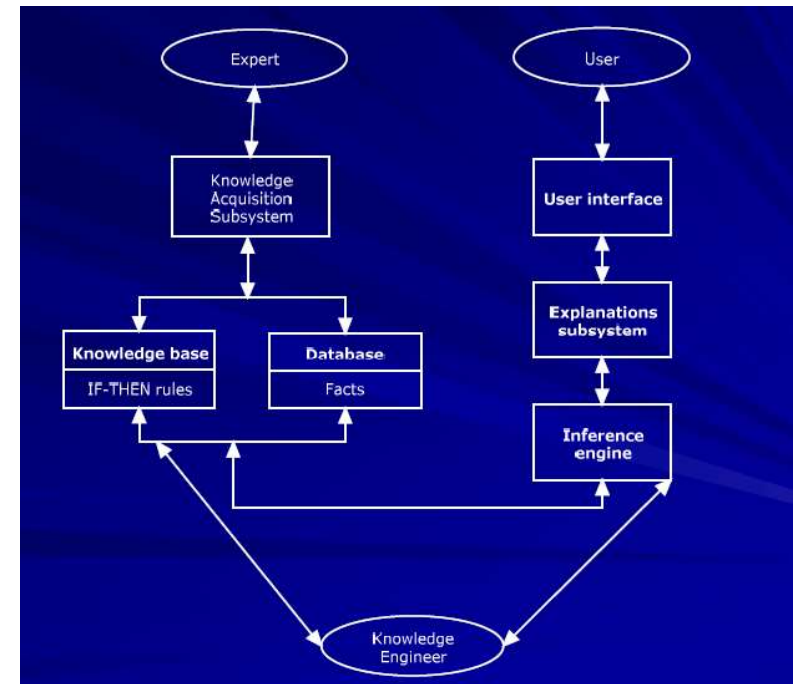
■ Limitări ale regulilor

- Cunoștințele complexe necesită exprimarea prin foarte multe reguli
- Căutarea în sistemele cu numeroase reguli devine greoaie

Sisteme inteligente – SBC – SBR

Arhitectură

- Baza de cunoștințe (BC)
 - Informațiile specifice despre un domeniu
- Modulul de control (MC)
 - Regulile prin care se pot obține informații noi
- Interfața cu utilizatorul
 - permite dialogul cu utilizatorii în timpul sesiunilor de consultare, precum și accesul acestora la faptele și cunoștințele din BC pentru adăugare sau actualizare
- Modulul de îmbogățire a cunoașterii
 - ajută utilizatorul expert să introducă în bază noi cunoștințe într-o formă acceptată de sistem sau să actualizeze baza de cunoștințe.
- Modulul explicativ
 - are rolul de a explica utilizatorilor atât cunoștințele de care dispune sistemul, cât și raționamentele sale pentru obținerea soluțiilor în cadrul sesiunilor de consultare. Explicațiile într-un astfel de sistem, atunci când sunt proiectate corespunzător, îmbunătățesc la rândul lor modul în care utilizatorul percepe și acceptă sistemul



Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control

□ Conținut

- Regulile prin care se pot obține informații noi
- Algoritmi independenți de domeniu
- Creierul SBR – un algoritm de deducere bazat pe BC și specific metodei de raționare
 - un program în care s-a implementat cunoașterea de control, procedurală sau operatorie, cu ajutorul căruia se exploatează baza de cunoștințe pentru efectuarea de raționamente în vederea obținerii de soluții, recomandări sau concluzii.
- depinde de complexitate și tipul cunoștințelor cu care are de-a face

□ Rol

- cu ajutorul lui se exploatează baza de cunoștințe pentru efectuarea de raționamente în vederea obținerii de soluții, recomandări sau concluzii

□ Tipologie – în funcție de direcția inferenței:

- MC cu legătură înainte (*forward chaining*)
- MC cu legătură înapoi (*backward chaining*)

Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control cu legătură înainte (*forward chaining*)

□ Ideea de bază

- Se pornește de la informația disponibilă (fapte date, condiții) și se încearcă ajungerea la o concluzie (fapte derivate) folosind regulile disponibile
- Regulile sunt de forma:
 - partea stângă (PS) => partea dreaptă (PD)
 - partea de condiții => partea de consecințe (efecte)
- Se bazează pe date (*data driven*)

□ Exemplu

- **Întrebare (problemă):** *Angajatul Popescu are telefon?*
- **Regulă:** *Dacă Popescu este angajat, atunci el are telefon.*
- **Fapt curent:** *Popescu este angajat.*
- **Concluzie:** *Popescu are telefon.*

Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control cu legătură înainte
(*forward chaining*)

□ Algoritm

■ Ciclul de execuție

□ Repetă

- Se selectează o regulă a cărei condiții din PS sunt satisfăcute de starea curentă a faptelor stocată în memoria curentă
- Se execută PD a regulii anterior selectate (schimbând starea curentă)

