

# PARADIGME DE PROGRAMARE

---

## Curs 9

Logica propozițională. Logica cu predicate de ordinul întâi.

# Context

**Scop:** modelarea raționamentelor logice ca procese de calcul efectuate pe mașini de calcul

## Abordare

- Descrierea proprietăților obiectelor din universul problemei într-un limbaj neambiguu
  - Sintactic
  - Semantic
- Reguli pentru deducerea (calculul) de noi proprietăți din cele existente

## Formalisme logice

- Logica propozițională
- Logica cu predicate de ordinul întâi

# Logica propozițională – Cuprins

- **Ontologie**
- Sintaxă
- Semantică
- Satisfiabilitate și validitate
- Derivabilitate
- Decidabilitate
- Inferență și reguli de inferență
- Demonstrație
- Rezoluție

# Ontologie

- Universul problemei este descris prin propoziții declarative, numite **fapte**

## **Exemple**

*Iepurii sunt mai rapizi decât broaștele țestoase.*

*Bugs e iepure.*

*Leo e broască țestoasă.*

*Bugs e mai rapid decât Leo.*

- **Faptele nu sunt structurate** astfel încât să surprindă obiectele din acest univers și relațiile dintre ele
  - În logica propozițională, ultima propoziție nu este o consecință logică a celorlalte, pentru că logica propozițională nu poate modela relațiile dintre aceste fapte

# Logica propozițională – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Satisfiabilitate și validitate
- Derivabilitate
- Decidabilitate
- Inferență și reguli de inferență
- Demonstrație
- Rezoluție

# Sintaxă

**Expresie în logica propozițională** = propoziție care poate fi adevărată sau falsă

**propoziție** = propoziție simplă (atomică) | propoziție compusă  
(*ex: Bugs e iepure.*) | (*ex: Bugs e iepure și Leo e broască țestoasă.*)  
(*notație: p, q, r, etc.*) | (*notație:  $\neg P, P \wedge Q, etc.$* )

**propoziție compusă** = negație | conjuncție | disjuncție | implicație | echivalență  
( $\neg P$ ) | ( $P \wedge Q$ ) | ( $P \vee Q$ ) | ( $P \Rightarrow Q$ ) | ( $P \Leftrightarrow Q$ )

# Logica propozițională – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Satisfiabilitate și validitate
- Derivabilitate
- Decidabilitate
- Inferență și reguli de inferență
- Demonstrație
- Rezoluție

# Semantică

**Interpretare** = mulțime de asocieri între fiecare propoziție atomică din limbaj și o valoare de adevăr

- **Exemplu:** Interpretarea I = { p = true, q = true, r = false }  
Interpretarea J = { p = false, q = true, r = false }
- Valoarea de adevăr a unei propoziții se stabilește
  - În raport cu o interpretare (ex:  $p \wedge q = \text{true}$  în interpretarea I /  $p \wedge q = \text{false}$  în interpretarea J)
  - Conform unor **reguli semantice** care determină valoarea de adevăr a unei propoziții compuse din valorile de adevăr ale propozițiilor componente



# Reguli semantice pentru propoziții compuse

$\neg P_1 = \text{true}$

dacă și numai dacă

$P_1 = \text{false}$

$P_1 \wedge P_2 = \text{true}$

dacă și numai dacă

$P_1 = \text{true}$

și

$P_2 = \text{true}$

$P_1 \vee P_2 = \text{true}$

dacă și numai dacă

$P_1 = \text{true}$

sau

$P_2 = \text{true}$

$P_1 \Rightarrow P_2 = \text{true}$

dacă și numai dacă

$P_1 = \text{false}$

sau

$P_2 = \text{true}$

$P_1 \Leftrightarrow P_2 = \text{true}$

dacă și numai dacă

$P_1 = P_2$

# Evaluare

**Evaluarea unei expresii (propoziții)** = determinarea valorii de adevăr a propoziției, într-o interpretare, prin aplicarea regulilor semantice

## Exemplu

$E = p \wedge (q \vee r)$                       în interpretarea  $I = \{ p = \text{true}, q = \text{true}, r = \text{false} \}$

$E = \text{true} \wedge (\text{true} \vee \text{false}) = \text{true} \wedge \text{true} = \text{true}$

# Logica propozițională – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Satisfiabilitate și validitate
- Derivabilitate
- Decidabilitate
- Inferență și reguli de inferență
- Demonstrație
- Rezoluție

# Satisfiabilitate

Propoziția P este **satisfiabilă**  $\Leftrightarrow$  P este adevărată (true) în cel puțin o interpretare

Propoziția P este **nesatisfiabilă**  $\Leftrightarrow$  P este falsă (false) în toate interpretările

- Satisfiabilitatea unei propoziții se determină folosind **tabele de adevăr**

## Exemple

- $p \wedge (q \vee r)$  satisfiabilă conform tabelii alăturate (întrucât este adevărată în 3 interpretări)
- $p \wedge \neg p$  nesatisfiabilă (se determină similar)

p	q	r	$p \wedge (q \vee r)$
true	true	true	true
true	true	false	true
true	false	true	true
true	false	false	false
false	true	true	false
false	true	false	false
false	false	true	false
false	false	false	false

