



Inteligența Artificială

Universitatea Politehnica București
Anul universitar 2013-2014

Adina Magda Florea



Curs nr. 4

Reprezentarea cunostintelor in IA

Modelul logicii simbolice

- Reprezentarea cunostintelor in logica simbolica
- Sistem formal
- Logica propozitiilor
- Logica cu predicate de ordinul I
- Demonstrarea teoremelor prin respingere rezolutiva
- Legatura cu limbajul Prolog



1. Reprezentarea cunostintelor in LS

- Logica – avantaje
- Puterea de reprezentare a diverselor logici simbolice
- Conceptualizare + exprimarea in limbaj
- Limbaj formal: sintaxa, semantica
- Reguli de inferenta



2. Sistem formal

- Un **sistem formal** este un cuadru plu $S = \langle A, F, A, \mathcal{R} \rangle$
- O *regula de inferenta* $R \in \mathcal{R}$ de aritate n este o corespondenta:

$$R \subseteq F^n \times F, \bar{y} = \langle y_1, \dots, y_n \rangle \xrightarrow{R} x, x, y_i \in F, \forall i = 1, n$$

- Fie multimea de premise $\Gamma = \{y_1, \dots, y_n\}$ $E_0 = \Gamma \cup A$

$$E_1 = E_0 \bigcup_{n \geq 1} \{x | \exists \bar{y} \in E_0^n, \bar{y} \mathcal{R} x\} \quad E_2 = E_1 \bigcup_{n \geq 1} \{x | \exists \bar{y} \in E_1^n, \bar{y} \mathcal{R} x\}$$

- Un element x din E_i ($i \geq 0$)

este o **consecinta** a multimii de premise Γ



Sistem formal - cont

- Daca $E_0 = \Gamma \cup A$ atunci $x \in E_i$ este **deductibil** din Γ
$$\Gamma \vdash_S x$$
- Secventa r.i. - **deductie**
- Daca $E_0 = A$ ($\Gamma = \phi$) atunci elementele lui E_i se numesc **teoreme**
- Fie $x \in E_i$ o teorema; se obtine prin aplicarea succesiva a r.i. asupra formulelor din E_i
- Secventa de reguli - **demonstratie** . $\vdash_{\mathcal{R}} x$



3. Logica propozitiilor

3.1 Sintaxa

- Alfabet

- O **formula bine formata** in calculul propozitional se defineste recursiv astfel:

(1) Un atom este o formula bine formata

(2) Daca P este formula bine formata, atunci $\sim P$ este formula bine formata.

(3) Daca P si Q sint formule bine formate atunci $P \wedge Q$, $P \vee Q$, $P \rightarrow Q$ si $P \leftrightarrow Q$ sint formule bine formate.

(4) Multimea formulelor bine formate este generata prin aplicarea repetata a regulilor (1)..(3) de un numar finit de ori.



3.2 Semantica

- Interpretare
- *Functia de evaluare a unei formule*
- Proprietatile fbf
 - Valida/tautologie
 - Realizabila
 - Inconsistenta
 - Formule echivalente



Semantica - cont

- O formula F este o *consecinta logica* a unei formule P daca F are valoarea adevarat in toate interpretarile in care P are valoarea adevarat.
- O formula F este *consecinta logica* a unei multimi de formule P_1, \dots, P_n daca formula F este adevarata in toate interpretarile in care P_1, \dots, P_n sunt adevarate.
- Consecinta logica se noteaza $P_1, \dots, P_n \Rightarrow F$.
- **Teorema.** Formula F este consecinta logica a unei multimi de formule P_1, \dots, P_n daca formula $P_1 \wedge \dots \wedge P_n \rightarrow F$ este valida.
- **Teorema.** Formula F este consecinta logica a unei multimi de formule P_1, \dots, P_n daca formula $P_1 \wedge \dots \wedge P_n \wedge \sim F$ este inconsistentă.

Legi de echivalenta

Idempotentă	$P \vee P \equiv P$	$P \wedge P \equiv P$	
Asociativitate	$(P \vee Q) \vee R \equiv P \vee (Q \vee R)$	$(P \wedge Q) \wedge R \equiv P \wedge (Q \wedge R)$	
Comutativitate	$P \vee Q \equiv Q \vee P$	$P \wedge Q \equiv Q \wedge P$	$P \leftrightarrow Q \equiv Q \leftrightarrow P$
Distributivitate	$P \wedge (Q \vee R) \equiv (P \wedge Q) \vee (P \wedge R)$	$P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$	
De Morgan	$\sim (P \vee Q) \equiv \sim P \wedge \sim Q$	$\sim (P \wedge Q) \equiv \sim P \vee \sim Q$	
Eliminarea implicatiei	$P \rightarrow Q \equiv \sim P \vee Q$		
Eliminarea implicatiei duble	$P \leftrightarrow Q \equiv (P \rightarrow Q) \wedge (Q \rightarrow P)$		



3.3 Obținerea de noi cunoștințe

- Conceptualizare
- Reprezentare în limbaj
- Teoria modelului

$$KB \models_S x$$

- Teoria demonstrației

$$KB \vdash_{\mathcal{R}} x$$

- Logici monotone
- Logici nemonotone



3.4 Reguli de inferenta

- *Modus Ponens*
$$\frac{P \quad P \rightarrow Q}{Q}$$
- *Substitutia*
- *Regula inlantuirii*
$$\frac{P \rightarrow Q \quad Q \rightarrow R}{P \rightarrow R}$$
- *Regula introducerii conjunctiei*
$$\frac{P \quad Q}{P \wedge Q}$$
- *Regula transpozitiei*
$$\frac{P \rightarrow Q}{\sim Q \rightarrow \sim P}$$



