



Inteligența Artificială

Universitatea Politehnica București
Anul universitar 2013-2014

Adina Magda Florea



Curs nr. 5

Sisteme bazate pe reguli

- Reprezentarea prin reguli
- Structura SBR
- Ciclul de inferenta al unui sistem bazat pe reguli
- Strategia de control
- SBR cu inlantuire inainte
- SBR cu inlantuire inapoi
- Sisteme bazate pe agenda



1. Reprezentarea prin reguli

- Reprezentare modulara a cunostintelor procedurale
- Cunostinte procedurale in maniera declarativa
- Permit atasare de actiuni (functii, proceduri)
- Se pot combina cu reprezentari structurate ale cunostintelor, de ex. ontologii
- Multiple limbaje bazate pe reguli



Limbaje bazate pe reguli

- OPS5 (Official Production System) – precursor
- MYCIN – precursor
- CLIPS - C Language Integrated Production System
- Jess – written in Java, superset of CLIPS
- RuleML – Rule Markup Language (Rule Markup Initiative)
- LegalXML, Drools

RuleML 2013

The 7th International Web Rule Symposium:
Research Based and Industry Focused

Seattle Metropolitan Area, USA, July 11-13, 2013.



The Symposium

- [About RuleML 2013](#)
- [Call for Papers](#)
- [Important Dates](#)
- [Organization Committee](#)
- [Doctoral Consortium](#)
- [Rule Challenge](#)
- [Workshops](#)
- [Keynotes and Tutorials](#)
- [Program and Accepted Papers](#)
- [Venue and Accommodation](#)
- [Registration](#)
- [Sponsorship](#)
- [SymposiumPlanner-2013 Q&A](#)

Ruleml 2013 on Twitter

#ruleml2013



Monica Palmirani

13 Jul

@MonicaPalmirani

#ruleml2013 Now available CEUR proceedings of the Rule Challenge, Human Language Technology

About RuleML 2013

The 7th International Web Rule Symposium (RuleML 2013) will be held at the University of Washington, Seattle in the U.S. state of Washington, on July 11-13th, 2013 co-located directly before the AAI 2013 conference.

RuleML 2013 also hosts:

- Human Language Technology - Special Track on "Translating between Human Language and Formal Rules: Business, Law, and Government"
- OASIS Legal RuleML Tutorial and Meeting
- 3rd RuleML Doctoral Consortium
- 7th International Rule Challenge
- ISO Common Logic Meeting

RuleML 2013 Springer LNCS Proceedings

Rules Challenge, Doctoral Consortium, Human Language Technology CEUR Proceedings

[Read more](#)



Sponsors



Partners





Modelul

Temperatura(p,v)

- R1: **daca** pacientul are temperatura mare
si tipul organismului este gram-pozitiv Tip(o,f)
si pacientul are gitul uscat GitUscat(p)
atunci organismul este streptococ Identitate(o,i)
- R2: **daca** masina nu porneste
si farurile nu se aprind
atunci bateria este consumata
sau bornele bateriei nu fac contact
- R3: **daca** temperatura $> 95^{\circ} \text{C}$
atunci deschide valva de protectie



Reprezentarea prin reguli - cont

R1

$$(\forall p)(\forall o)(\text{Temperatura}(p, \text{mare}) \wedge \text{Tip}(o, \text{gram - pozitiv}) \wedge \text{GitUscat}(p)) \rightarrow$$

Identitate(o, streptococ)

R2

$$(\forall m)(\exists b)(\text{NuPorneste}(m) \wedge \text{NuAprinde}(m, \text{far})) \rightarrow$$
$$(\text{Stare}(b, \text{consumata}) \vee \text{Borne}(b, \text{fara_contact}))$$

R3: **daca** temperatura > 95° C
 atunci deschide valva de protectie

■ R3? Reprezentare actiuni in LP ?



Forme de reguli

■ Reguli de inferenta

- **if** conditie(i) **then** concluzie/actiune

■ Utilizare

- obtinerea de noi cunostinte
- executarea actiunilor in functie de conditii

■ Reguli reactive

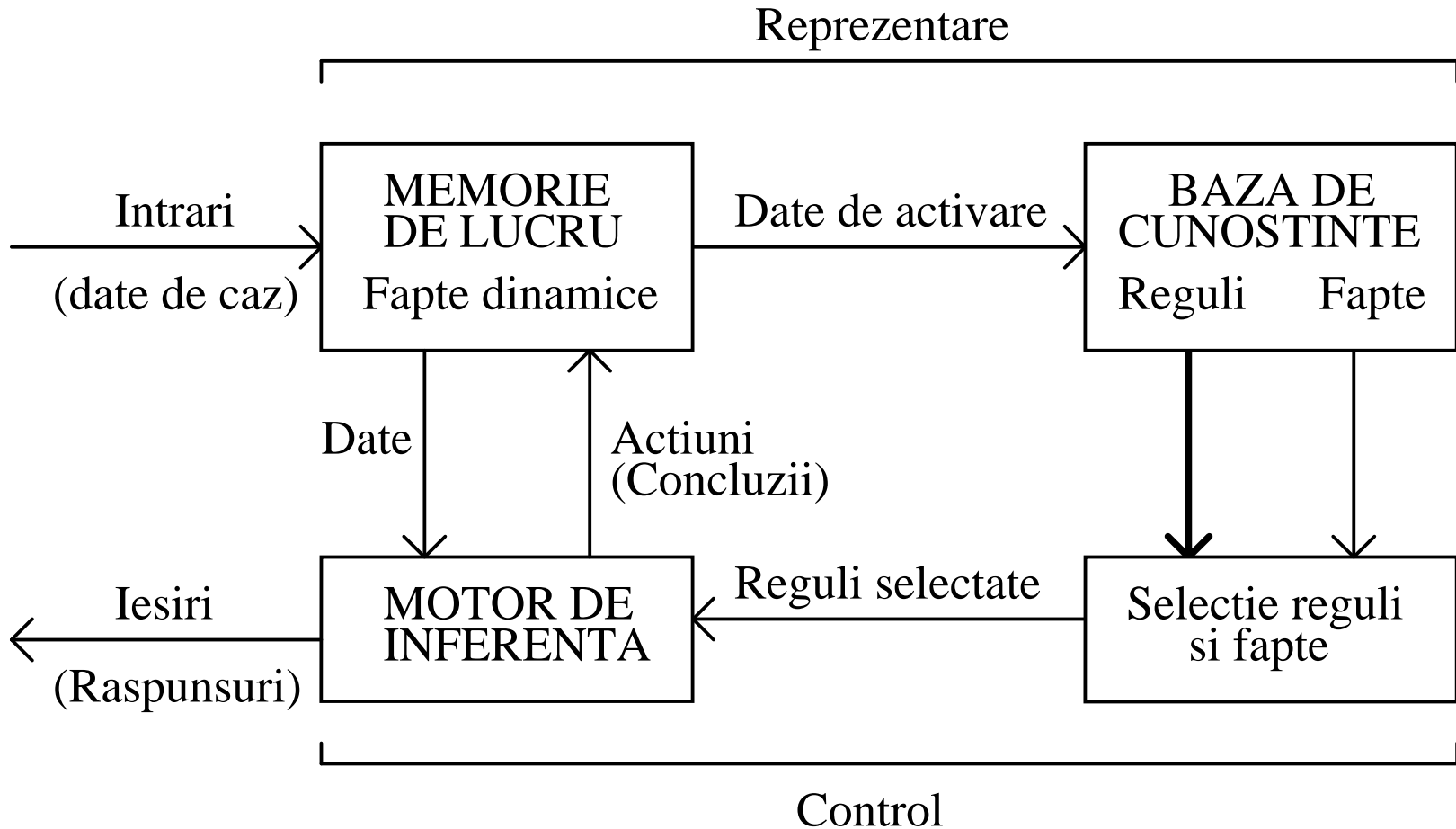
- **on** eveniment **if** conditie(i) **then** (executa) actiune

■ Utilizare

- triggere in SMBD
- detectarea exceptiilor
- reguli de integritate

■ Se pot combina

2. Structura SBR





3. Ciclul de inferenta al unui sistem bazat pe reguli

- Identificare
- Selectie
- Executie



Ciclul de inferenta al unui sistem bazat pe reguli - cont

Algorithm: Functionarea unui sistem bazat pe reguli

1. $ML \leftarrow$ Date de caz

2. **repetă**

2.1. Execută identificare între ML și BC (partea stînga sau partea dreaptă a regulilor și fapte) și creează mulțimea de conflicte a regulilor aplicabile (MC)

2.2. Selectează o regulă după un criteriu de selecție

2.3. Aplică regulă prin executia părții drepte a regulii

pană nu mai sunt reguli aplicabile **sau**

memoria de lucru satisface condiția de stare scop **sau**

o cantitate predefinită de efort a fost epuizată

sfârșit.