□ Până când nu se mai poate aplica nici o regulă

■ Observații

- Faptele sunt reprezentate în memoria curentă (de lucru) care este continuu actualizată
- Regulile reprezintă acțiuni care pot fi executate atunci când condițiile specificate sunt satisfăcute de elementele stocate în memoria curentă
- Condițiile sunt, de obicei, șabloane care se potrivesc cu elementele din memoria curentă
- Acțiunile implică, de obicei, adăugarea sau eliminarea unor elemente în memoria curentă

Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control cu legătură înapoi
(*backward chaining*)

- Ideea de bază

- Se pornește de la o concluzie potențială (ipoteză) și se caută evidențe care să o suporte/contrazică (explicații)
- Regulile sunt de forma:
 - partea stângă (PS) ==> partea dreaptă (PD)
- Se bazează pe scop (goal driven)

- Exemplu

- **Întrebare (problemă):** *Angajatul Popescu are calculator?*
- **Afirmație:** *Popescu are calculator.*
- **Fapt curent:** *Popescu este programator*
- **Regulă:** *Dacă Popescu este programator, atunci el are calculator*
- Se verifică setul de reguli și se caută ce trebuie să fie Adevărat (în PS) pentru ca Popescu să aibă calculator: un programator. Popescu este programator este un fapt, deci atunci el are calculator.

Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control cu legătură înapoi (*backward chaining*)

□ Algoritm

■ Ciclul de execuție

- Se începe cu starea obiectiv
- Se verifică dacă obiectivul nu se potrivește cu unul din faptele inițiale. Dacă da, atunci STOP. Altfel, se caută acele reguli a căror concluzie se potrivește cu starea obiectiv.
- Se alege una dintre reguli și se încearcă demonstrarea faptelor din precondiție (folosind același mecanism), care devin noi obiective.

■ Observații

- este necesară memorarea obiectivelor urmărite

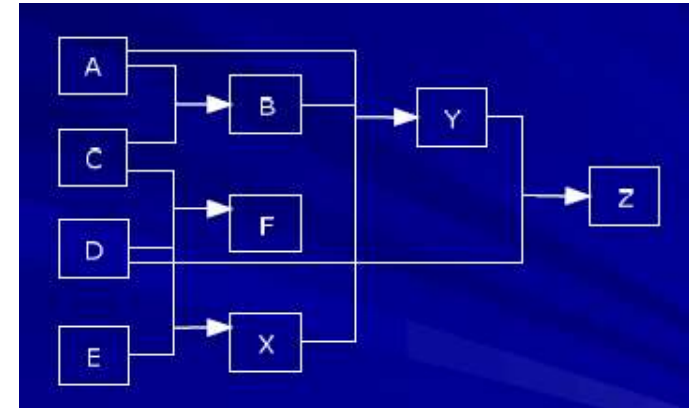
Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

■ Baza de cunoștințe

■ Fapte

- A – secreții nazale
- B – sinuzită
- C – dureri de cap
- D - amețeli
- E – febră
- F – probleme cu tensiunea
- X – infecție
- Y – antibiotic
- Z – repaus la pat



■ Reguli

- R1: *dacă A este adevărat și C este adevărat atunci B este adevărat*
- R2: *dacă C este adevărat și D este adevărat atunci F este adevărat*
- R3: *dacă C este adevărat și D este adevărat și E este adevărat atunci X este adevărat*
- R4: *dacă A este adevărat și B este adevărat și X este adevărat atunci Y este adevărat*
- R5: *dacă D este adevărat și Y este adevărat atunci Z este adevărat*

■ Scop

- faptul Z

Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

□ Algoritm *forward chaining*

■ Se repetă

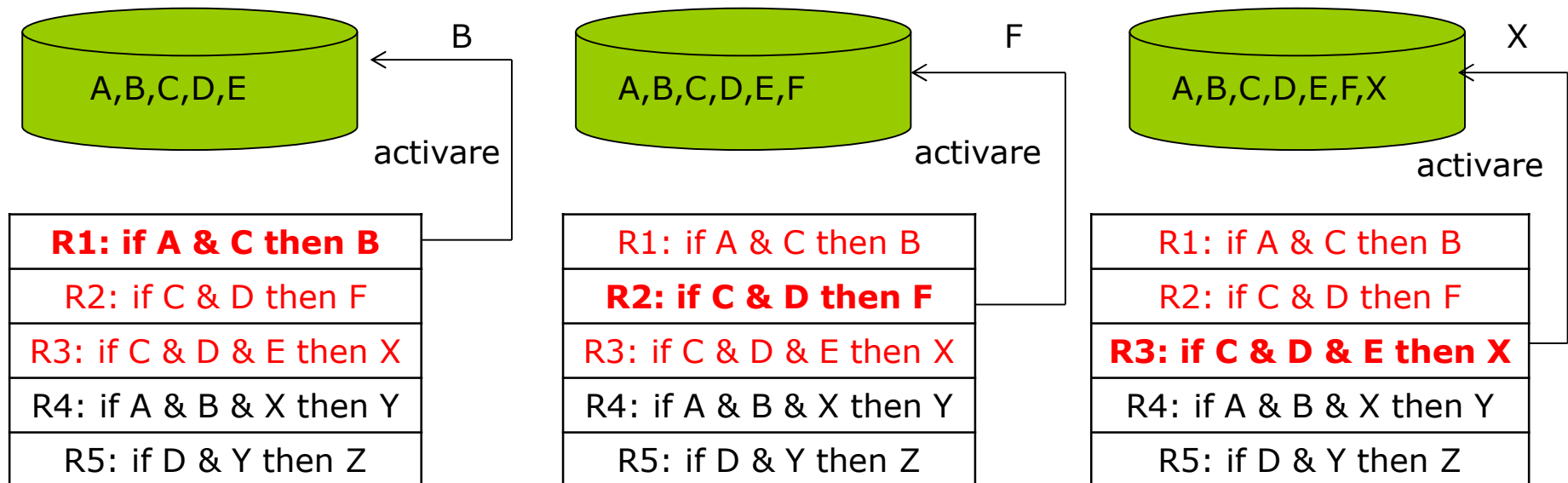
- Se selectează regulile **aplicabile** pentru faptele existente în BC
 - Regulile care conțin în PS a lor doar fapte deja existente în BC
- Dacă pentru un fapt se pot aplica mai multe reguli, se alege doar una dintre ele (care nu a mai fost folosită)
- Se aplică regulile selectate, iar faptele noi obținute se adaugă în BC