Exemplu

- *Mihai are bani*
- *Masina este alba*
- *Masina este frumoasa*
- *Daca masina este alba sau masina este frumoasa si Mihai are bani atunci Mihai pleaca in vacanta*
- B
- A
- F
- $(A \vee F) \wedge B \rightarrow C$



4. Logica cu predicate de ordinul I

4.1 Sintaxa

Fie D un domeniu de valori. Un *termen* se definește astfel:

- (1) O constantă este un termen cu valoare fixă aparținând domeniului D .
- (2) O variabilă este un termen ce poate primi valori diferite din domeniul D .
- (3) Dacă f este o funcție de n argumente și t_1, \dots, t_n sunt termeni, atunci $f(t_1, \dots, t_n)$ este termen.
- (4) Toți termenii sunt generați prin aplicarea regulilor (1)...(3).



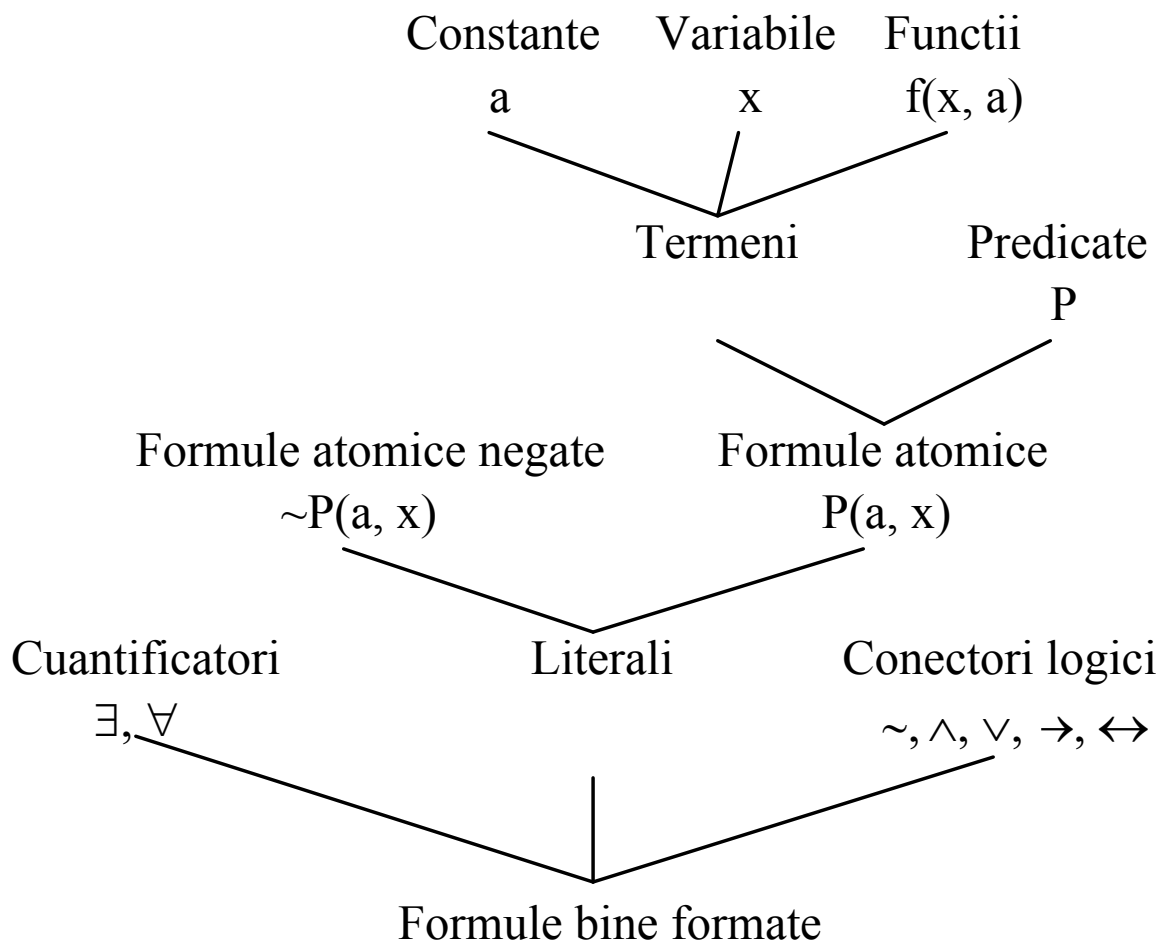
Sintaxa LP - cont

- Predicat de aritate n
- Atom sau formula atomica.
- Literal

O *formula bine formata* in logica cu predicate de ordinul I se defineste astfel:

- (1) Un atom este o formula bine formata
- (2) Daca $P[x]$ este fbf, atunci $\sim P[x]$ este fbf.
- (3) Daca $P[x]$ si $Q[x]$ sunt fbf atunci $P[x] \wedge Q[x]$, $P[x] \vee Q[x]$, $P \rightarrow Q$ si $P \leftrightarrow Q$ sunt fbf.
- (4) Daca $P[x]$ este fbf atunci $\forall x P[x]$, $\exists x P[x]$ sunt fbf.
- (5) Multimea formulelor bine formate este generata prin aplicarea repetata a regulilor (1)..(4) de un numar finit de ori.

Sintaxa pe scurt





FNC, FND

- O formula bine formata este in *forma normala conjunctiva*, pe scurt FNC, daca formula are forma

$$F_1 \wedge \dots \wedge F_n,$$

unde F_i , $i=1, n$ sunt formule formate dintr-o disjunctie de literali ($L_{i1} \vee \dots \vee L_{im}$).