4. Strategia de control

- Criteriile de selectie din MC
- Directia de aplicare a regulilor



4.1 Criteriile de selectie din MC

- Selectia primei reguli aplicabile
- Alegerea unei reguli din multimea de conflicte
 - Preferinte bazate pe natura regulilor
 - Specificitate*
 - Momentului folosirii anterioare*
 - Preferinte bazate pe obiectele identificate
 - Preferinte bazate pe natura starilor

daca

regula refera {nu refera} X

$$\{$$

de loc/

numai in partea stinga/

numai in partea dreapta }

atunci

regula va fi in special utila

$$\{$$

probabil utila/

probabil inutila/

sigur inutila

}



4.2 Directia de aplicare a regulilor

INLANTUIRE INAINTE INLANTUIRE INAPOI

daca A atunci B
daca B atunci C
A (data)
C (concluzie)

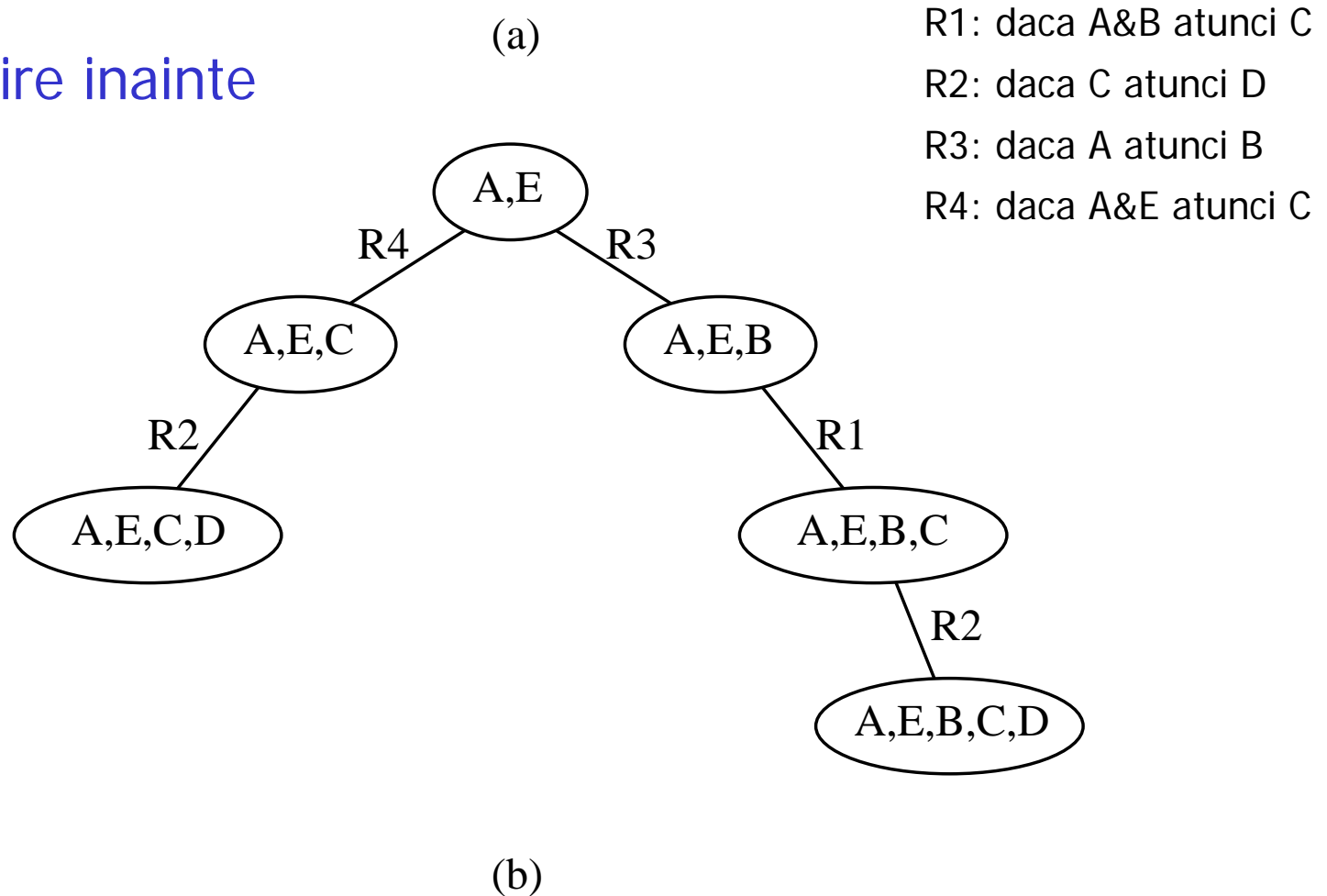
determina C
daca B atunci C
daca A atunci B

(daca A atunci C, implicit)
Este A adevarata? (data)

Continut initial al memoriei de lucru: A,E
Stare scop: D

$A,E \xrightarrow{R3} A,E,B \xrightarrow{R1} A,E,B,C \xrightarrow{R2} A,E,B,C,D$

(a) Inlantuire inainte

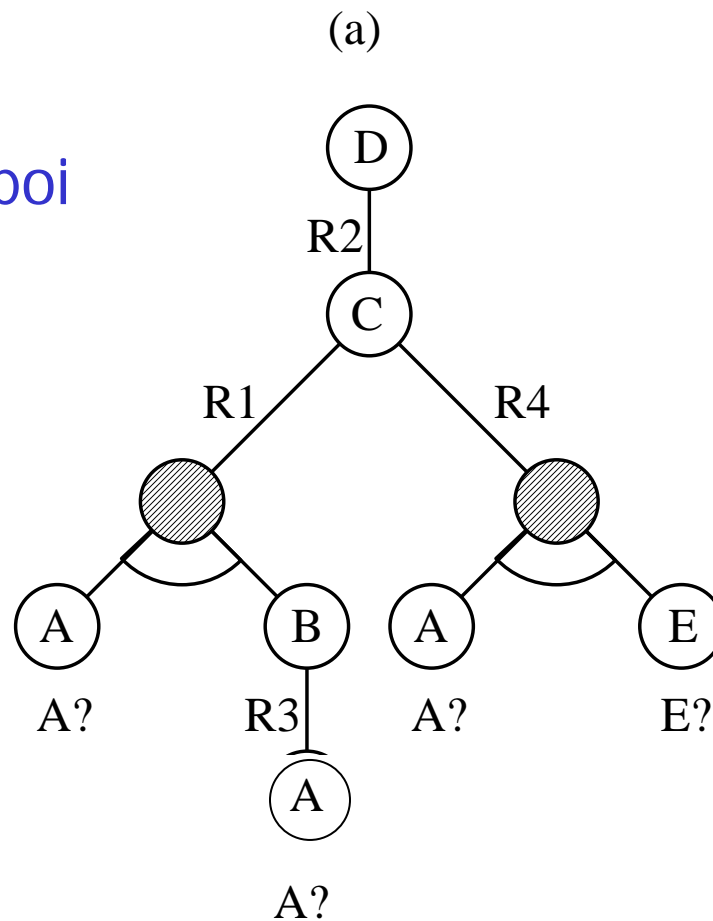


Continut initial al memoriei de lucru: A,E

Stare scop: D

$D? \xrightarrow{R2} C? \xrightarrow{R1} A?, B? \xrightarrow{R3} A?, E?$

(b) Inlantuire inapoi



R1: daca A&B atunci C

R2: daca C atunci D

R3: daca A atunci B

R4: daca A&E atunci C

Algoritm: Determinarea unei valori cu inlantuire inainte a regulilor (a)

DetValoareD(A)

1. **daca** A are valoare
atunci intoarce SUCCES
2. Construiește multimea de conflicte, MC, și initializează Efort
3. **daca** $MC = \{ \}$
atunci intoarce INSUCCES
4. Alege o regula $R \in MC$ după un criteriu de selecție
5. Aplica regula R
 Adauga faptul din concluzia regulii R la ML (daca nu exista deja)
 sau executa actiunea din concluzie
6. **daca** A are valoare ($A \in ML$)
atunci intoarce SUCCES
7. $MC \leftarrow MC - R$
8. $MC \leftarrow MC \cup \text{Noile reguli aplicabile}$
9. Actualizeaza Efort
10. Daca $Efort = \{ \}$ atunci **intoarce** INSUCCES
11. **repeta** de la 3
sfarsit.