# Validitate

Propoziția P este **validă (tautologie)**  $\Leftrightarrow$  P este adevărată (true) în toate interpretările

Propoziția P este **invalidă**  $\Leftrightarrow$  P este falsă (false) în cel puțin o interpretare

- Validitatea unei propoziții se determină folosind **tabele de adevăr**

## Exemple

- $(p \vee q) \vee (\neg q \vee r)$  validă conform tabelii alăturate (întrucât este adevărată în toate interpretările)
- $p \wedge (q \vee r)$  invalidă conform paginii anterioare (întrucât este falsă în 5 interpretări)

p	q	r	$(p \vee q) \vee (\neg q \vee r)$
true	true	true	true
true	true	false	true
true	false	true	true
true	false	false	true
false	true	true	true
false	true	false	true
false	false	true	true
false	false	false	true

# Logica propozițională – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Satisfiabilitate și validitate
- **Derivabilitate**
- Decidabilitate
- Inferență și reguli de inferență
- Demonstrație
- Rezoluție

# Derivabilitate

Mulțimea de propoziții  $\Delta$  **derivă** propoziția  $P$  ( $\Delta \models P$ )  $\Leftrightarrow$  orice interpretare care satisface toate propozițiile din  $\Delta$  satisface și propoziția  $P$

- Derivabilitatea unei propoziții se determină folosind **tabele de adevăr**

## Exemplu

- $\{p, p \Rightarrow q\} \models q$  conform tabelii alăturate  
(Singura interpretare care satisface  $p$  și  $p \Rightarrow q$  este prima, și prima interpretare satisface și  $q$ )

p	q	$p \Rightarrow q$
true	true	true
true	false	false
false	true	true
false	false	true

# Logica propozițională – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Satisfiabilitate și validitate
- Derivabilitate
- Decidabilitate
- Inferență și reguli de inferență
- Demonstrație
- Rezoluție



# Decidabilitate

Logica propozițională este **decidabilă**: există o procedură care se termină pentru a decide dacă o propoziție este validă

- $n$  propoziții atomice  $\Rightarrow 2^n$  interpretări
  - Pentru fiecare interpretare se evaluează propoziția și dacă rezultatul este mereu true, atunci propoziția este validă
  - Întrucât trebuie să trecem prin  $2^n$  interpretări, problema determinării validității unei propoziții este **NP-completă**
- $\Rightarrow$  Pentru eficiență, se vor prefera metode bazate pe **reguli de inferență** (în dauna celor bazate pe tabele de adevăr)

# Logica propozițională – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Satisfiabilitate și validitate
- Derivabilitate
- Decidabilitate
- Inferență și reguli de inferență
- Demonstrație
- Rezoluție

# Inferență

**Inferență** = derivarea prin calcul a concluziilor unei mulțimi de premise

## Regulă de inferență

$$P_1, P_2, \dots, P_n$$
$$\frac{\quad}{C} \text{ (id\_regulă)}$$

$P_i$  = premise (propoziții considerate adevărate)

$C$  = concluzie (propoziție care derivă din mulțimea de premise)

## Notatii

$\Delta \vDash C$  - derivabilitate logică

$\Delta \vdash_{\text{id\_regulă}} C$  - derivabilitate mecanică (folosind regula de inferență *id\_regulă*)

# Reguli de inferență

$$\frac{A \Rightarrow B}{\frac{A}{B}} \text{ (ModusPonens)}$$

$$\frac{A \Rightarrow B}{\frac{\neg B}{\neg A}} \text{ (ModusTollens)}$$

$$\frac{A \vee B}{\frac{\neg A \vee C}{B \vee C}} \text{ (Rezoluție)}$$

$$\frac{}{A \vee \neg A} \text{ (AxiomaA)}$$

Regulile fără premise se numesc **axiome**

# Proprietăți ale regulilor de inferență

- **Consistență:** regula determină doar concluzii care derivă logic din premise

Dacă  $\Delta \vdash_{\text{id\_regulă}} C$ , atunci  $\Delta \models C$

- **Completitudine:** regula determină toate concluziile care derivă logic din premise

Dacă  $\Delta \models C$ , atunci  $\Delta \vdash_{\text{id\_regulă}} C$

- O regulă de inferență consistentă și completă ar fi suficientă pentru a deduce toate adevărurile din universul problemei (nici mai multe, nici mai puține)

# Proprietăți Modus Ponens

- **Consistentă** (vezi tabela de adevăr)
- **Incompletă** (vezi exemplul de mai jos)

$$\frac{A \Rightarrow B}{A} \text{ (Modus Ponens)}$$

## Exemplu

$\{p \Rightarrow q, q \Rightarrow r\} \models p \Rightarrow r$  (se poate verifica prin tabela de adevăr)

$\{p \Rightarrow q, q \Rightarrow r\} \not\models_{\text{Modus Ponens}} p \Rightarrow r$  (nu am informație despre vreo propoziție atomică)