- O formula bine formata este in *forma normala disjunctiva*, pe scurt FND, daca formula are forma ,

$$F_1 \vee \dots \vee F_n,$$

unde F_i , $i=1, n$ sunt formule formate dintr-o conjunctie de literali ($L_{i1} \wedge \dots \wedge L_{im}$)



4.2 Semantica LP

- *Interpretarea* unei formule F in logica cu predicate de ordinul I consta in fixarea unui domeniu de valori nevid D si a unei asignari de valori pentru fiecare constanta, functie si predicat ce apar in F astfel:
- (1) Fiecarei constante i se asociaza un element din D .
- (2) Fiecarei functii f , de aritate n , i se asociaza o corespondenta $D^n \rightarrow D$, unde
$$D^n = \{(x_1, \dots, x_n) | x_1 \in D, \dots, x_n \in D\}$$
- (3) Fiecarui predicat de aritate n , i se asociaza o corespondenta $P: D^n \rightarrow \{\mathbf{a}, \mathbf{f}\}$

Interpretare I

$$(\forall x)((A(a, x) \vee B(f(x))) \wedge C(x)) \rightarrow D(x)$$

$$D = \{1, 2\}$$

a	f(1)	f(2)	A(2,1)	A(2,2)	B(1)	B(2)	C(1)	C(2)	D(1)	D(2)
2	2	1	a	f	a	f	a	f	f	a

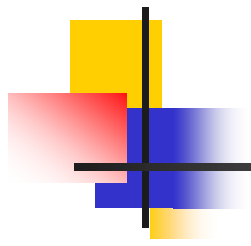
$$X=1 \quad ((\mathbf{a} \vee \mathbf{f}) \wedge \mathbf{a}) \rightarrow \mathbf{f}$$

$$X=2 \quad ((\mathbf{f} \vee \mathbf{a}) \wedge \mathbf{f}) \rightarrow \mathbf{a}$$



4.3 Proprietatile fbf in LP

- Valida/tautologie
- Realizabila
- Inconsistenta
- Echivalente
- F - consecinta logica a unei formule P
- F - consecinta logica a unei multimi de formule P_1, \dots, P_n
- **Teorema.** Formula F este consecinta logica a unei multimi de formule P_1, \dots, P_n daca formula $P_1 \wedge \dots \wedge P_n \rightarrow F$ este valida.
- **Teorema.** Formula F este consecinta logica a unei multimi de formule P_1, \dots, P_n daca formula $P_1 \wedge \dots \wedge P_n \wedge \sim F$ este inconsistentă.



Echivalenta cuantificatorilor

$(Qx)F[x] \vee G \equiv (Qx)(F[x] \vee G)$	$(Qx)F[x] \wedge G \equiv (Qx)(F[x] \wedge G)$
$\sim ((\forall x)F[x]) \equiv (\exists x)(\sim F[x])$	$\sim ((\exists x)F[x]) \equiv (\forall x)(\sim F[x])$
$(\forall x)F[x] \wedge (\forall x)H[x] \equiv (\forall x)(F[x] \wedge H[x])$	$(\exists x)F[x] \vee (\exists x)H[x] \equiv (\exists x)(F[x] \vee H[x])$
$(Q_1x)F[x] \wedge (Q_2x)H[x] \equiv (Q_1x)(Q_2z)(F[x] \wedge H[z])$	$(Q_1x)F[x] \vee (Q_2x)H[x] \equiv (Q_1x)(Q_2z)(F[x] \vee H[z])$



Exemple

- Toate merele sunt rosii
- Toate obiectele sunt mere rosii
- Exista un mar rosu
- Toate pachetele din camera 27 sunt mai mici decat orice pachet din camera 28

Toate ciupercile purpurii sunt otravitoare

- $\forall x (\text{Purpuriu}(x) \wedge \text{Ciuperca}(x)) \Rightarrow \text{Otravitor}(x)$
- $\forall x \text{ Purpuriu}(x) \Rightarrow (\text{Ciuperca}(x) \Rightarrow \text{Otravitor}(x))$
- $\forall x \text{ Ciuperca}(x) \Rightarrow (\text{Purpuriu}(x) \Rightarrow \text{Otravitor}(x))$

$(\forall x)(\exists y) \text{ iubeste}(x,y)$

$(\exists y)(\forall x) \text{ iubeste}(x,y)$



4.4. Reguli de inferenta in LP

- Modus Ponens (MP)
$$\frac{P(a) \quad (\forall x)(P(x) \rightarrow Q(x))}{Q(a)}$$
- Substitutia
- Regula inlantuirii
- Transpozitia
- Eliminarea conjunctiei (ElimC)
- Introducerea conjunctiei (IntrC)
- Instantierea universală (InstU)
- Instantierea existentială (InstE)
- Rezolutia

Exemplu

- Caii sunt mai rapizi decat cainii si exista un ogar care este mai rapid decat orice iepure. Se stie ca Harry este un cal si ca Ralph este un iepure. Sa se demonstreze faptul ca Harry este mai rapid decat Ralph.
- $\text{Cal}(x)$ $\text{Ogar}(y)$
- $\text{Caine}(y)$ $\text{Iepure}(z)$
- $\text{MaiRapid}(y,z)$

$$\forall x \forall y \text{ Cal}(x) \wedge \text{Caine}(y) \Rightarrow \text{MaiRapid}(x,y)$$

$$\exists y \text{ Ogar}(y) \wedge (\forall z \text{ Iepure}(z) \Rightarrow \text{MaiRapid}(y,z))$$

$$\text{Cal}(\text{Harry})$$

$$\text{Iepure}(\text{Ralph})$$

$$\forall y \text{ Ogar}(y) \Rightarrow \text{Caine}(y)$$

$$\forall x \forall y \forall z \text{ MaiRapid}(x,y) \wedge \text{MaiRapid}(y,z) \Rightarrow \text{MaiRapid}(x,z)$$

Exemplu de demonstrare

■ Teorema: MaiRapid(Harry, Ralph) ?