Algoritm: **Determinarea unei valori cu inlantuirea inapoi a regulilor (b)**

DetValoare(A)

1. Construiește multimea regulilor care refera A în concluzie, MC
2. **daca** MC = { }
 atunci
 - 2.1. Intreaba utilizatorul valoarea lui A
 - 2.2. **daca** se cunoaste valoarea lui A
 atunci depune în ML aceasta valoare
 altfel intoarce INSUCCES
3. **altfel**
 - 3.1. Alege o regula după un criteriu de selecție
 - 3.2. **pentru** fiecare ipoteza I_j , $j=1,N$, a premisei lui R **executa**
 - 3.2.1. Fie A_j atributul referit de ipoteza I_j
 - 3.2.2. **daca** A_j are valoare
 atunci atunci evalueaza adevarul ipotezei I_j
 - 3.2.3. **altfel**
 - i. **daca** DetValoare(A_j) = SUCCES
 atunci evalueaza adevarul ipotezei I_j
 - ii. **altfel** considera ipoteza I_j nesatisfacuta

3.3. **daca** toate ipotezele $I_j, j=1,N$ sunt satisfacute

atunci depune in ML valoarea lui A dedusa pe baza concluziei R

4. **daca** A are valoare (in ML)

atunci intoarce SUCCES

5. **altfel**

5.1 $MC \leftarrow MC - R$

5.2 **repeta** de la 2

sfarsit.

1'. **daca** A este data primara

atunci

1'.1. Intreaba utilizatorul valoarea lui A

1'.2. **daca** se cunoaste valoarea lui A

atunci - depune in ML valoarea lui A

- **intoarce** SUCCES

1'.3. **altfel intoarce** INSUCCES



5. SBR cu inlantuire inainte

Familia OPS5

- OPS5
- ART
- CLIPS
- Jess
- Familia Web Rule languages
 - In special RuleML si SWRL (OWL + Ruel language)
 - Interoperabilitatea regulilor

Baze de date

- Datalog – subset al Prolog pt interogarea bazelor de date



SBR cu inlantuire inainte - eficienta

Cresterea eficientei implementarii

- Partitionarea regulilor in functie de o conditie – OPS 5
- Elimin necesitatea sa verific la fiecare ciclu care reguli sunt aplicabile

Exemplu SBR tip OPS5 cu inlantuire inainte

Nume	Greutate	Dimensiune	Fragil	Impachetat
Statuie	mediu	mare	nu	nu
Tablou	usor	medie	nu	nu
Vaza	usor	mica	da	nu
Soclu	greu	mare	nu	nu

Etapă : Verifica - Decor

Cutie 1 : { }

Obiecte– Neimpachetate: Statuie, Tablou, Vaza, Soclu

R11: **daca** Etapa este Verifica-Decor

si exista o statuie

si nu exista soclu

atunci adauga soclu la Obiecte-Neimpachetate

R17: **daca** Etapa este Verifica-Decor

atunci termina Etapa Verifica-Decor

si incepe Etapa Obiecte-Mari

Exemplu SBR cu inlantuire inainte - cont

R21: **daca** Etapa este Obiecte-Mari

si exista un obiect mare de ambalat

si exista un obiect greu de ambalat

si exista o cutie cu mai putin de trei obiecte mari

atunci pune obiectul greu in cutie

R22: **daca** Etapa este Obiecte-Mari

si exista un obiect mare de ambalat

si exista o cutie cu mai putin de trei obiecte mari

atunci pune obiectul mare in cutie

R28: **daca** Etapa este Obiecte-Mari

si exista un obiect mare de pus

atunci foloseste o cutie noua

R29: **daca** Etapa este Obiecte-Mari

atunci termina Etapa Obiecte-Mari

si incepe Etapa Obiecte-Medii

Etapa :

Cutie 1 :

Obiecte– Neimpachetate:

Obiecte - Medii

Soclu, Statuie

Tablou, Vaza



SBR cu inlantuire inainte - eficienta

- Verifica la fiecare ciclu care reguli sunt aplicabile

Forward chaining incremental

- Algoritmul RETE
- **Match** – cam 80% din timpul ciclului recunoastere-actiune
- Fiecare WME este comparat cu fiecare conditie din fiecare regula
 - $O(R * F^P)$, R – nr de reguli, F – nr WME, P nr mediu de conditii in LHS regula



SBR cu inlantuire inainte - eficienta

- RETE match algorithm (1982 Forgy)
 - Salveaza starea de match intre 2 cicluri recunoastere-actiune
 - La crearea unui WME, acesta este comparat cu toate elementele conditie din program si este memorat impreuna cu fiecare element conditie cu care a identificat =>
 - Numai schimbarile incrementale din WM sunt identificate in fiecare ciclu
 - $O(R * F * P)$ aprox

Exemplu JESS

```
(deftemplate student
```

```
  (slot name)
```

```
  (slot sex)
```

```
  (slot placed_in)
```

```
  (slot special_considerations (default no)))
```

```
(deftemplate room
```

```
  (slot number)
```

```
  (slot capacity (type INTEGER)(default 4))
```

```
  (slot sexes_are)
```

```
  (slot vacancies (type INTEGER))
```

```
  (multislot occupants))
```

Exemplu JESS

```
(assert (room (number 112) (capacity 4) (vacancies 4)
              (sexes_are nil) (occupants nil)))
(assert (student (name Mihai) (placed_in nil) (sex M)))
```

WMEs

```
(room 112 4 4 nil nil)
(student Mihai nil M)
```

```
time tag1
time tag 2
```

Stud/cam in Jess

```
(defrule assign-student-empty-room
```

```
  ?unplaced_student <- (student (name ?stud_name)
                                (placed_in nil)
                                (sex ?gender))
```

```
  ?empty_room <- (room (number ?room_no)
                      (capacity ?room_cap)
                      (vacancies ?max_size))
```

```
  (test (= ?room_cap ?max_size))
```

```
=>
```

```
  (modify ?unplaced_student (placed_in ?room_no))
```

```
  (modify ?empty_room (occupants ?stud_name) (sexes_are ?gender)
            (vacancies (-- ?max_size))
```

```
)
```



Strategii SBR familia OPS5

- Ciclul recunoastere-actiune:
 - Match
 - Select
 - Act
- **WME** = working memory element
 - Identificat printr-un "time tag"
- Instantiere: multime de WME care satisface o regula
- Multime de conflicte (MC)
- Rezolvarea conflictelor

Multimea de conflicte

(**defrule** **assign-student-empty-room**

(student (name ?stud_name)

(placed_in nil)

(sex ?gender))

(room (number ?room_no)

(capacity ?room_cap)

(vacancies ?max_size))

(test (= ?room_cap ?max_size))

=> ...