■ Până când se ajunge la concluzie sau la o regulă care indică oprirea procesului

Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

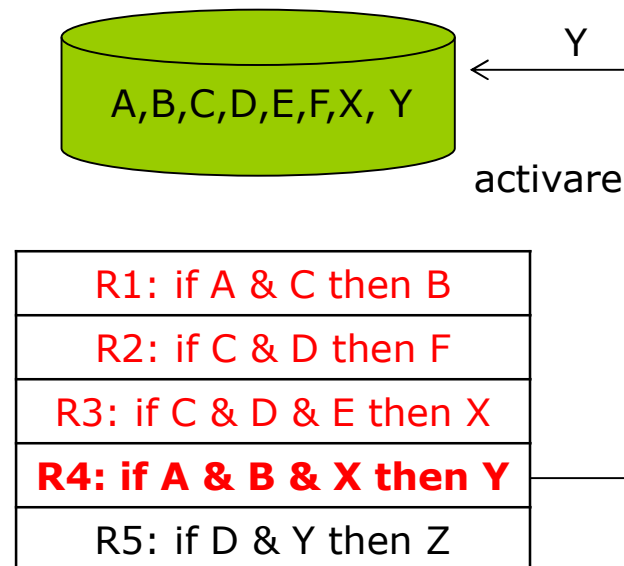
▣ Iterația 1



Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

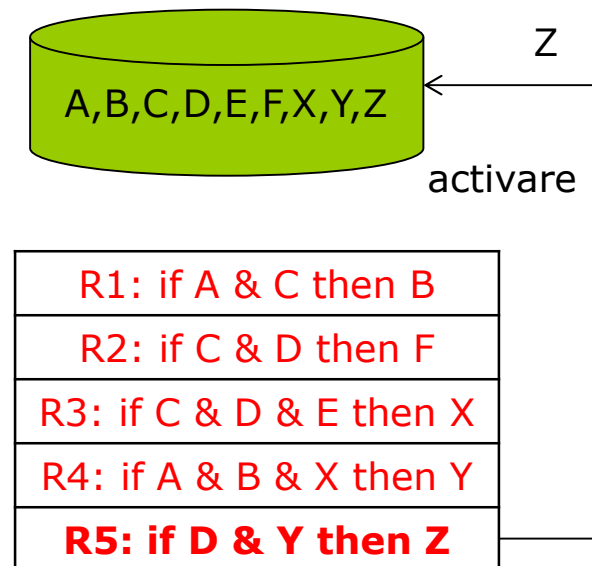
▣ Iterația 2



Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

▣ Iterația 3



Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

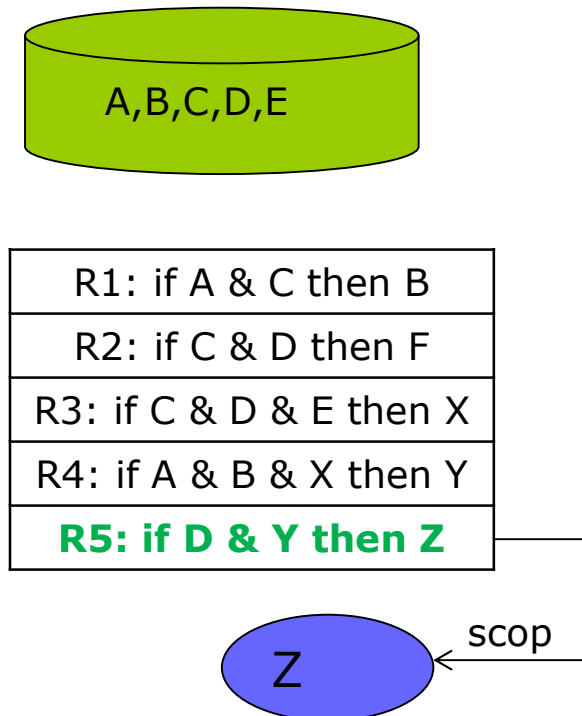
□ Algoritm *backward chaining*

- Se repetă
 - Se selectează regulile **care se potrivesc** cu scopul
 - Regulile care conțin în PD a lor scopul urmărit
 - Dacă pentru un scop se pot aplica mai multe reguli, se alege doar una dintre ele