■ Demonstrare folosind reguli de inferenta

1. $\forall x \forall y \text{ Cal}(x) \wedge \text{Caine}(y) \Rightarrow \text{MaiRapid}(x,y)$

2. $\exists y \text{ Ogar}(y) \wedge (\forall z \text{ Iepure}(z) \Rightarrow \text{MaiRapid}(y,z))$

3. $\forall y \text{ Ogar}(y) \Rightarrow \text{Caine}(y)$

4. $\forall x \forall y \forall z \text{ MaiRapid}(x,y) \wedge \text{MaiRapid}(y,z) \Rightarrow \text{MaiRapid}(x,z)$

5. $\text{Cal}(\text{Harry})$

6. $\text{Iepure}(\text{Ralph})$

7. $\text{Ogar}(\text{Greg}) \wedge (\forall z \text{ Iepure}(z) \Rightarrow \text{MaiRapid}(\text{Greg},z))$ 2, InstE

8. $\text{Ogar}(\text{Greg})$ 7, ElimC

9. $\forall z \text{ Iepure}(z) \Rightarrow \text{MaiRapid}(\text{Greg},z)$ 7, ElimC

Exemplu de demonstrare - cont

- | | | |
|-----|--|---------------|
| 10. | $Iepure(Ralph) \Rightarrow MaiRapid(Greg, Ralph)$ | 9, InstU |
| 11. | $MaiRapid(Greg, Ralph)$ | 6, 10, MP |
| 12. | $Ogar(Greg) \Rightarrow Caine(Greg)$ | 3, InstU |
| 13. | $Caine(Greg)$ | 12, 8, MP |
| 14. | $Cal(Harry) \wedge Caine(Greg) \Rightarrow MaiRapid(Harry, Greg)$ | 1, InstU |
| 15. | $Cal(Harry) \wedge Caine(Greg)$ | 5, 13, IntrC |
| 16. | $MaiRapid(Harry, Greg)$ | 14, 15, MP |
| 17. | $MaiRapid(Harry, Greg) \wedge MaiRapid(Greg, Ralph) \Rightarrow$
$MaiRapid(Harry, Ralph)$ | 4, InstU |
| 18. | $MaiRapid(Harry, Greg) \wedge MaiRapid(Greg, Ralph)$ | 16, 11, IntrC |
| 19. | $MaiRapid(Harry, Ralph)$ | 17, 18, MP |



5. Demonstrarea teoremelor prin respingere rezolutiva

5.1 Forma standard

- Clauza
- Clauza Horn
- Clauza Horn definita
- Clauza vida
- Transformare in forma clauzala
- Forme de notare
 - Forma clauzala $\sim A(x) \vee B(x)$
 - Forma clauzala multime $\{\sim A(x), B(x)\}$
 - Forma consecinta (Gentzer) $A(x) \Rightarrow B(x)$
 - Forma Prolog $B(x) :- A(x)$



5.2 Unificarea expresiilor

- Substitutie α
- Unificator
- Cel mai general unificator β
- Expresie
- Algoritm de unificare

Algoritm Unifica(E1,E2): Unificarea expresiilor

1. **daca** E1 si E2 sunt constante
atunci
 - 1.1. **daca** E1=E2
atunci intoarce { }
 - 1.2. **intoarce** INSUCCES
2. **daca** E1 este variabila **atunci**
 - 2.1 **daca** E1 apare in E2 **atunci intoarce** INSUCCES
altfel intoarce {E1/E2}
3. **daca** E2 este variabila **atunci**
 - 3.1 **daca** E2 apare in E1 **atunci intoarce** INSUCCES
altfel intoarce {E2/E1}
4. **daca** E1= { } sau E2 = { } **atunci intoarce** INSUCCES

5. **daca** $E1 = \text{simb}(t_{11}, \dots, t_{1n})$ **si** $E2 = \text{simb}(t_{21}, \dots, t_{2n})$
atunci

5.1 $X \leftarrow t_{11}, Y \leftarrow t_{21}$

5.2 $\text{Rest}_1 \leftarrow t_{12}, \dots, t_{1n}, \text{Rest}_2 \leftarrow t_{22}, \dots, t_{2n}$

5.3 $\alpha1 \leftarrow \text{Unifica}(X, Y)$

5.4 **daca** $\alpha1 = \text{INSUCCES}$
atunci intoarce INSUCCES

5.5 $G_1 \leftarrow \text{aplica}(\alpha1, \text{Rest}_1), G_2 \leftarrow \text{aplica}(\alpha1, \text{Rest}_2)$

5.6 $\alpha2 \leftarrow \text{Unifica}(G_1, G_2)$

5.7 **daca** $\alpha2 = \text{INSUCCES}$
atunci intoarce INSUCCES
altfel intoarce $(\alpha1, \alpha2)$

6. **intoarce** INSUCCES
sfarsit.