**Consecință:** Modus Ponens se folosește în demonstrații împreună cu un set de axiome

# Logica propozițională – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Satisfiabilitate și validitate
- Derivabilitate
- Decidabilitate
- Inferență și reguli de inferență
- **Demonstrație**
- Rezoluție

# Demonstrație

**Teoremă** = propoziție care se dorește demonstrată

**Demonstrație (a unei teoreme)** = secvență de propoziții adevărate, încheiată cu teorema, conținând:

- Premise
- Instance ale axiomelor
- Rezultate ale aplicării regulilor de inferență asupra propozițiilor anterioare în secvență

**Demonstrație prin reducere la absurd** = secvență de propoziții considerate adevărate, încheiată cu o propoziție nesatisfiabilă, conținând

- Aceleași elemente de mai sus +
- La premise se adaugă  $\neg C$  (negația teoremei, singurul posibil neadevăr, cauza pentru care o propoziție falsă a fost determinată ca fiind adevărată)



# Demonstrație cu Modus Ponens - Exemplu

Să se demonstreze  $\{p \Rightarrow q, q \Rightarrow r\} \vdash p \Rightarrow r$ , folosind regulile de inferență:

$$\frac{A \Rightarrow B}{\frac{A}{B}} \text{ (ModusPonens)}$$

$$\frac{}{A \Rightarrow (B \Rightarrow A)} \text{ (IntroImp)}$$

Axiome

$$\frac{}{(A \Rightarrow (B \Rightarrow C)) \Rightarrow ((A \Rightarrow B) \Rightarrow (A \Rightarrow C))} \text{ (DistriImp)}$$

1.  $p \Rightarrow q$
2.  $q \Rightarrow r$
3.  $(q \Rightarrow r) \Rightarrow (p \Rightarrow (q \Rightarrow r))$
4.  $p \Rightarrow (q \Rightarrow r)$
5.  $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$
6.  $(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$
7.  $p \Rightarrow r$

- (Premisă)  
 (Premisă)  
 (IntroImp)  
 (ModusPonens 2,3)  
 (DistriImp)  
 (ModusPonens 4,5)  
 (ModusPonens 1,6)

# Strategii de control

- În timpul unei demonstrații, se întâmplă să existe mai multe reguli aplicabile simultan  
**Strategia de control** dictează care reguli au prioritate în asemenea situații.

## Strategii de control uzuale

- **Forward chaining**
  - Se pleacă de la premise către scop (ceea ce trebuie demonstrat)
  - Se derivează toate concluziile posibile (în orice ordine)
  - Oprește la satisfacerea scopului
- **Backward chaining**
  - Se pleacă de la scop către premise
  - Se folosesc doar regulile care pot contribui la satisfacerea scopului
  - Premisele acestor reguli devin noi scopuri, etc.

# Logica propozițională – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Satisfiabilitate și validitate
- Derivabilitate
- Decidabilitate
- Inferență și reguli de inferență
- Demonstrație
- Rezoluție

# Proprietăți Rezoluție

- **Consistentă** (vezi tabela de adevăr)
- **Completă** (demonstrația nu face obiectul cursului)
  - în sensul că pot determina valoarea de adevăr a unei propoziții date, dar nu pot genera toate propozițiile adevărate)

$$\frac{A \vee B \quad \neg A \vee C}{B \vee C} \text{ (Rezoluție)}$$

**Consecință:** Un demonstrator de teoreme se poate baza pe rezoluție.

**Observație:** Pentru a folosi rezoluția în demonstrații, propozițiile sunt aduse la o formă normală numita **forma clauzală** (acest lucru este mereu posibil).

# Forma clauzală a unei propoziții

**Literal** =  $p \mid \neg p$  (propoziții atomice sau negațiile lor)

**Clauză** = mulțime de literali aflați în relația de disjuncție

- $\{p, q, \neg r\}$  echivalent cu  $p \vee q \vee \neg r$

**Clauză Horn** = clauză în care un singur literal este nenegat

- $\{p, \neg q, \neg r\}$  echivalent cu  $p \vee \neg q \vee \neg r$  și cu  $q \wedge r \Rightarrow p$

**Propoziție în formă clauzală** = mulțime de clauze aflate în relația de conjuncție

- $\{\{p\}, \{q, \neg r\}, \{\neg p, r\}\}$  echivalent cu  $p \wedge (q \vee \neg r) \wedge (\neg p \vee r)$

# Rezoluție în forma clauzală

## Forma simplă

$$\frac{\begin{array}{l} \{p, q\} \\ \{\neg p, r\} \end{array}}{\{q, r\}} \text{ (Rezoluție)}$$

## Forma generală

$$\frac{\begin{array}{l} \{q_1, \dots, p, \dots, q_m\} \\ \{r_1, \dots, \neg p, \dots, r_n\} \end{array}}{\{q_1 \dots q_m, r_1 \dots r_n\}} \text{ (Rezoluție)}$$