 - Se **verifică** regulile selectate
 - Se înlocuiește scopul cu premisele (cauzele) regulii selectate, acestea devenind sub-scopuri
- Până când toate sub-scopurile sunt adevărate
 - Sunt fapte cunoscute (existente inițial în BC)
 - sunt informații oferite de utilizator
- Se repetă
 - Se **aplică** regulile anterior verificate în ordine inversă
- Până la aplicarea tuturor regulilor și obținerea scopului urmărit (ca fapt în BC)

Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

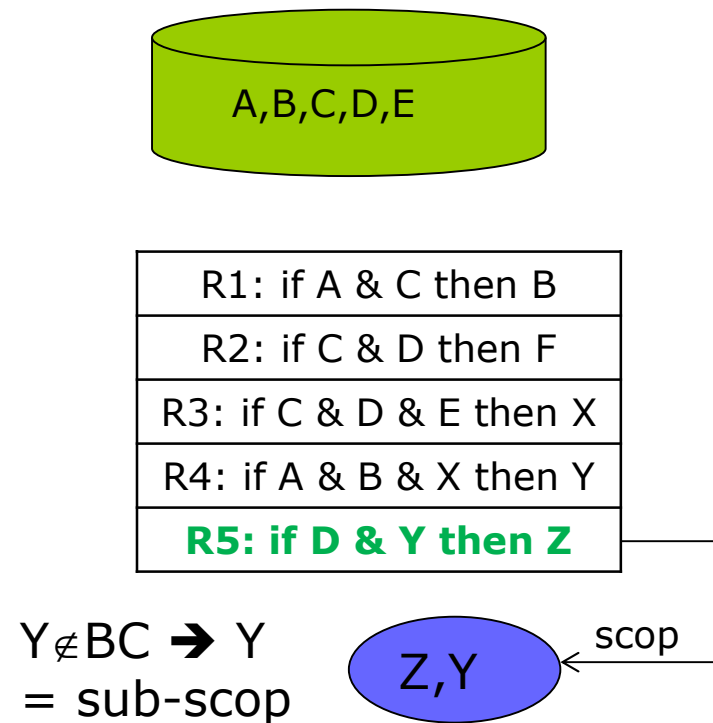
□ Iterația 1.1



Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

□ Iterația 1.1

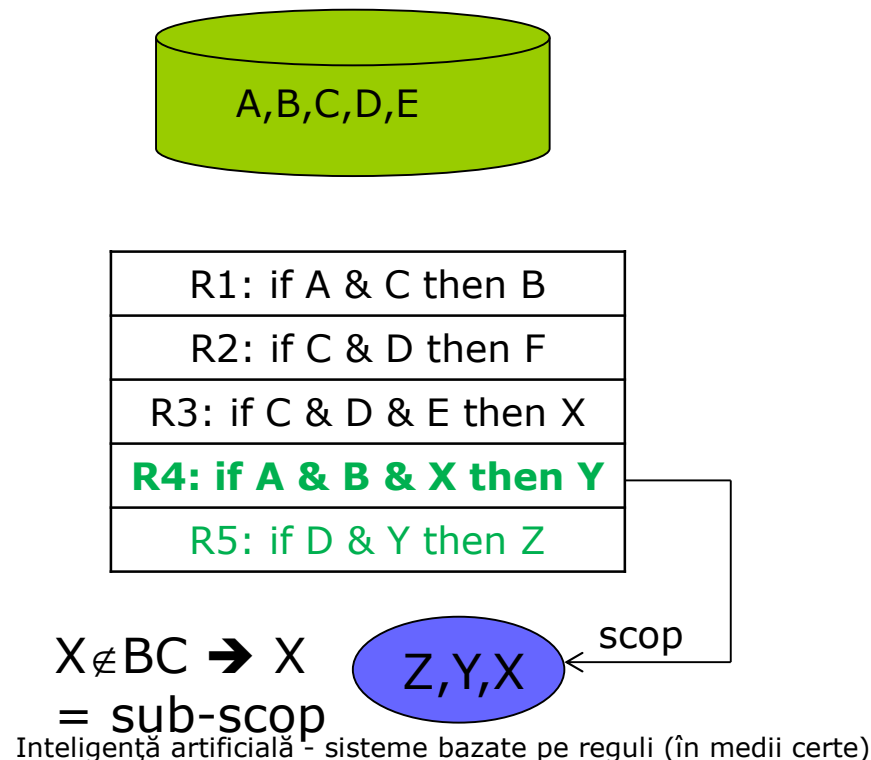


Inteligență artificială - sisteme bazate pe reguli (în medii certe)

Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

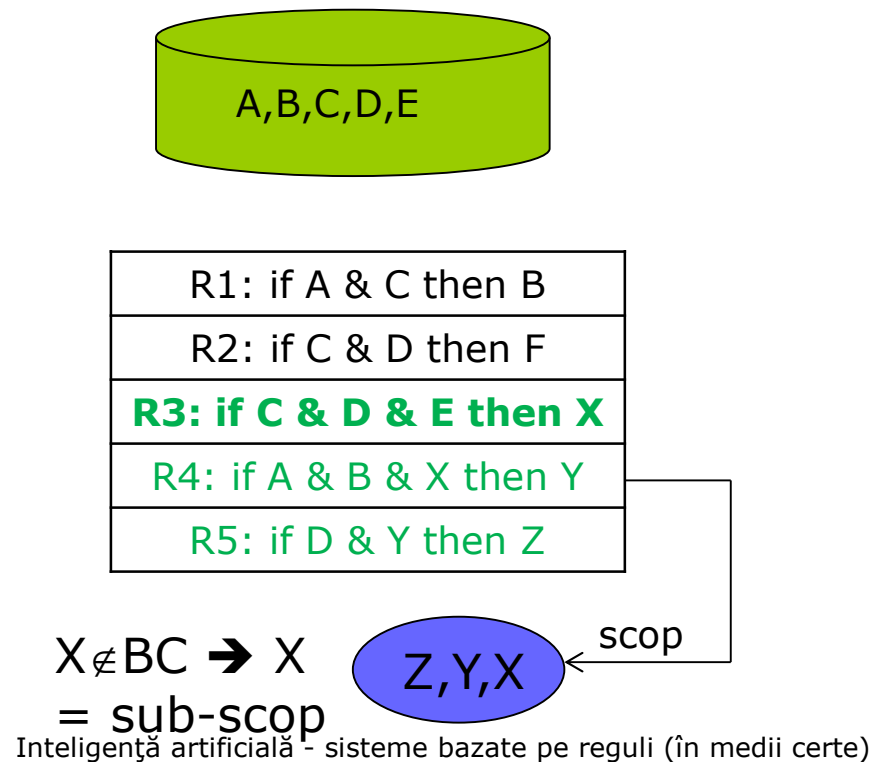
□ Iterația 1.2



Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

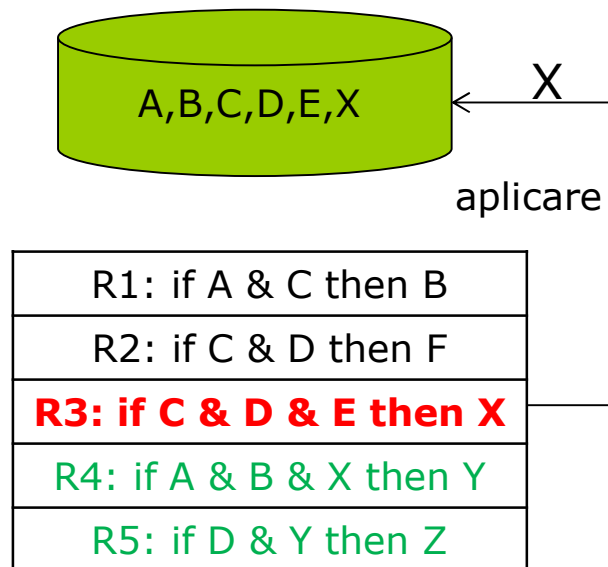
□ Iterația 1.2



Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

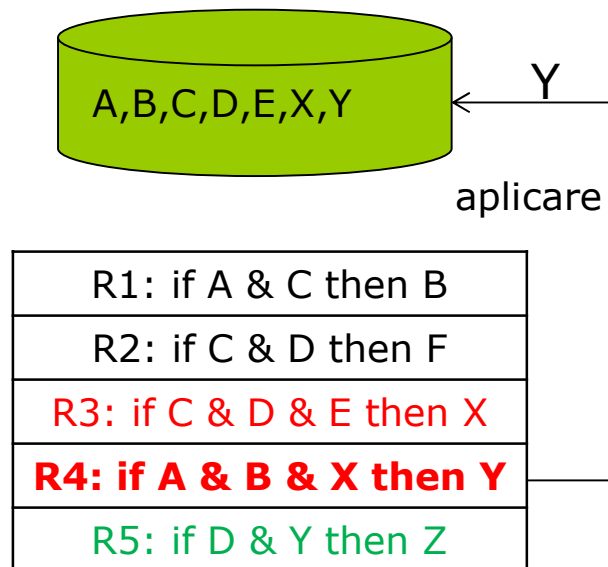
▣ Iterația 2.1



Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

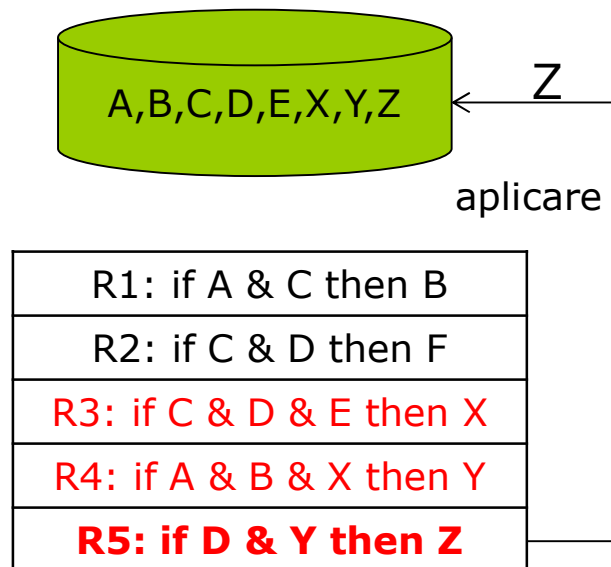
□ Iterația 2.1



Sisteme inteligente – SBC – SBR

Arhitectură → modulul de control – exemplu

▣ Iterația 2.1



Sisteme inteligente – SBC – SBR

Arhitectură – modulul de control

■ Dificultăți

- *Forward Chaining* (FC) sau *Backward chaining* (BC)?
- Rezolvarea conflictelor

Sisteme inteligente – SBC – SBR

Arhitectură – modulul de control – dificultăți

- *Forward Chaining* (FC) sau *Backward chaining* (BC)?
 - FC se recomandă a fi folosit atunci când:
 - Toate (sau aproape toate) informațiile se dau de la început în problemă
 - Există un număr mare de scopuri potențiale, dar doar o parte din ele sunt realizabile pentru o instanță dată a problemei
 - Este dificilă formularea unui scop sau a unor ipoteze
 - BC se recomandă a fi folosit atunci când:
 - Scopul sau ipotezele se dau în problemă sau sunt ușor de formulat
 - Există numeroase reguli care se potrivesc cu faptele din BC, producând numeroase concluzii
 - Datele problemei nu se dau (sau nu sunt ușor accesibile), dar trebuie achiziționate (în anumite sisteme)

Sisteme inteligente – SBC – SBR

Arhitectură – modulul de control – dificultăți

□ Rezolvarea conflictelor

- Dacă se pot aplica mai multe reguli, care regulă este aleasă?
 - De ex.
 - R1: *dacă culoarea este galben atunci fructul este măr*
 - R2: *dacă culoarea este galben și forma este lunguiață atunci fructul este banană*
 - R3: *dacă forma este rotundă atunci fructul este măr*
- Strategii de alegere a regulii care se aplică
 - **prima** regulă
 - o regulă **aleatoare**
 - regula **cea mai specifică**
 - **cea mai veche** regulă
 - **cea mai bună** regulă

Sisteme inteligente – SBC – SBR

Arhitectură – modulul de control – dificultăți

□ Rezolvarea conflictelor

■ Alegerea **primei** reguli care se potrivește (***First in first serve***)

□ Exemplu