Instantieri (MC)

assign-student-empty-room 41 9

assign-student-empty-room 41 17

assign-student-empty-room 52 9

assign-student-empty-room 52 17

WMEs

41 (student name Mary sex F placed_in nil)

52 (student name Peter sex M placed_in nil)

9 (room number 221 capacity 1 vacancies 1)

12 (room number 346 capacity 2 vacancies 1)

17 (room number 761 capacity 1 vacancies 1)



Rezolvarea conflictelor

■ Diferite strategii

- **Refractie** = o aceeași instanțiere nu este executată de mai multe ori (2 instanțieri sunt la fel dacă au același nume de regulă și aceleași time tags, în aceeași ordine)
- **Momentul utilizării** = Se preferă instanțierile care au identificat cu WMEs cu cele mai recente time tags (sau invers)
- **Specificitate** = Au prioritate instanțierile cu LHS specifice = nr de valori testate în LHS
 - teste asupra: nume clasă, predicat cu 1 arg constantă, predicat cu un operator variabilă legată



Rezolvarea conflictelor

- **Strategia LEX**

- Elimina din MC instantierile care au fost deja executate
- Ordoneaza instantierile pe baza momentului utilizarii
- Daca mai multe instantieri au aceleasi time tags, utilizeaza specificitate
- Daca mai multe instantieri au aceeasi specificitate alege arbitrar

- **Strategia MEA** – aceeasi dar utilizeaza primul time tag



Construirea MC

Match

- O regula se poate aplica daca conditiile din antecedent sunt satisfacute de WMEs din ML
- Procesul are 2 aspecte:
 - match intra-element
 - match inter-element

```
(defrule assign-student-room
  (student (name ?stud_name) (placed_in nil)
           (sex ?gender))
  (room (number ?room_no) (sexes_are ?gender)
        (vacancies ?size))
  (test (> ?size 0))
```

=> ...

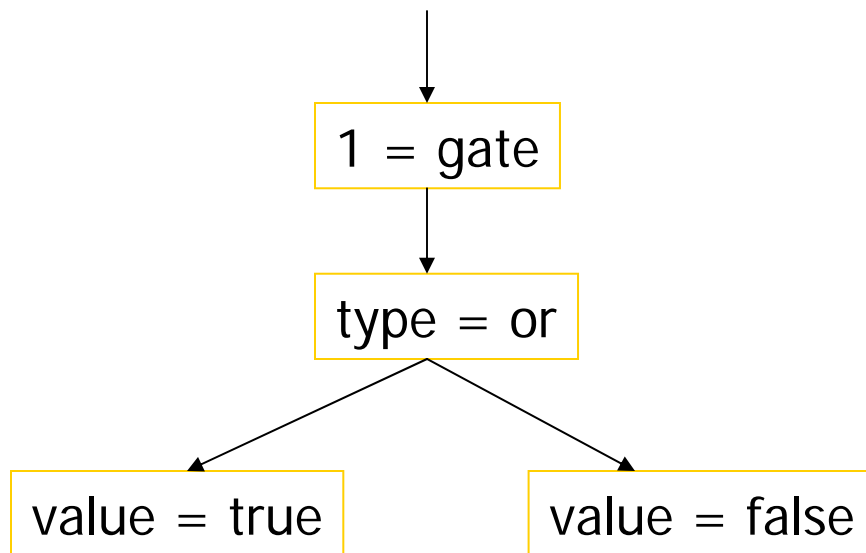
RETE - Compilarea regulilor

- Se compileaza conditiile regulilor intr-o **retea de noduri de test** – retea de flux de date
- Exista noduri pt teste **intra-element** (**Alpha nodes**) si **inter-element** (**Beta nodes**)
- Intrarea – schimbari in ML
- Iesirea – schimbari in MC

- **Partajarea nodurilor**

(gate (type or) (value true))

(gate (type or) (value false))





RETE - Compilarea regulilor

- Un pointer la noul WME este trecut prin retea, incepand de la nodul radacina
- Fiecare nod actioneaza ca un switch
- Cand un nod primeste un pointer la un WME, testeaza WME asociat. Daca testul reuseste, nodul se deschide si WME trece mai departe
- Altfel nu se intampla nimic
- Daca un WME va trece prin retea, va fi combinat cu alte WME care trec pentru a forma o instantiere in MC
- Pointerii la WME sunt trimisi prin retea ca tokens = pointer + stare (assert sau retract)

(defrule cat-job-size

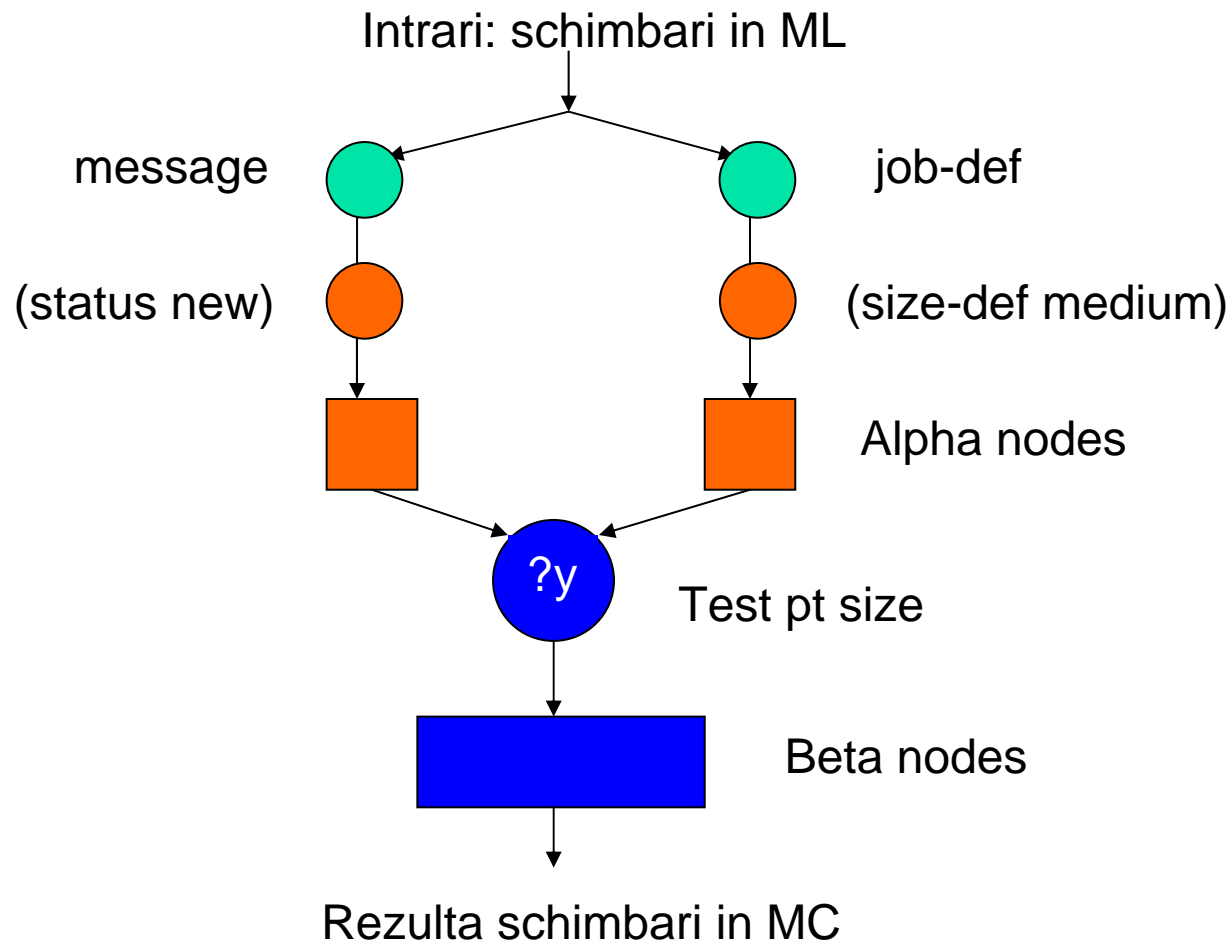
(message (jobs ?x) (size ?y) (status new))

(job-def (size ?y) (size-def medium))

=>

(assert (job (job-name ?x) (job-size ?y))))

Retea RETE pt regula



(defrule show-act

(a ?x)

(b ?x ?y)

(c ?y)

=> ...