## Contradicție

$$\frac{\begin{array}{l} \{p\} \\ \{\neg p\} \end{array}}{\{\}} \text{ (Rezoluție)}$$

← Rezolvent vid: indică existența unei contradicții între premise

# Cazuri particulare ale Rezoluției

## Modus Ponens

$\{\neg p, q\}$

$\frac{\{p\}}{\{q\}}$  (Rezoluție)

$A \Rightarrow B$

$\frac{A}{B}$  (Modus Ponens)

## Modus Tollens

$\{\neg p, q\}$

$\frac{\{\neg q\}}{\{\neg p\}}$  (Rezoluție)

$A \Rightarrow B$

$\frac{\neg B}{\neg A}$  (Modus Tollens)

## Tranzitivitatea implicației

$\{\neg p, q\}$

$\frac{\{\neg q, r\}}{\{\neg p, r\}}$  (Rezoluție)

$A \Rightarrow B$

$B \Rightarrow C$

$A \Rightarrow C$

(TranziImp)

Diferența de notație (A, B vs. p, q) indică faptul că în cazul formei cluzale lucrăm doar cu propoziții atomice și negațiile lor

# Demonstrații bazate pe Rezoluție

Fie  $P$  o propoziție în formă clauzală. Pentru a demonstra

- $P$  nesatisfiabilă
- $P$  derivabilă din premise
- $P$  validă



# Demonstrații bazate pe Rezoluție

Fie  $P$  o propoziție în formă clauzală. Pentru a demonstra

- **$P$  nesatisfiabilă**
  - Pornesc cu premisa  $P$
  - Derivez clauza vidă
- **$P$  derivabilă din premise**
- **$P$  validă**

# Demonstrații bazate pe Rezoluție

Fie  $P$  o propoziție în formă clauzală. Pentru a demonstra

- **$P$  nesatisfiabilă**
  - Pornesc cu premisa  $P$
  - Derivez clauza vidă
- **$P$  derivabilă din premise**
  - Introduc  $\neg P$  în premise ← reducere la absurd
  - Derivez clauza vidă
- **$P$  validă**

# Demonstrații bazate pe Rezoluție

Fie  $P$  o propoziție în formă clauzală. Pentru a demonstra

- **$P$  nesatisfiabilă**
  - Pornesc cu premisa  $P$
  - Derivez clauza vidă
- **$P$  derivabilă din premise**
  - Introduc  $\neg P$  în premise ← reducere la absurd
  - Derivez clauza vidă
- **$P$  validă**
  - Pornesc cu premisa  $\neg P$
  - Derivez clauza vidă

Se observă că nu avem o metodă de a demonstra satisfiabilitatea / invaliditatea

# Demonstrație cu Rezoluție – Exemplu

Să se demonstreze  $\{p \Rightarrow q, q \Rightarrow r\} \vdash p \Rightarrow r$ , folosind doar Rezoluția.

## Demonstrație prin reducere la absurd

- Introduc în premise  $\neg(p \Rightarrow r) = \neg(\neg p \vee r) = p \wedge \neg r = \{ \{p\}, \{\neg r\} \}$

1.  $\{\neg p, q\}$  (Premisă)
2.  $\{\neg q, r\}$  (Premisă)
3.  $\{p\}$  (Concluzie negată)
4.  $\{\neg r\}$  (Concluzie negată)
5.  $\{q\}$  (Rezoluție 1,3)
6.  $\{r\}$  (Rezoluție 2,5)
7.  $\{\}$  (Rezoluție 4,6)

# Logica cu predicate de ordinul întâi – Cuprins

- **Ontologie**
- Sintaxă
- Semantică
- Exemple
- Decidabilitate
- Forma clauzală
- Algoritm de aducere la forma clauzală
- Unificare

# Ontologie

- Universul problemei este descris prin **obiecte, proprietățile lor și relațiile** dintre ele

## Exemple

$\forall X, Y. ((iepure(X) \wedge \text{testoasă}(Y)) \Rightarrow mai\_rapid(X, Y)).$

$iepure(bugs).$

$\text{testoasă}(leo).$

$mai\_rapid(bugs, leo).$

*Iepurii sunt mai rapizi decât broaștele țestoase.*

*Bugs e iepure.*

*Leo e broască țestoasă.*

*Bugs e mai rapid decât Leo.*

- **Propozițiile sunt structurate** astfel încât să surprindă obiectele din acest univers și relațiile dintre ele
  - În logica propozițională, ultima propoziție nu era o consecință logică a celorlalte
  - În logica cu predicate de ordinul întâi este

# Logica cu predicate de ordinul întâi – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Exemple
- Decidabilitate
- Forma clauzală
- Algoritm de aducere la forma clauzală
- Unificare

# Elemente de sintaxă

- **Constante:** athos, porthos, aramis (cu litere mici)
- **Variabile:** Mușchetar, Cardinal (cu litere mari)
- **Funcții:** regină(franța) (calculează o constantă)
- **Predicate:** respectă(porthos,athos)
- **Conective:**  $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$  (după cum scade prioritatea) } Asociativi la dreapta
- **Cuantificatori:**  $\forall, \exists$
- **Egalitate:** =