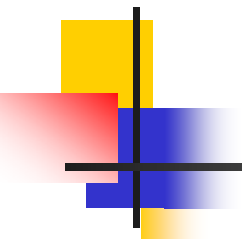


5.3 Rezolutia in logica propozitiilor

- Rezolvent
- Clauze care rezolva
- Principiul demonstratiei
- Arbore de demonstrare

Algorithm: **Respingerea prin rezolutie in logica propozitionala.**

1. Converteste setul de axiome A in forma clauzala si obtine multimea de clauze S
 2. Neaga teorema, transforma teorema negata in forma clauzala si adauga rezultatul la S
 3. **repeta**
 - 3.1. Selecteaza o pereche de clauze C_1 si C_2 din S
 - 3.2. Determina $R = \text{Res}(C_1, C_2)$
 - 3.3. **daca** $R \neq \square$
 atunci adauga R la S
 - pana** $R = \square$ **sau** nu mai exista nici o pereche de clauze care rezolva
 4. **daca** s-a obtinut clauza vida
 atunci teorema este adevarata (este demonstrata)
 5. **altfel** teorema este falsa
- sfarsit.**

- 
-
- Comentarii
 - strategie
 - decidabilitate



Exemplu

- *Mihai are bani*
- *Masina este alba*
- *Masina este frumoasa*
- *Daca masina este alba sau masina este frumoasa si Mihai are bani atunci Mihai pleaca in vacanta*
- B
- A
- F
- $(A \vee F) \wedge B \rightarrow C$



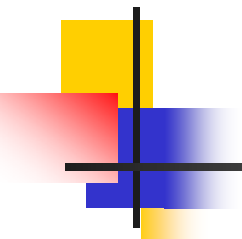
5.4 Rezolutia in logica predicatelor

- Rezolvent binar
- Clauze care rezolva
- Factor al unei clauze
- Rezolvent

Algoritm: **Respingerea prin rezolutie in logica predicatelor.**

1. Converteste setul de axiome A in forma clauzala si obtine multimea de clauze S
2. Neaga teorema, transforma teorema negata in forma clauzala si adauga rezultatul la S
3. **repeta**
 - 3.1. Selecteaza o pereche de clauze C_1 si C_2 din S
 - 3.2. Determina $R = \{\text{Res}(C_1, C_2)\}$
 - 3.3. **daca** $\square \notin R$
 atunci reuneste R cu S
pana s-a obtinut \square **sau** nu mai exista nici o pereche de clauze care rezolva **sau** un efort a fost epuizat
4. **daca** s-a obtinut clauza vida
 atunci teorema este adevarata (este demonstrata)
5. **altfel daca** nu mai exista nici o pereche de clauze care rezolva
 atunci teorema este falsa
 altfel nu se stie

sfarsit.

- 
-
- Comentarii
 - strategie
 - decidabilitate
 - completitudine

Exemplu

- Horses are faster than dogs and there is a greyhound that is faster than every rabbit. We know that Harry is a horse and that Ralph is a rabbit. Derive that Harry is faster than Ralph.
- Horse(x) Greyhound(y)
- Dog(y) Rabbit(z)
- Faster(y,z))

$$\forall x \forall y \text{ Horse}(x) \wedge \text{Dog}(y) \Rightarrow \text{Faster}(x,y)$$

$$\exists y \text{ Greyhound}(y) \wedge (\forall z \text{ Rabbit}(z) \Rightarrow \text{Faster}(y,z))$$

$$\text{Horse}(\text{Harry})$$

$$\text{Rabbit}(\text{Ralph})$$

$$\forall y \text{ Greyhound}(y) \Rightarrow \text{Dog}(y)$$

$$\forall x \forall y \forall z \text{ Faster}(x,y) \wedge \text{Faster}(y,z) \Rightarrow \text{Faster}(x,z)$$

A1. $\forall x \forall y \text{Horse}(x) \wedge \text{Dog}(y) \Rightarrow \text{Faster}(x,y)$
A2. $\exists y \text{Greyhound}(y) \wedge (\forall z \text{Rabbit}(z) \Rightarrow \text{Faster}(y,z))$
A3. $\text{Horse}(\text{Harry})$
A4. $\text{Rabbit}(\text{Ralph})$
A5. $\forall y \text{Greyhound}(y) \Rightarrow \text{Dog}(y)$
A6. $\forall x \forall y \forall z \text{Faster}(x,y) \wedge \text{Faster}(y,z) \Rightarrow \text{Faster}(x,z)$
T $\text{Faster}(\text{Harry},\text{Ralph})$