ML

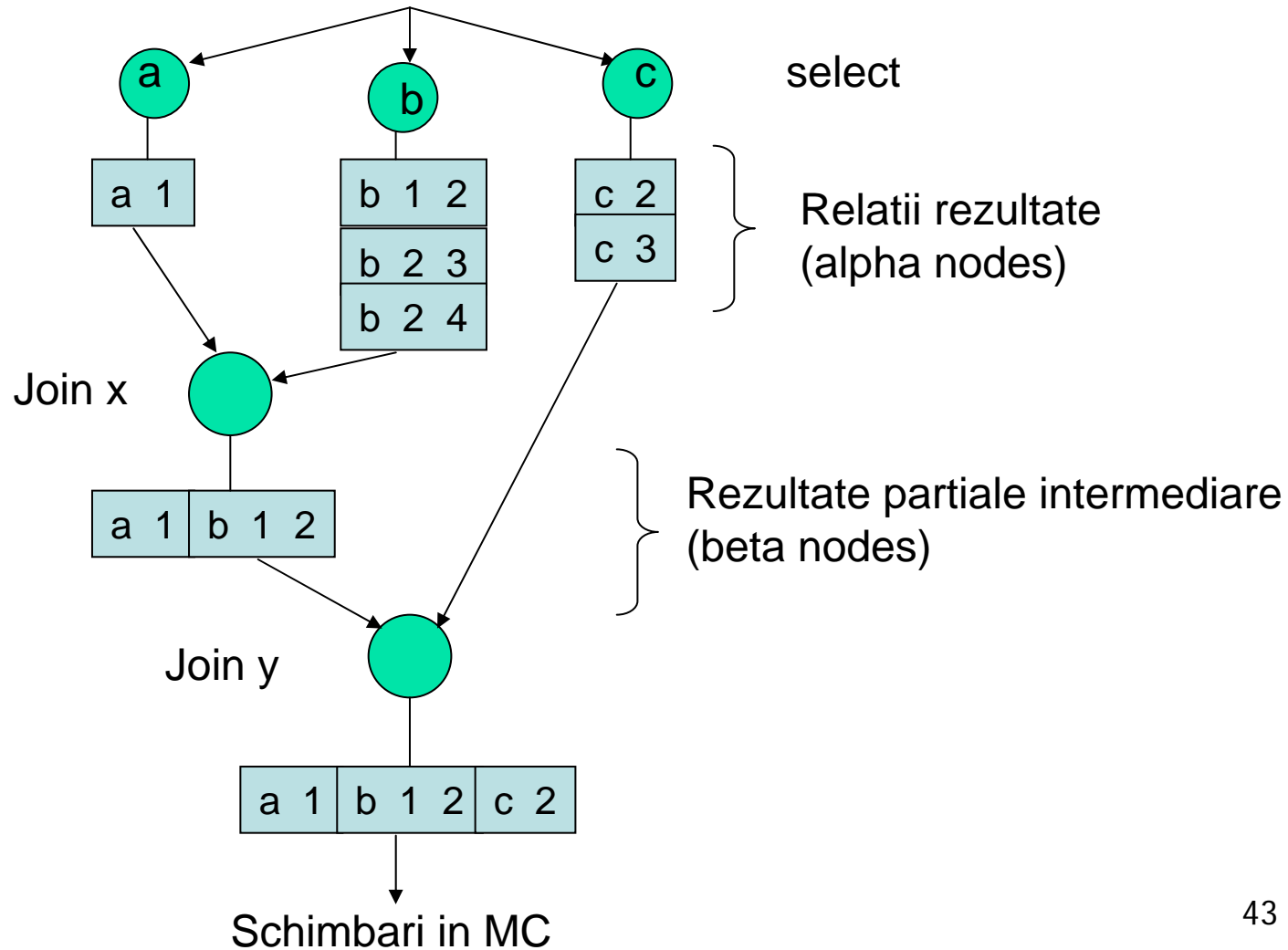
(a 1)

(b 1 2) (b 2 3) (b 2 4)

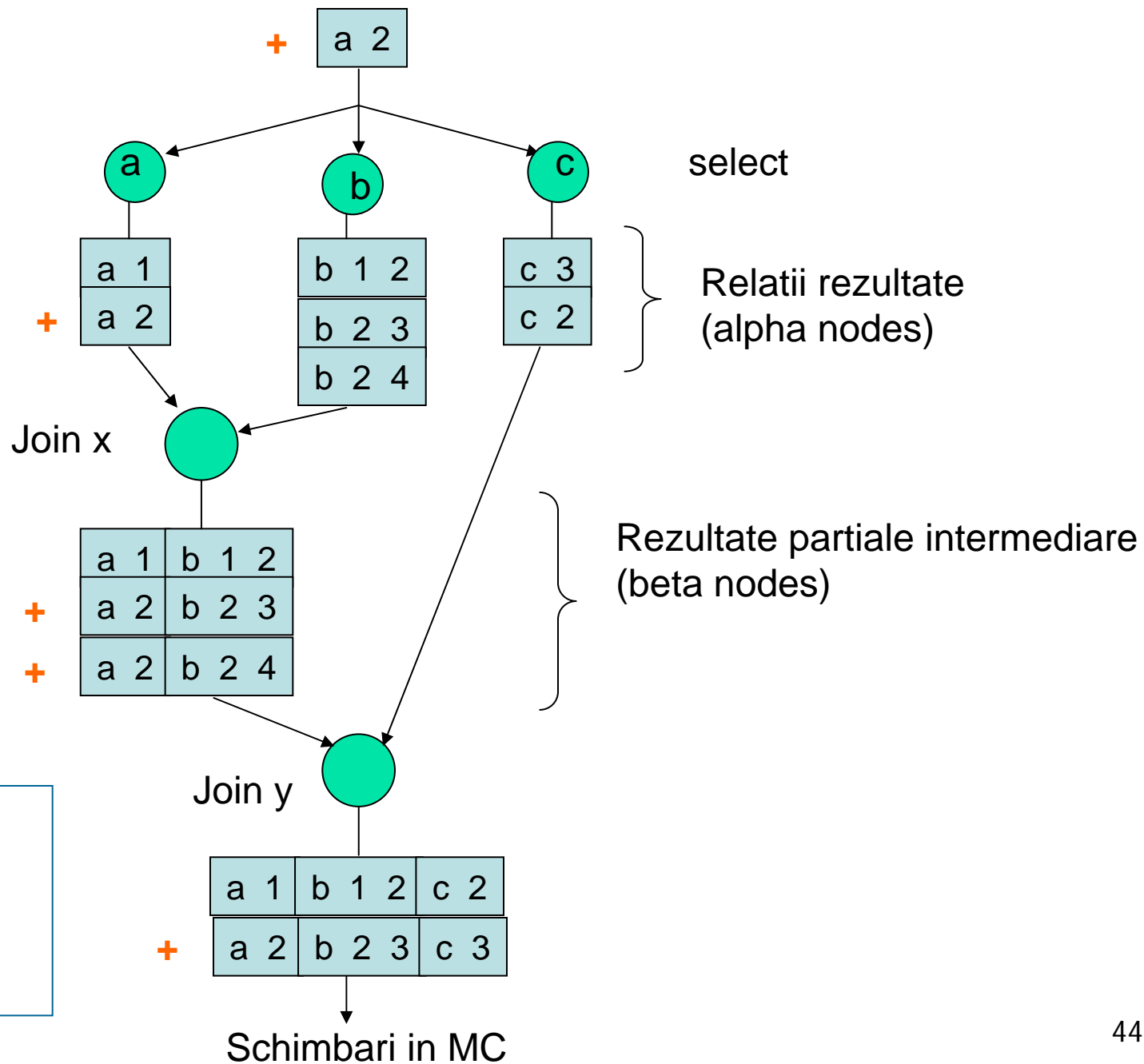
(c 2) (c 3)

Stare initiala a retelei

show-act



Adaugare WME (assert)