# Sintaxă

**termen** = constantă | variabilă | funcție(termen,... termen)

**propoziție atomică** = predicat(termen,... termen) | termen = termen

**propoziție** = propoziție atomică | (propoziție) |  $\neg$ propoziție |  
propoziție conectivă\_binară propoziție | cuantificator  $var_1 \dots var_n$ .propoziție

- Termenii reprezintă obiecte
- Propozițiile reprezintă proprietăți și relații

# Logica cu predicate de ordinul întâi – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Exemple
- Decidabilitate
- Forma clauzală
- Algoritm de aducere la forma clauzală
- Unificare

# Semantică

**Interpretare** = pereche  $(D, A)$ , unde

- $D$  = domeniu (specifică obiectele din problemă)
- $A$  = mulțime de asocieri pentru termeni și propoziții
  - Constantelor și funcțiilor li se asociază câte un obiect din  $D$  ( $c \in D, f: D^n \rightarrow D$ )
  - Predicatelor li se asociază o valoare de adevăr (true sau false) ( $p: D^n \rightarrow \{\text{true}, \text{false}\}$ )

## Reguli semantice

Cele de la logica propozițională (pentru conective) +

$\forall X.\text{prop} = \text{true}$     dacă și numai dacă    nu există  $d \in D$  a.î.  $\text{prop}_{[d/X]} = \text{false}$

$\exists X.\text{prop} = \text{true}$     dacă și numai dacă    există  $d \in D$  a.î.  $\text{prop}_{[d/X]} = \text{true}$

# Logica cu predicate de ordinul întâi – Cuprins

- Ontologie
- Sintaxă
- Semantică
- **Exemple**
- Decidabilitate
- Forma clauzală
- Algoritm de aducere la forma clauzală
- Unificare

# Exemple



1. Vrăbia mălai visează.
2. Unele vrăbii visează mălai.
3. Nu toate vrăbiile visează mălai.
4. Nicio vrăbie nu visează mălai.
5. Numai vrăbiile visează mălai.
6. Toate și numai vrăbiile visează mălai.
7. Vivi este o vrăbie care visează mălai.

# Exemple



1. Vrăbia mălai visează.  
 $\forall X \bullet (vrăbie(X) \Rightarrow visează(X, mălai))$
2. Unele vrăbii visează mălai.
3. Nu toate vrăbiile visează mălai.
4. Nicio vrăbie nu visează mălai.
5. Numai vrăbiile visează mălai.
6. Toate și numai vrăbiile visează mălai.
7. Vivi este o vrăbie care visează mălai.

# Exemple



1. Vrăbia mălai visează.  
 $\forall X \bullet (vrăbie(X) \Rightarrow visează(X, mălai))$
2. Unele vrăbii visează mălai.  
 $\exists X \bullet (vrăbie(X) \wedge visează(X, mălai))$
3. Nu toate vrăbiile visează mălai.
4. Nicio vrăbie nu visează mălai.
5. Numai vrăbiile visează mălai.
6. Toate și numai vrăbiile visează mălai.
7. Vivi este o vrăbie care visează mălai.

# Exemple



1. Vrăbia mălai visează.  
 $\forall X \bullet (\text{vrăbie}(X) \Rightarrow \text{visează}(X, \text{mălai}))$
2. Unele vrăbii visează mălai.  
 $\exists X \bullet (\text{vrăbie}(X) \wedge \text{visează}(X, \text{mălai}))$
3. Nu toate vrăbiile visează mălai.  
 $\neg(\forall X \bullet (\text{vrăbie}(X) \Rightarrow \text{visează}(X, \text{mălai})))$  sau  $\exists X \bullet (\text{vrăbie}(X) \wedge \neg \text{visează}(X, \text{mălai}))$
4. Nicio vrăbie nu visează mălai.
5. Numai vrăbiile visează mălai.
6. Toate și numai vrăbiile visează mălai.
7. Vivi este o vrăbie care visează mălai.



# Exemple



1. Vrăbia málai visează.  
 $\forall X \bullet (\text{vrăbie}(X) \Rightarrow \text{visează}(X, \text{málai}))$
2. Unele vrăbii visează málai.  
 $\exists X \bullet (\text{vrăbie}(X) \wedge \text{visează}(X, \text{málai}))$
3. Nu toate vrăbiile visează málai.  
 $\neg(\forall X \bullet (\text{vrăbie}(X) \Rightarrow \text{visează}(X, \text{málai})))$  sau  $\exists X \bullet (\text{vrăbie}(X) \wedge \neg \text{visează}(X, \text{málai}))$
4. Nicio vrăbie nu visează málai.  
 $\neg(\exists X \bullet (\text{vrăbie}(X) \wedge \text{visează}(X, \text{málai})))$  sau  $\forall X \bullet (\text{vrăbie}(X) \Rightarrow \neg \text{visează}(X, \text{málai}))$
5. Numai vrăbiile visează málai.
6. Toate și numai vrăbiile visează málai.
7. Vivi este o vrăbie care visează málai.