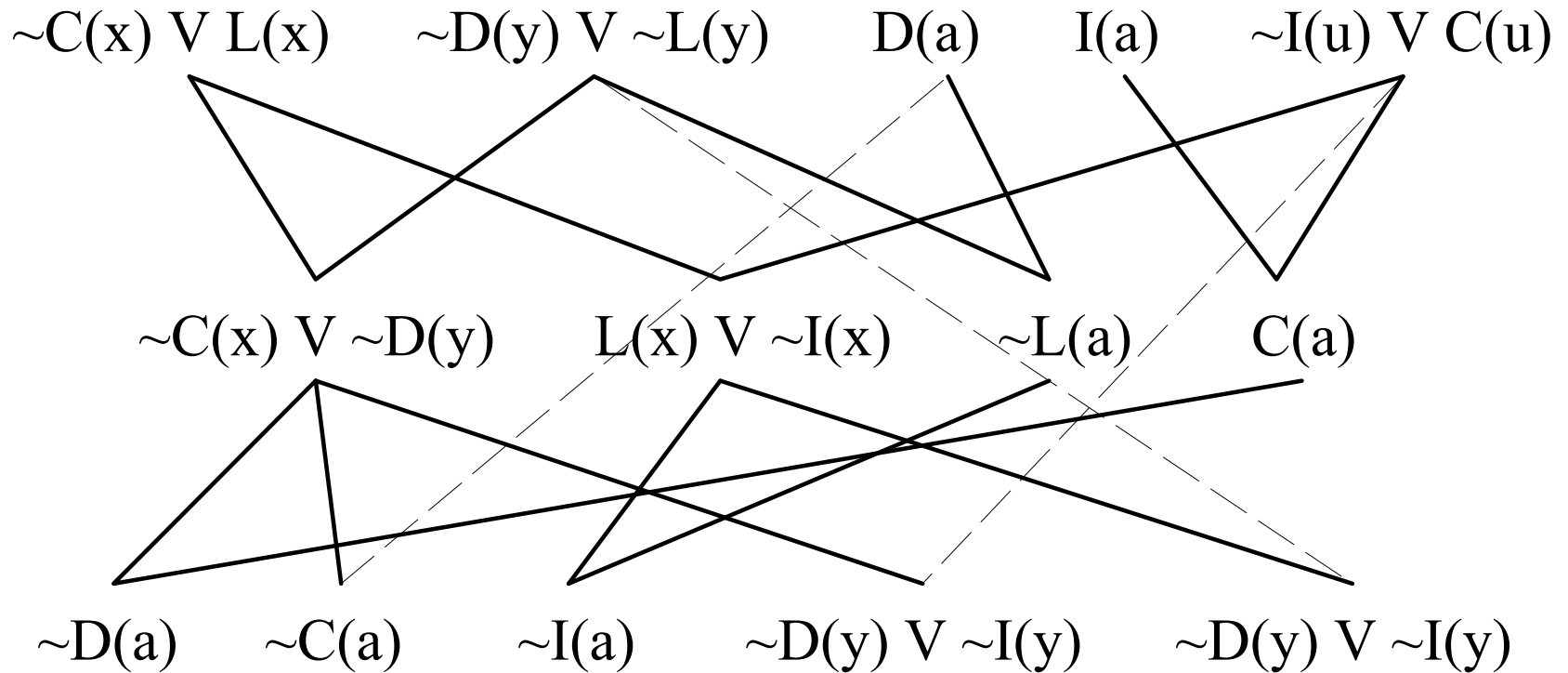
C1. $\sim \text{Horse}(x) \vee \sim \text{Dog}(y) \vee \text{Faster}(x,y)$
C2. $\text{Greyhound}(\text{Greg})$
C2' $\sim \text{Rabbit}(z) \vee \text{Faster}(\text{Greg},z)$
C3. $\text{Horse}(\text{Harry})$
C4. $\text{Rabbit}(\text{Ralph})$
C5. $\sim \text{Greyhound}(y) \vee \text{Dog}(y)$
C6. $\sim \text{Faster}(x,y) \vee \sim \text{Faster}(y,z) \vee \text{Faster}(x,z)$
C7. $\sim \text{Faster}(\text{Harry},\text{Ralph})$



5.5 Strategii rezolutive

- *Strategia dezvoltarii pe latime*
- *Strategia multimii suport*
- *Strategia rezolutiei liniare*
- *Strategia rezolutiei liniare de intrare*
- *Strategia rezolutiei unitare*
- *Strategia eliminarii*

Strategia dezvoltarii pe latime





Strategia dezvoltarii pe latime

- S_0, L_0
- $L_{k+1} \leftarrow \{\text{Res}(C_i, C_j) \mid C_i \in L_k, C_j \in S_k\}$
- $S_{k+1} \leftarrow S_k \cup L_{k+1}$
 $k = 0, 1, 2, \dots$

Strategia multimii suport

- S, T $S_0 = S \cup T, S \cap T = \emptyset$
- $S \leftarrow S \cup \{\text{Res}(C_i, C_j) \mid C_i \in S, C_j \in T\}$

Strategia rezolutiei liniare

- $S, \quad C_0 \in S$
- $C_1 \leftarrow \{\text{Res}(C_0, C_i) \mid C_0, C_i \in S\}$
- $C_{k+1} \leftarrow \{\text{Res}(C_k, C_i) \mid C_i \in \{C_{k-1}, C_{k-2}, \dots\} \cup S\}$
 $k=1, 2, 3, \dots$

Strategia rezolutiei liniare de intrare

- $S, \quad C_0 \in S$
- $C_1 \leftarrow \{\text{Res}(C_0, C_i) \mid C_0, C_i \in S\}$
- $C_{k+1} \leftarrow \{\text{Res}(C_k, C_i) \mid C_i \in S\}$
 $k=1, 2, 3, \dots$

Strategia rezolutiei unitare



Clauze subsumate

- O clauza C subsumeaza o clauza D daca si numai daca exista o substitutie α astfel incat $C\alpha \subseteq D$. D se numeste clauza subsumata.
- $C=P(x)$ $D=P(a) \vee Q(a)$

Strategia eliminarii

Verificare C subsumeaza D

$$\begin{aligned} D &= L_1 \vee \dots \vee L_m, & \alpha &= \{x_1/a_1, \dots, x_n/a_n\} \\ D\alpha &= L_1\alpha \vee \dots \vee L_m\alpha, & \sim D\alpha &= \sim L_1\alpha \wedge \dots \wedge \sim L_m\alpha \end{aligned}$$