(defrule show-act

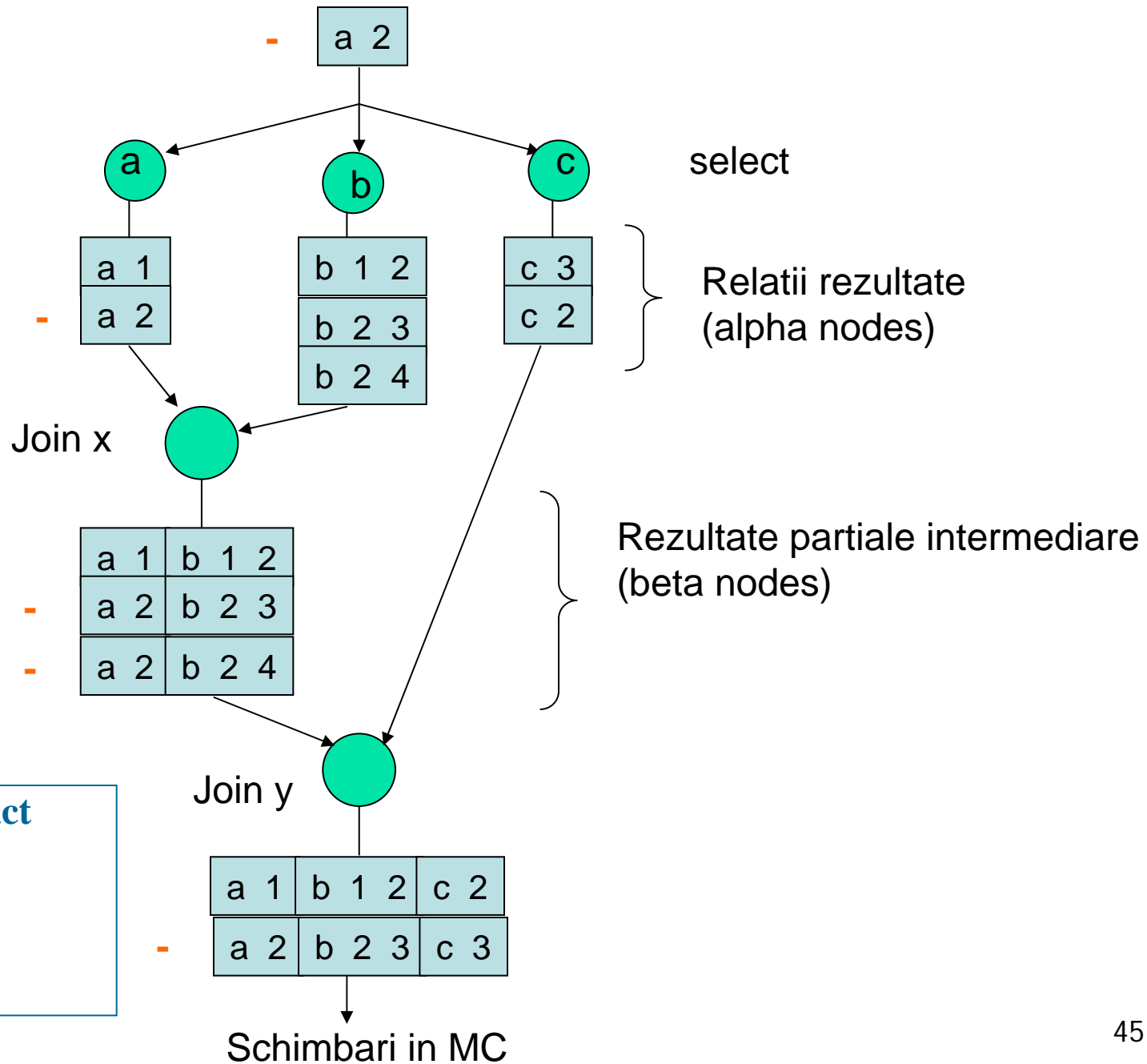
(a ?x)

(b ?x ?y)

(c ?y)

=> ...

Eliminare WME (retract)



```
(defrule show-act
```

```
  (a ?x)
```

```
  (b ?x ?y)
```

```
  (c ?y)
```

```
=> ...
```

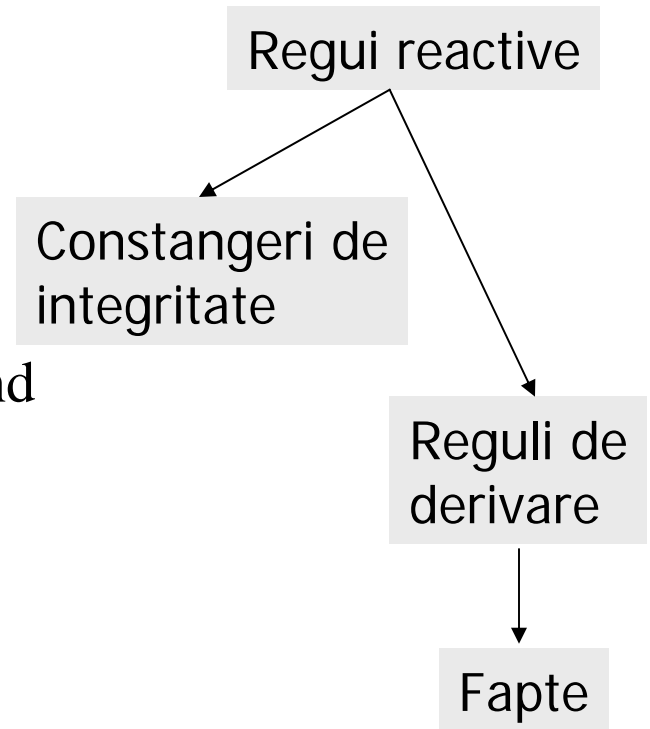



RuleML – pt Web

- **RuleML Initiative** - August 2000 Pacific Rim International Conference on Artificial Intelligence.
- **RuleML** – se bazeaza pe Datalog
- RuleML Initiative - dezvoltă limbaje bazate pe reguli deschise XML/RDF
- Aceasta permite schimbul de reguli între diferite sisteme, inclusiv componente software distribuite pe Web și sisteme client-server eterogene
- Limbajul RuleML – sintaxa XML pentru reprezentarea cunostintelor sub forma de reguli
- Integrarea cu ontologii: sistemul de reguli trebuie să deriveze/utilizeze cunoștințe din ontologie

RuleML

- **Reguli reactive** = Observa/verifica anumite evenimente/conditii si executa o actiune – numai forward
- **Constrangeri de integritate** = reguli speciale care semnaleaza inconsistente cand se indeplinesc anumite conditii – numai forward
- **Reguli de inferenta** (derivare) = reguli reactive speciale cu actiuni care adauga o concluzie daca conditiile (premisele sunt adevarate) - Se pot aplica atat forward cat si backward
- **Fapte** = reguli de inferenta particulare





RuleML

■ Reguli reactive

```
<rule>  <_body> <and> prem1 ... premN </and> </_body>
        <_head> action </_head>
</rule>
```

■ Constrangeri de integritate

```
<ic>    <_head> inconsistency </_head>
        <_body> <and> prem1 ... premN </and> </_body>
</ic>
```

implementate ca

```
<rule>  <_body> <and> prem1 ... premN </and> </_body>
        <_head> <signal> inconsistency </signal> </_head>
</rule>
```

RuleML

■ Reguli de inferenta/derivare

```
<imp>  <_head> conc </_head>  
        <_body> <and> prem1 ... premN </and> </_body>  
</imp>
```

implementate prin

```
<rule>  <_body> <and> prem1 ... premN </and> </_body>  
        <_head> <assert> conc </assert> </_head>  
</rule>
```

■ Fapte

```
<atom> <_head> conc </_head> </atom>
```

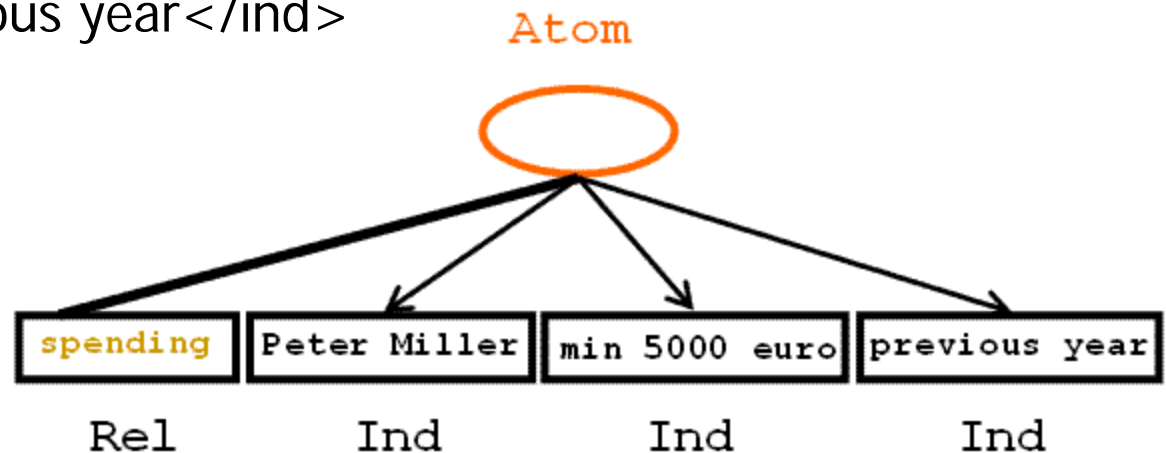
implementate prin

```
<imp>  <_head> conc </_head>  
        <_body> <and> </and> </_body> </imp>
```