# Exemple



1. Vrăbia mălai visează.  
 $\forall X \bullet (vrăbie(X) \Rightarrow visează(X, mălai))$
2. Unele vrăbii visează mălai.  
 $\exists X \bullet (vrăbie(X) \wedge visează(X, mălai))$
3. Nu toate vrăbiile visează mălai.  
 $\neg(\forall X \bullet (vrăbie(X) \Rightarrow visează(X, mălai)))$  sau  $\exists X \bullet (vrăbie(X) \wedge \neg visează(X, mălai))$
4. Nicio vrăbie nu visează mălai.  
 $\neg(\exists X \bullet (vrăbie(X) \wedge visează(X, mălai)))$  sau  $\forall X \bullet (vrăbie(X) \Rightarrow \neg visează(X, mălai))$
5. Numai vrăbiile visează mălai.  
 $\forall X \bullet (visează(X, mălai) \Rightarrow vrăbie(X))$
6. Toate și numai vrăbiile visează mălai.
7. Vivi este o vrăbie care visează mălai.

# Exemple



1. Vrăbia mălai visează.  
 $\forall X \bullet (vrăbie(X) \Rightarrow visează(X, mălai))$
2. Unele vrăbii visează mălai.  
 $\exists X \bullet (vrăbie(X) \wedge visează(X, mălai))$
3. Nu toate vrăbiile visează mălai.  
 $\neg(\forall X \bullet (vrăbie(X) \Rightarrow visează(X, mălai)))$  sau  $\exists X \bullet (vrăbie(X) \wedge \neg visează(X, mălai))$
4. Nicio vrăbie nu visează mălai.  
 $\neg(\exists X \bullet (vrăbie(X) \wedge visează(X, mălai)))$  sau  $\forall X \bullet (vrăbie(X) \Rightarrow \neg visează(X, mălai))$
5. Numai vrăbiile visează mălai.  
 $\forall X \bullet (visează(X, mălai) \Rightarrow vrăbie(X))$
6. Toate și numai vrăbiile visează mălai.  
 $\forall X \bullet (visează(X, mălai) \Leftrightarrow vrăbie(X))$
7. Vivi este o vrăbie care visează mălai.

# Exemple



1. Vrăbia málai visează.  
 $\forall X \bullet (vrăbie(X) \Rightarrow visează(X, málai))$
2. Unele vrăbii visează málai.  
 $\exists X \bullet (vrăbie(X) \wedge visează(X, málai))$
3. Nu toate vrăbiile visează málai.  
 $\neg(\forall X \bullet (vrăbie(X) \Rightarrow visează(X, málai)))$  sau  $\exists X \bullet (vrăbie(X) \wedge \neg visează(X, málai))$
4. Nicio vrăbie nu visează málai.  
 $\neg(\exists X \bullet (vrăbie(X) \wedge visează(X, málai)))$  sau  $\forall X \bullet (vrăbie(X) \Rightarrow \neg visează(X, málai))$
5. Numai vrăbiile visează málai.  
 $\forall X \bullet (visează(X, málai) \Rightarrow vrăbie(X))$
6. Toate și numai vrăbiile visează málai.  
 $\forall X \bullet (visează(X, málai) \Leftrightarrow vrăbie(X))$
7. Vivi este o vrăbie care visează málai.  
 $vrăbie(vivi) \wedge visează(vivi, málai)$

# Greșeli frecvente

- În general:
  - $\forall$  se folosește împreună cu  $\Rightarrow$
  - $\exists$  se folosește împreună cu  $\wedge$
- $\forall X \cdot (\text{vrabie}(X) \wedge \text{visează}(X, \text{mălai}))$   
?
- $\exists X \cdot (\text{vrabie}(X) \Rightarrow \text{visează}(X, \text{mălai}))$   
?

# Greșeli frecvente

- În general:
  - $\forall$  se folosește împreună cu  $\Rightarrow$
  - $\exists$  se folosește împreună cu  $\wedge$
- $\forall X \cdot (\text{vrabie}(X) \wedge \text{visează}(X, \text{mălai}))$   
înseamnă că toată lumea e vrabie și toată lumea visează mălai
- $\exists X \cdot (\text{vrabie}(X) \Rightarrow \text{visează}(X, \text{mălai}))$   
este o propoziție adevărată și dacă există cineva care nu e vrabie

# Proprietăți cuantificatori și conective

- **(Ne)Comutativitate**

$\forall X.\forall Y$  este totuna cu  $\forall Y.\forall X$  și se scrie prescurtat  $\forall X,Y.$

$\exists X.\exists Y$  este totuna cu  $\exists Y.\exists X$  și se scrie prescurtat  $\exists X,Y.$

$\exists X.\forall Y$  NU este totuna cu  $\forall Y.\exists X$

- $\exists X.\forall Y.\text{visează}(X,Y)$  înseamnă că .....
- $\forall Y.\exists X.\text{visează}(X,Y)$  înseamnă că .....