1. $W \leftarrow \{\sim L_1\alpha, \dots, \sim L_m\alpha\}$
2. $k \leftarrow 0, U^0 \leftarrow \{C\}$
3. **daca** $\square \in U^k$ **atunci intoarce** SUCCES
4. $U^{k+1} \leftarrow \{\text{rez}(C_1, C_2) \mid C_1 \in U^k, C_2 \in W\}$
5. **daca** $U^{k+1} = \emptyset$ **atunci intoarce** INSUCCES
6. $k \leftarrow k+1$
7. **repeta** de la 3

sfarsit

5.6 Obținerea răspunsurilor

$(\forall x)(\text{EsteLa}(\text{mihai}, x) \rightarrow \text{EsteLa}(\text{grivei}, x))$

$\text{EsteLa}(\text{mihai}, \text{scoala})$

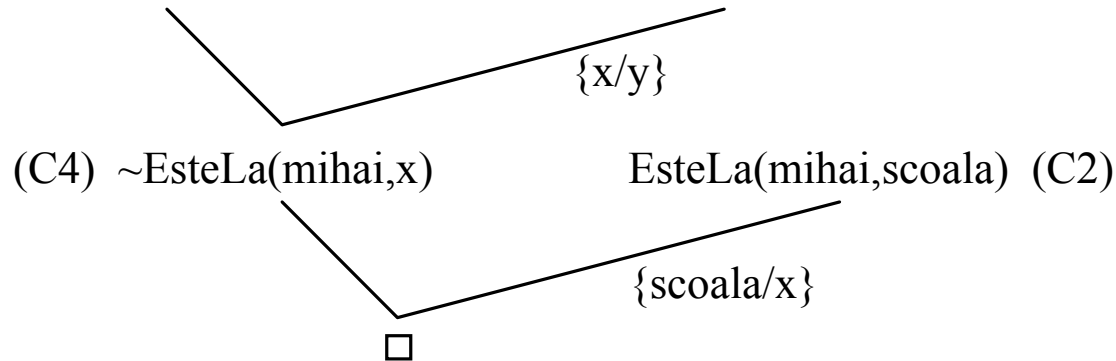
c1 $\sim \text{EsteLa}(\text{mihai}, x) \vee \text{EsteLa}(\text{grivei}, x)$

$(\exists y)\text{EsteLa}(\text{grivei}, y)$

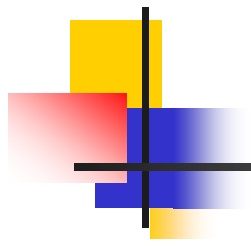
c2 $\text{EsteLa}(\text{mihai}, \text{scoala})$

c3 $\sim \text{EsteLa}(\text{grivei}, y)$

(C3) $\sim \text{EsteLa}(\text{grivei}, y)$ $\sim \text{EsteLa}(\text{mihai}, x) \vee \text{EsteLa}(\text{grivei}, x)$ (C1)



(a)



(C3) $\sim \text{EsteLa}(\text{grivei}, y) \vee \mathbf{\text{EsteLa}(\text{grivei}, y)}$ $\sim \text{EsteLa}(\text{mihai}, x) \vee \text{EsteLa}(\text{grivei}, x)$ (C1)

(C4) $\sim \text{EsteLa}(\text{mihai}, x) \vee \mathbf{\text{EsteLa}(\text{grivei}, x)}$ $\text{EsteLa}(\text{mihai}, \text{scoala})$ (C2)

$\mathbf{\text{EsteLa}(\text{grivei}, \text{scoala})}$

(b)



6. Legatura cu limbajul Prolog

- R. Kowalski, A. Colmerauer - începutul anilor '70
- Corespondenta Logica cu Predicate si limbajul Prolog – *clauze Horn definite*
- Structura logica vs structura de control si executie



6.1 Structura logica a limbajului Prolog

- Un scop **S** este **adevarat** intr-un program Prolog, adica *poate fi satisfacut* sau *derivă logic din program*, daca si numai daca:
 1. exista o clauza **C** a programului;
 2. exista o instanta **I** a clauzei **C** astfel incat:
 - antetul lui **I** sa fie identic cu cel al lui **S**;
 - toate scopurile din corpul lui **I** sunt adevarate, deci pot fi satisfacute



6.2 Structura de control si executie a limbajului Prolog

- Rezolutie liniara de intrare
- Ordinea faptelor si clauzelor
- Ordinea scopurilor in corpul clauzelor
- Backtracking

6.3 Prolog – exemplu

% parinte(IndividX, IndividY)

% stramos(IndividX, IndividZ)

parinte(vali, gelu).

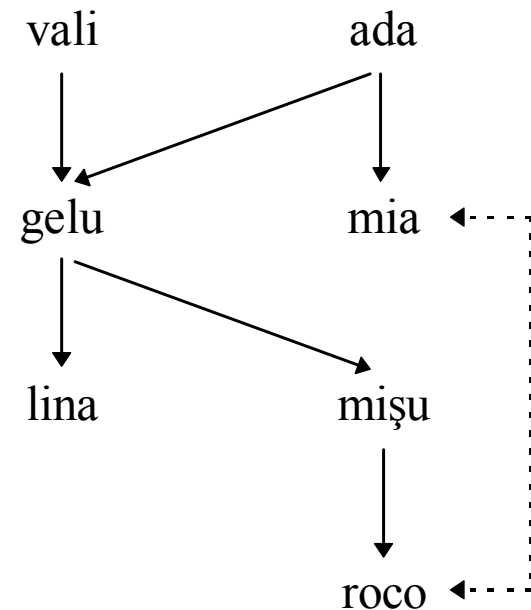
parinte(ada, gelu).

parinte(ada, mia).

parinte(gelu, lina).

parinte(gelu, misu).

parinte(misu, roco).



str1(X, Z) :- parinte(X, Z).

str1(X, Z) :- parinte(X, Y), str1(Y, Z).