RuleML

Fapte

```
<atom> <rel>spending</rel>  
  <ind>Peter Miller</ind>  
  <ind>min 5000 euro</ind>  
  <ind>previous year</ind>  
</atom>
```



`spending(petterMiller, min5000euro, previousYear).`

Reguli de inferenta/derivare

<imp>

<head> <atom>

<rel>discount</rel>

<var>customer</var>

<var>product</var>

<ind>7.5 percent</ind>

</atom>

</head>

<body>

<and>

<atom>

<rel>premium</rel>

<var>customer</var>

</atom>

<atom>

<rel>luxury</rel>

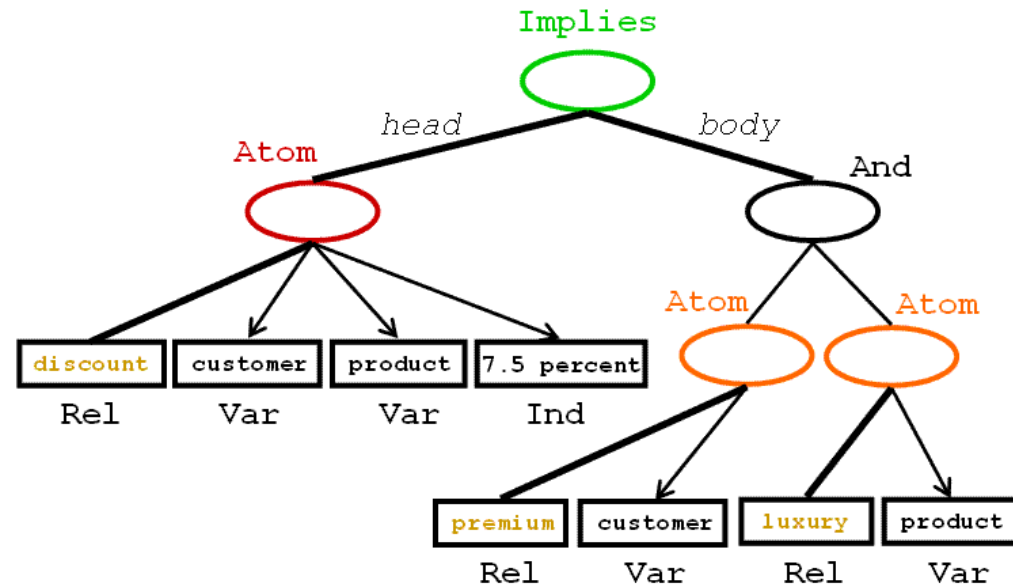
<var>product</var>

</atom>

</and>

</body>

</imp>



discount(Customer, Product, 7.5_percent):-
premium(Customer), luxury(Product).