- **Dualitate (se pot scrie unul/una în funcție de altul/alta)**

$$\forall X.p = \neg(\exists X.\neg p) \qquad p \vee q = \neg(\neg p \wedge \neg q)$$

$$\exists X.p = \neg(\forall X.\neg p) \qquad p \wedge q = \neg(\neg p \vee \neg q)$$

$$\neg(\forall X.p) = \exists X.\neg p \qquad \neg(p \vee q) = \neg p \wedge \neg q$$

$$\neg(\exists X.p) = \forall X.\neg p \qquad \neg(p \wedge q) = \neg p \vee \neg q$$

# Proprietăți cuantificatori și conective

- **(Ne)Comutativitate**

$\forall X.\forall Y$  este totuna cu  $\forall Y.\forall X$  și se scrie prescurtat  $\forall X,Y.$

$\exists X.\exists Y$  este totuna cu  $\exists Y.\exists X$  și se scrie prescurtat  $\exists X,Y.$

$\exists X.\forall Y$  NU este totuna cu  $\forall Y.\exists X$

- $\exists X.\forall Y.\text{visează}(X,Y)$  înseamnă că există cineva care visează la toată lumea
- $\forall Y.\exists X.\text{visează}(X,Y)$  înseamnă că la oricine visează măcar cineva

- **Dualitate (se pot scrie unul/una în funcție de altul/alta)**

$$\forall X.p = \neg(\exists X.\neg p) \qquad p \vee q = \neg(\neg p \wedge \neg q)$$

$$\exists X.p = \neg(\forall X.\neg p) \qquad p \wedge q = \neg(\neg p \vee \neg q)$$

$$\neg(\forall X.p) = \exists X.\neg p \qquad \neg(p \vee q) = \neg p \wedge \neg q$$

$$\neg(\exists X.p) = \forall X.\neg p \qquad \neg(p \wedge q) = \neg p \vee \neg q$$



# Definiții moștenite de la logica propozițională

- **Evaluare:** determinarea valorii de adevăr a lui  $P$  (într-o interpretare) prin aplicarea regulilor semantice
- **Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare
- **Validitate:**  $P = \text{true}$  în toate interpretările
- **Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele
- **Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise
- **Demonstrație:** secvență de propoziții adevărate încheiată cu propoziția care se dorea demonstrată (teorema)

# Logica cu predicate de ordinul întâi – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Exemple
- Decidabilitate
- Forma clauzală
- Algoritm de aducere la forma clauzală
- Unificare

# Decidabilitate

Logica cu predicate de ordinul întâi este **semidecidabilă**

- O infinitate de domenii posibile  $\Rightarrow$  o infinitate de interpretări (nu le putem enumera)
  - Se poate demonstra validitatea (folosind, de exemplu, Rezoluția)
  - Nu se poate demonstra invaliditatea
- Compromis între decidabilitate și puterea de reprezentare
  - **Logica propozițională:** decidabilă / putere de reprezentare redusă
  - **Logica cu predicate de ordinul întâi:** semidecidabilă / putere de reprezentare sporită

# Logica cu predicate de ordinul întâi – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Exemple
- Decidabilitate
- Forma clauzală
- Algoritm de aducere la forma clauzală
- Unificare

# Forma clauzală a unei propoziții

## Literal, Clauză, Clauză Horn, Propoziție în formă clauzală

- se definesc ca în cazul logicii propoziționale  
(doar că diferă ceea ce se înțelege prin propoziție atomică)

## Exemplu de clauză

{ prieten(X,porthos),  $\neg$ fricos(X) } echivalent cu  $\text{prieten}(X,\text{porthos}) \vee \neg\text{fricos}(X)$

**Observație:** forma clauzală se mai numește și **forma normal conjunctivă**

# Logica cu predicate de ordinul întâi – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Exemple
- Decidabilitate
- Forma clauzală
- Algoritm de aducere la forma clauzală
- Unificare

# Algoritm de aducere la forma clauzală (I)

1. **Elimină implicațiile**

$a \Rightarrow b$  devine  $\neg a \vee b$

2. **Mută negațiile spre interior**

$\neg(a \vee b) = \neg a \wedge \neg b$

$\neg(a \wedge b) = \neg a \vee \neg b$

$\neg(\forall X.p) = \exists X.\neg p$

$\neg(\exists X.p) = \forall X.\neg p$

$\neg(\neg p) = p$

3. **Redenumeste variabilele cuantificate a.î. să nu existe cuantificatori diferiți care folosesc același nume de variabilă**

(ex:  $\forall X.p(X) \wedge \forall X.q(X) \vee \exists X.r(X)$  devine  $\forall X.p(X) \wedge \forall Y.q(Y) \vee \exists Z.r(Z)$ )

4. **Forma prenex: Mută toți cuantificatorii la începutul propoziției, în aceeași ordine**

(ex:  $\forall X.p(X) \wedge \forall Y.q(Y) \vee \exists Z.r(Z)$  devine  $\forall X.\forall Y.\exists Z.(p(X) \wedge q(Y) \vee r(Z))$ )