- R1: dacă culoarea este galben atunci fructul este măr
- R2: dacă culoarea este galben și forma este lunguiață atunci fructul este banană
- R3: dacă forma este rotundă atunci fructul este măr

□ Observații

- Regulile sunt ordonate doar în sistemele mici

Sisteme inteligente – SBC – SBR

Arhitectură – modulul de control – dificultăți

□ Rezolvarea conflictelor

■ Alegerea **aleatoare** a unei reguli care se potrivește

□ Exemplu

- R1: dacă culoarea este galben atunci fructul este măr
- R2: dacă culoarea este galben și forma este lunguiață atunci fructul este banană
- R3: dacă forma este rotundă atunci fructul este măr

□ Observații

- Alegerea poate fi bună sau mai puțin bună

Sisteme inteligente – SBC – SBR

Arhitectură – modulul de control – dificultăți

□ Rezolvarea conflictelor

■ Alegerea **cele mai specifice** reguli (*Specificity*)

- Cea cu cele mai multe condiții, fiind cea mai relevantă pentru datele existente
- Exemplu
 - R1: dacă culoarea este galben atunci fructul este măr
 - R2: dacă culoarea este galben și forma este lunguiață atunci fructul este banană
 - R3: dacă forma este rotundă atunci fructul este măr
- Observații
 - O regulă specifică procesează mai multă informație decât o regulă generală
→ *longest matching strategy*

Sisteme inteligente – SBC – SBR

Arhitectură – modulul de control – dificultăți

□ Rezolvarea conflictelor

■ Alegerea **cele mai vechi** reguli utilizate (**Recency**):

- Fiecare regulă are asociată o marcă temporală – ultima dată când a fost folosită
- Exemplu
 - R1: dacă culoarea este galben atunci fructul este măr [12.01.2012 – 13.45]
 - R2: dacă culoarea este galben și forma este lunguiață atunci fructul este banană [7.02.2012 – 21.10]
 - R3: dacă forma este rotundă atunci fructul este măr [10.01.2012 – 10.25]
- Observații
 - Noile reguli au fost adăugate de experți mai puțin pregătiți decât vechile reguli (adăugate de experți mai bine pregătiți – cu mai multe cunoștințe în domeniu)

Sisteme inteligente – SBC – SBR

Arhitectură – modulul de control – dificultăți

□ Rezolvarea conflictelor

■ Alegerea **cele mai bune** reguli (***Prioritization***)

- Fiecare regulă are asociată o pondere care specifică importanța ei (relativ la alte reguli)