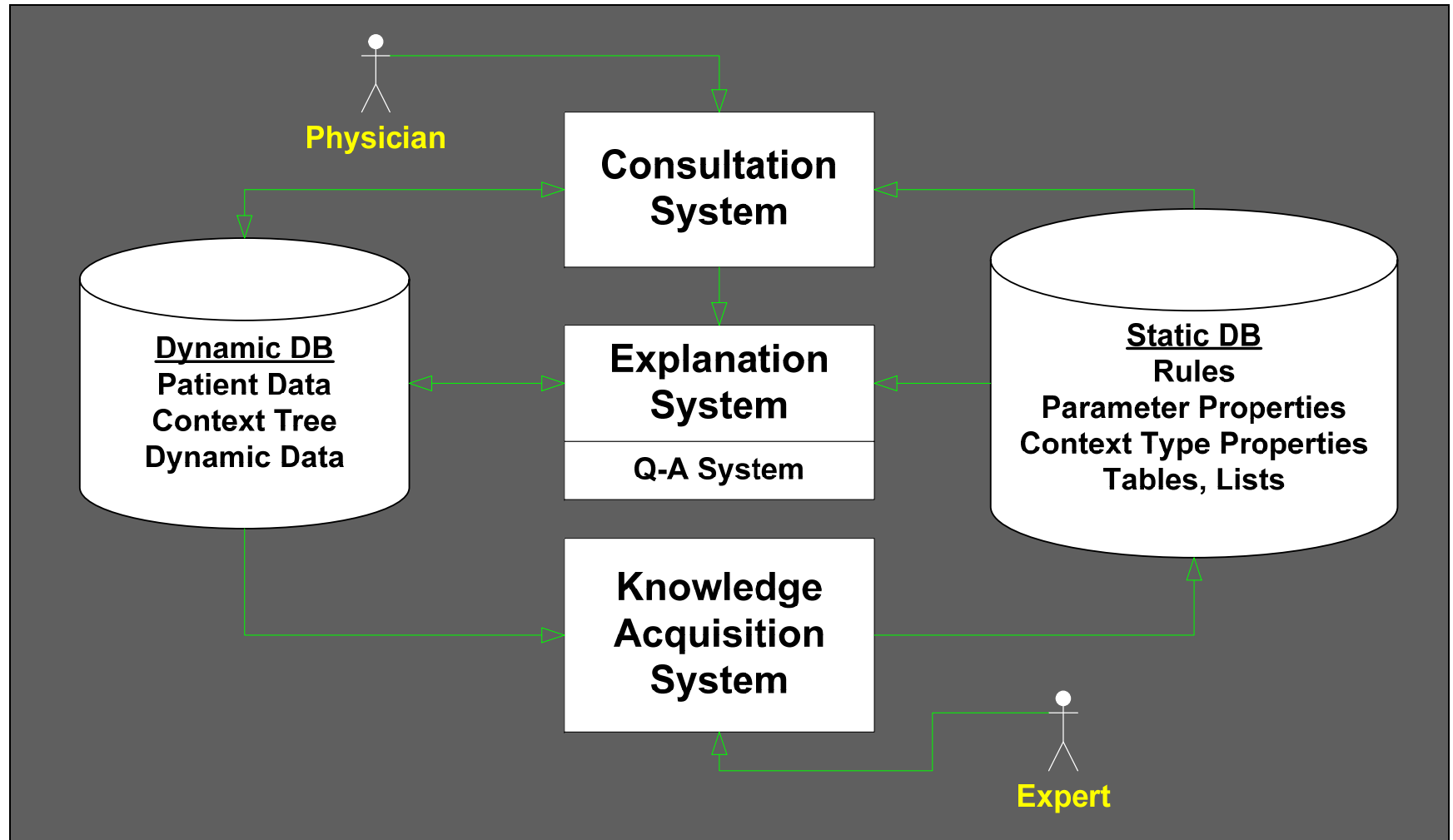
6. SBR cu inlantuire inapoi

Multe sisteme sunt combinate cu o forma de rationament incert

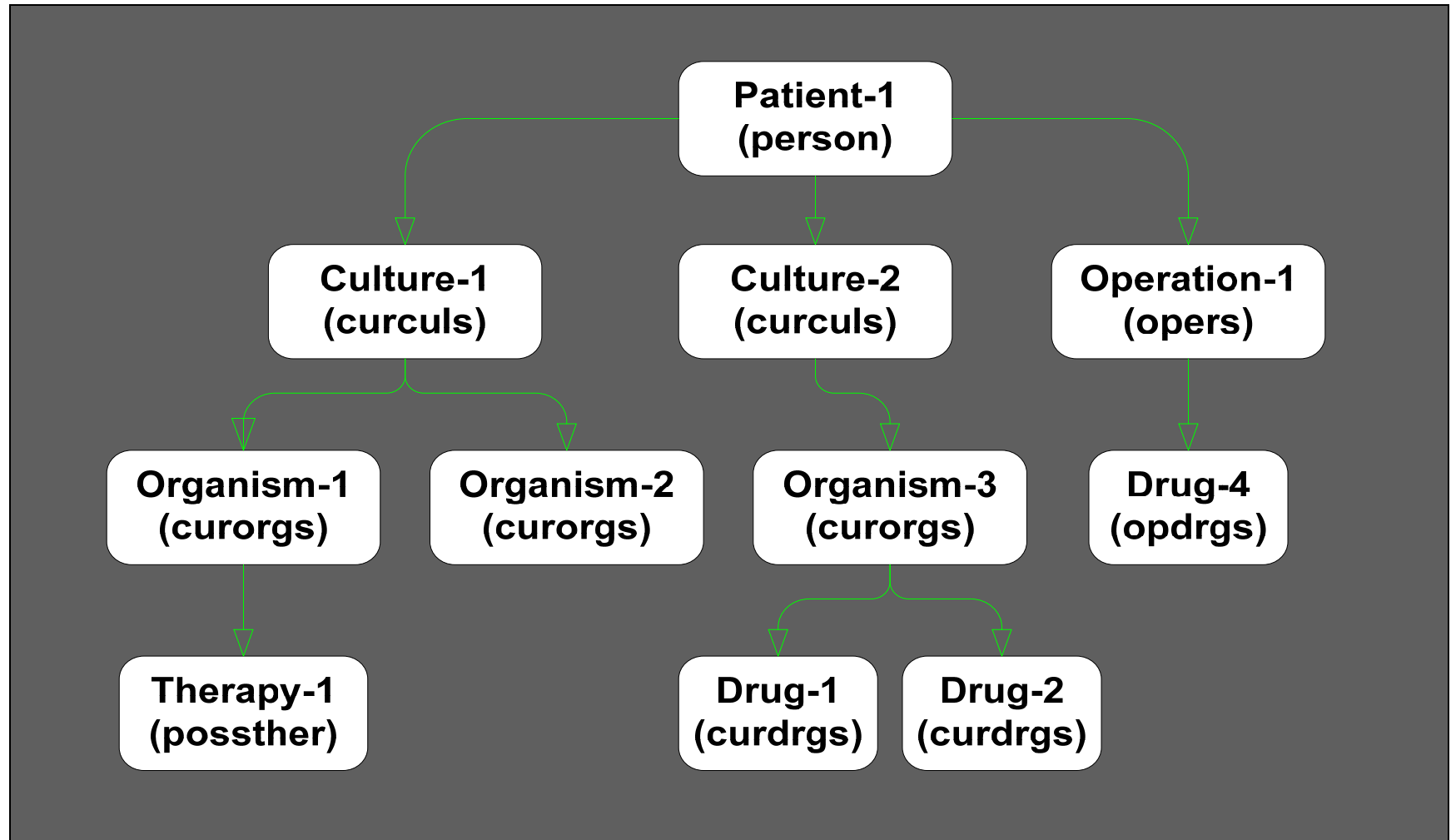
MYCIN – sistem expert pentru diagnosticarea infectiilor bacteriene ale sangelui

- Shortliffe, Davis, Buchanan, van Melle, si altii
 - Stanford Heuristic Programming Project
 - Infectious Disease Group, Stanford Medical

MYCIN - arhitectura



MYCIN – Context tree



Inlantuire inapoi bazat pe cautare in adancime
cu acumulare de probe

Control la nivel inalt

- Daca pacientul are o infectie semnificativa
- Identitatea organismului
- Care sunt medicamentele adecvate
- Alege pe cel mai bun



MYCIN - Explicatii

32) Was penicillinase added to this blood culture (CULTURE-1)?

****WHY**

[i.e. WHY is it important to determine whether penicillinase was added to CULTURE-1?]

[3.0] This will aid in determining whether ORGANISM-1 is a contaminant. It has already been established that

[3.1] the site of CULTURE-1 is blood, and

[3.2] the gram stain of ORGANISM-1 is grampos

Therefore, if

[3.3] penicillinase was added to this blood culture then there is weakly suggestive evidence...

Exemplu SBR cu inlantuire inapoi si calcul incert (CF)

- Exemplu: o baza de cunostinte pentru alegerea vinului adecvat unui meniu
- Valoarea fiecarui atribut este memorata impreuna cu coeficientul de certitudine asociat.
- Coeficientii de certitudine sunt valori pozitive in intervalul $[0,1]$.

(vin chardonney 0.8 riesling 0.6)

- O regula poate avea asociat un coeficient de certitudine
daca sos-meniu = sos-alb
atunci culoare-vin = alba 0.6

R11:daca componenta-meniu = curcan
atunci culoare-vin = rosie 0.7
si culoare-vin = alba 0.2

R12:daca componenta-meniu = peste
atunci culoare-vin = alba

R13:daca sos-meniu = sos-alb
atunci culoare-vin = alba 0.6

R14:daca componenta-meniu = porc
atunci culoare-vin = rosie

R21:daca sos-meniu = sos-alb
atunci tip-vin = sec 0.8
si tip-vin = demisec 0.6

R22:daca sos-meniu = sos-tomat
atunci tip-vin = dulce 0.8
si tip-vin = demisec 0.5

R23:daca sos-meniu = necunoscut
atunci tip-vin = demisec

R24:daca componenta-meniu = curcan
atunci tip-vin = dulce 0.6
si tip-vin = demisec 0.4

R31:daca culoare-vin = rosie

si tip-vin = dulce

atunci vin = gamay

R32:daca culoare-vin = rosie

si tip-vin = sec

atunci vin = cabernet-sauvignon

R33:daca culoare-vin = rosie

si tip-vin = demisec

atunci vin = pinot-noir

R34:daca culoare-vin = alba

si tip-vin = dulce

atunci vin = chenin-blanc

R35:daca culoare-vin = alba

si tip-vin = sec

atunci vin = chardonnay

R36:daca culoare-vin = alba

si tip-vin = demisec

atunci vin = riesling

scop(vin)

monovaloare(componenta-meniu)

monovaloare(culoare-vin)

monovaloare(sos-meniu)

multivaloare(tip-vin)

multivaloare(vin)

valori-legale(componenta-meniu) = [curcan, peste, porc]

valori-legale(sos-meniu) = [sos-alb, sos-tomat]

valori-legale(tip-vin) = [sec, demisec, dulce]

**valori-legale(vin) = [gamay, cabernet-sauvignon, pinot-noir,
chenin-blanc, chardonnay, riesling]**

valori-legale(culoare-vin) = [rosie, alba]



7. Sisteme bazate pe agenda

- Structura de control a anumitor sisteme bazate pe reguli nu foloseste nici inlantuirea inainte nici inlantuirea inapoi a regulilor, ci o *strategie de control de tip agenda*.
- Strategie utila in cazul in care se foloseste un *criteriu de selectie* a regulilor bazat pe **preferinta starilor**, deci o functie euristica de evaluare.
- O *agenda* este o lista de sarcini pe care sistemul trebuie sa le execute.
- O sarcina are asociata una sau mai multe **justificari** - prioritate

Algoritm: Strategia de control de tip "agenda"

1. Initializeaza agenda cu sarcina de executat
2. **repeta**
 - 2.1. Selecteaza sarcina cu prioritate maxima, T
 - 2.2. Executa sarcina T in limitele unor resurse de timp si spatiu determinate de importanta lui T
 - 2.3. **pentru** fiecare noua sarcina T_i generata de T
executa
 - 2.3.1. **daca** T_i este deja in agenda
atunci
 - i. Fie T'_i copia lui T_i din agenda
 - ii. **daca** justificarile lui T'_i nu includ justificarea lui T_i
atunci adauga justificarea lui T_i justificarilor lui T'_i
 - iii. Inlocuieste T'_i cu T_i

2.3.2. **altfel** adauga T_i si justificarea asociata in agenda

2.3.3. Calculeaza prioritatea sarcinii T_i pe baza
justificarilor asociate

pana agenda satisface conditia de stare finala **sau**
agenda este vida

sfarsit.