Prolog – exemplu

% Se schimba ordinea regulilor:

str2(X, Z) :- parinte(X, Y), str2(Y, Z).

str2(X, Z) :- parinte(X, Z).

% Se schimba ordinea scopurilor in prima varianta:

str3(X, Z) :- parinte(X, Z).

str3(X, Z) :- str3(X, Y), parinte(Y, Z).

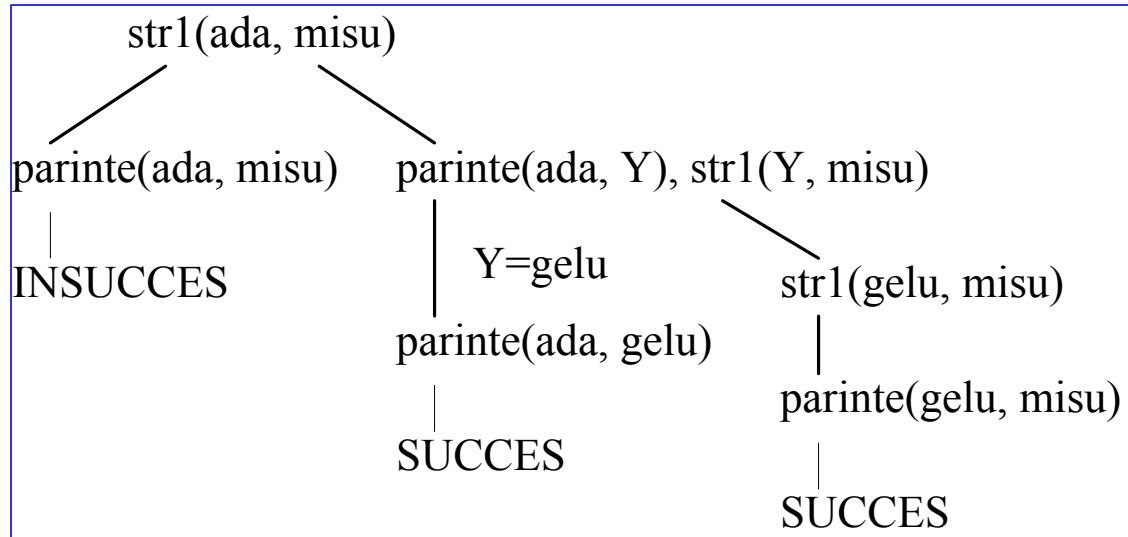
% Se schimba atat ordinea regulilor, cat si ordinea scopurilor:

str4(X, Z) :- str4(X, Y), parinte(Y, Z).

str4(X, Z) :- parinte(X, Z).

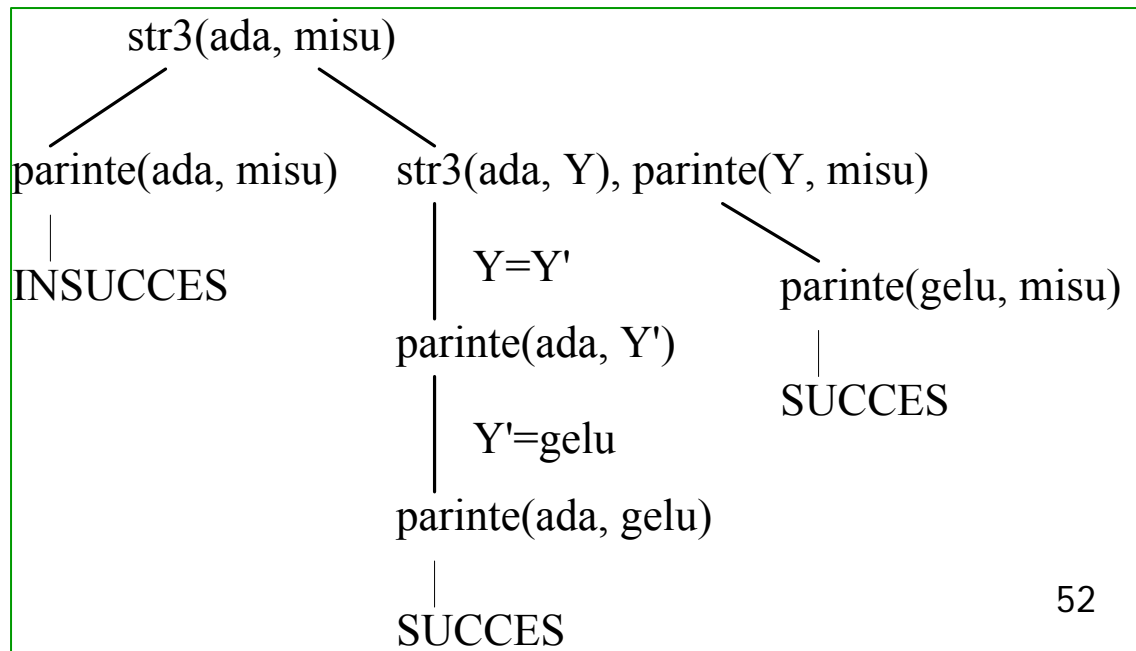
Prolog - exemplu

?- **str1(ada, misu).**
yes



?- **str2(ada, misu).**
yes

?- **str3(ada, misu).**
yes



Prolog – exemplu

?- str4(ada, misu).

?- str3(mia, roco).

```
str4(ada, misu)
|
str4(ada, Y), parinte(Y, misu)
|
str4(ada, Y'), parinte(Y', Y)
|
str4(ada, Y''), parinte(Y'', Y)
|
...
arbore infinit
```