□ Exemplu

- R1: dacă culoarea este galben atunci fructul este măr [30%]
- R2: dacă culoarea este galben și forma este lunguiață atunci fructul este banană [30%]
- R3: dacă forma este rotundă atunci fructul este măr [40%]

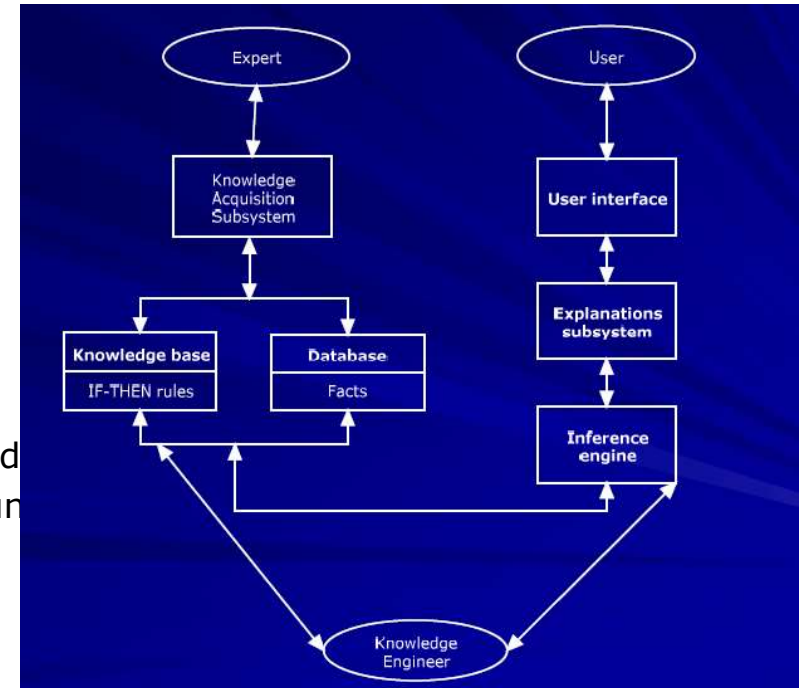
□ Observații

- Necesită expertiză umană pentru stabilirea importanței

Sisteme inteligente – SBC – SBR

Arhitectură

- Baza de cunoștințe (BC)
 - Informațiile specifice despre un domeniu
- Modulul de control (MC)
 - Regulile prin care se pot obține informații noi
- Interfața cu utilizatorul
 - permite dialogul cu utilizatorii în timpul sesiunilor de consultare, precum și accesul acestora la faptele și cunoștințele din bază pentru adăugarea sau actualizarea bazei.
- Modulul de îmbogățire a cunoașterii
 - ajută utilizatorul expert să introducă în bază noi cunoștințe într-o formă acceptată de sistem sau să actualizeze baza de cunoștințe.
- Modulul explicativ
 - are rolul de a explica utilizatorilor atât cunoștințele de care dispune sistemul, cât și raționamentele sale pentru obținerea soluțiilor în cadrul sesiunilor de consultare. Explicațiile într-un astfel de sistem, atunci când sunt proiectate corespunzător, îmbunătățesc la rândul lor modul în care utilizatorul percepe și acceptă sistemul



Sisteme inteligente – SBC – SBR

Arhitectură

- Interfața cu utilizatorul
 - Procesarea limbajului de dialog
 - Tehnici de procesare a limbajului
 - Meniuri
 - Elemente grafice, etc
- Modulul de îmbogățire a cunoașterii
 - ajută utilizatorul expert să introducă în bază noi cunoștințe într-o formă acceptată de sistem sau să actualizeze baza de cunoștințe

Sisteme inteligente – SBC – SBR

Arhitectură

□ Modulul explicativ

- are rolul de a explica utilizatorilor
 - cunoștințele de care dispune sistemul,
 - raționamentele sale pentru obținerea soluțiilor în cadrul sesiunilor de consultare.
- explicațiile într-un astfel de sistem, atunci când sunt proiectate corespunzător, îmbunătățesc la rândul lor modul în care utilizatorul percepe și acceptă sistemul

■ Exemplu

- Un expert medical care prescrie un tratament unui pacient trebuie să explice
 - motivele pentru care a ajuns la acea recomandare
 - riscurile unui astfel de tratament
 - alternative la acest tratament

Sisteme inteligente – SBC – SBR

- Conținut și obiective
- Proiectare
- Arhitectură
- Tool-uri și exemple
- Avantaje și limite

Sisteme inteligente – SBC – SBR

Tool-uri si limbaje existente

- PROLOG
 - Limbaj de programare care utilizează *backward chaining*
- ART (Inference Corporation)
 - În 1984, Inference Corporation a dezvoltat Automated Reasoning Tool (ART), un sistem expert bazat pe *forward chaining*
- CLIPS
 - NASA preia abilitățile de *forward chaining* ale sintaxei ART și dezvoltă *C Language Integrated Production System*" (CLIPS)
- ART-IM (Inference Corporation)
 - Inference Corporation implementează o versiunea *forward-chaining* a ART/CLIPS, numită ART-IM.
- OPS5 (Carnegie Mellon University)
 - Primul limbaj de IA utilizat pentru sisteme de producție (XCON)
- Eclipse (The Haley Enterprise, Inc.)
 - Eclipse este singurul modul de control pentru C/C++ care suportă atât *forward chaining*, cât și *backward chaining*

Sisteme inteligente – SBC – SBR

Exemple

- ❑ DENDRAL (1965-1983)
 - Analizează structura moleculelor și propune structuri plauzibile pentru compuși chimici noi sau necunoscuți
- ❑ MYCIN (1972-1980)
 - Program interactiv pentru
 - ❑ diagnosticarea unor boli infecțioase sangvine
 - ❑ Recomandări terapeutice antimicrobiene
- ❑ EMYCIN, HEADMED, CASNET și INTERNIST
 - pentru domeniul medical
- ❑ PROSPECTOR (1974-1983)
 - Oferă recomandări pentru explorările mineralelor
- ❑ TEIRESIAS
 - pentru achiziția inteligentă a cunoașterii
- ❑ XCON (1978-1999)
 - Oferă recomandări pentru configurarea calculatoarelor
- ❑ SBR financiare
 - ExpertTAX, Risk Advisor (Coopers & Lybrand), Loan Probe, Peat/1040 (KPMG), VATIA, Flow Eval (Ernst & Young), Planet, Compas, Comet (Price Waterhouse), Rice (Arthur Andersen), Audit Planning Advisor, World Tax Planner (Deloitte Touche)

Sisteme inteligente – SBC – SBR

Avantaje și limite

□ Avantaje

- Oferă recomandări celor mai puțin experți în anumite domenii
- Permit companiilor replicarea celor mai buni angajați
 - Preiua informația și cunoștințele intelectuale ale experților și le pun la dispoziția celorlalți oameni
- Se reduc erorile datorate proceselor de automatizare a sarcinilor monotone, repetitive sau critice
- Se reduce necesarul de forță umană și de timp pentru testarea și analizarea datelor
- Se reduc costurile prin accelerarea procesului de observare a greșelilor
- Se elimină munca pe care oamenii nu ar trebui să o facă (dificilă, consumatoare de timp, susceptibilă de erori, care necesită antrenare lungă și costisitoare)
- Se elimină munca pe care oamenii nu și-o doresc să o facă (luarea unor decizii care nu-i pot mulțumi pe toți – sistemele expert nu pot fi acuzate de favoritsime)

□ Dezavantaje

- Domeniu îngust de aplicare a unui SBR
- Focus limitat la anumite obiective
- Lipsa capacității de învățare și adaptare
- Probleme de întreținere
- Costuri de dezvoltare mari



Recapitulare

□ SBC

- Sisteme computaționale în care
- baza de cunoștințe și modulul de control se suprapun

□ SBC pot fi

■ SBL

- explorează o multitudine de cunoștințe date pentru a obține concluzii noi despre activități dificil de examinat, folosind metode specifice logicii formale
- Componentă
 - limbaj (sintaxă + semantică) și
 - metoda de deducție (inferență)

■ SBR

- explorează o multitudine de cunoștințe date pentru a obține concluzii noi despre activități dificil de examinat, folosind metode asemănătoare cu experții umani
- pot avea succes la problemele fără soluție algoritmică deterministică
- încearcă să imite un expert uman (într-un anumit domeniu)
- Componentă
 - Baza de cunoștințe → fapte și reguli
 - Modulul de control → inferență înainte sau înapoi

Cursul următor

A. Scurtă introducere în Inteligența Artificială (IA)

B. Rezolvarea problemelor prin căutare

- Definirea problemelor de căutare
- Strategii de căutare
 - Strategii de căutare neinformate
 - Strategii de căutare informate
 - Strategii de căutare locale (Hill Climbing, Simulated Annealing, Tabu Search, Algoritmi evolutivi, PSO, ACO)
 - Strategii de căutare adversială

C. Sisteme inteligente

- Sisteme care învață singure
 - Arbori de decizie
 - Rețele neuronale artificiale
 - Mașini cu suport vectorial
 - Algoritmi evolutivi
- Sisteme hibride
- Sisteme bazate pe reguli în medii certe
- Sisteme bazate pe reguli în medii incerte (Bayes, factori de certitudine, Fuzzy)

Cursul următor – Materiale de citit și legături utile

- ❑ Capitolul V din *S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, 1995*
- ❑ capitolul 3 din *Adrian A. Hopgood, Intelligent Systems for Engineers and Scientists, CRC Press, 2001*
- ❑ capitolul 8 și 9 din *C. Groșan, A. Abraham, Intelligent Systems: A Modern Approach, Springer, 2011*

□ Informațiile prezentate au fost colectate din diferite surse de pe internet, precum și din cursurile de inteligență artificială ținute în anii anteriori de către:

■ Conf. Dr. Mihai Oltean –
www.cs.ubbcluj.ro/~moltean

■ Lect. Dr. Crina Groșan -
www.cs.ubbcluj.ro/~cgrosan

■ Prof. Dr. Horia F. Pop -
www.cs.ubbcluj.ro/~hfpop