# Algoritm de aducere la forma clauzală (II)

- Skolemizare: Elimină cuantificatorii existențiali prin înlocuire cu**
  - O **constantă** - dacă acest  $\exists$  nu era precedat de niciun  $\forall$
  - O **funcție** de toate variabilele cuantificate universal care îl precedau - altfel  
(ex:  $\forall X \cdot \forall Y \cdot \exists Z \cdot (p(X) \wedge q(Y) \vee r(Z))$  devine  $\forall X \cdot \forall Y \cdot (p(X) \wedge q(Y) \vee r(f(X, Y)))$ )
- Elimină cuantificatorii universalii (ce rămâne e automat cuantificat universal)**  
(ex:  $\forall X \cdot \forall Y \cdot (p(X) \wedge q(Y) \vee r(f(X, Y)))$  devine  $p(X) \wedge q(Y) \vee r(f(X, Y))$ )
- Distribuie  $\vee$  prin  $\wedge$**   
 $(a \wedge b) \vee c$  devine  $(a \vee c) \wedge (b \vee c)$
- Transformă expresiile în clauze**



# Aducere la forma clauzală – Exemplu (I)

“Oricine rezolvă toate laboratoarele este apreciat de cineva.”

$$\forall X.(\forall Y.(\text{laborator}(Y) \Rightarrow \text{rezolvă}(X,Y)) \Rightarrow \exists Y.\text{apreciază}(Y,X))$$

1. **Elimină implicațiile**

$$\forall X.(\neg(\forall Y. \neg \text{laborator}(Y) \vee \text{rezolvă}(X,Y)) \vee \exists Y.\text{apreciază}(Y,X))$$

2. **Mută negațiile spre interior**

$$\forall X.(\exists Y. \neg(\neg \text{laborator}(Y) \vee \text{rezolvă}(X,Y)) \vee \exists Y.\text{apreciază}(Y,X))$$

$$\forall X.(\exists Y.(\text{laborator}(Y) \wedge \neg \text{rezolvă}(X,Y)) \vee \exists Y.\text{apreciază}(Y,X))$$

3. **Redenumeste variabilele cuantificate**

$$\forall X.(\exists Y.(\text{laborator}(Y) \wedge \neg \text{rezolvă}(X,Y)) \vee \exists Z.\text{apreciază}(Z,X))$$

4. **Mută cuantorii la început**

$$\forall X.\exists Y.\exists Z.((\text{laborator}(Y) \wedge \neg \text{rezolvă}(X,Y)) \vee \text{apreciază}(Z,X))$$

# Aducere la forma clauzală – Exemplu (II)

“Oricine rezolvă toate laboratoarele este apreciat de cineva.”

$$\forall X. \exists Y. \exists Z. ((\text{laborator}(Y) \wedge \neg \text{rezolvă}(X, Y)) \vee \text{apreciază}(Z, X))$$

5. **Elimină cuantificatorii existențiali**

$$\forall X. ((\text{laborator}(f_y(X)) \wedge \neg \text{rezolvă}(X, f_y(X))) \vee \text{apreciază}(f_z(X), X))$$

6. **Elimină cuantificatorii universali**

$$(\text{laborator}(f_y(X)) \wedge \neg \text{rezolvă}(X, f_y(X))) \vee \text{apreciază}(f_z(X), X)$$

7. **Distribuie  $\vee$  prin  $\wedge$**

$$(\text{laborator}(f_y(X)) \vee \text{apreciază}(f_z(X), X)) \wedge (\neg \text{rezolvă}(X, f_y(X)) \vee \text{apreciază}(f_z(X), X))$$

8. **Transformă expresiile în clauze**

$$\{ \text{laborator}(f_y(X)) , \text{apreciază}(f_z(X), X) \}$$

$$\{ \neg \text{rezolvă}(X, f_y(X)) , \text{apreciază}(f_z(X), X) \}$$

# Logica cu predicate de ordinul întâi – Cuprins

- Ontologie
- Sintaxă
- Semantică
- Exemple
- Decidabilitate
- Forma clauzală
- Algoritm de aducere la forma clauzală
- Unificare

# Unificare

Unificarea a 2 expresii = găsirea celei mai generale substituții

$$S = \{e_1 / var_1, e_2 / var_2, \dots, e_n / var_n\}$$

pentru variabilele din cele 2 expresii astfel încât, în urma substituției, cele 2 expresii devin una și aceeași

## Utilitate

$$\frac{\{\neg\text{cal}(X), \text{mănâncă}(X, \text{ovăz})\}}{\{\text{cal}(X)\}} \text{ (Rezoluție)} \\ \text{mănâncă}(X, \text{ovăz})$$

$$\frac{\{\neg\text{cal}(X), \text{mănâncă}(X, \text{ovăz})\}}{\{\text{cal}(\text{rocinante})\}} \text{ (Rezoluție)} \\ ?$$

Mulțumită unificării, putem conchide și că Rocinante mănâncă ovăz.

# Unificare – Exemple

- P = mănâncă(vivi, X)  
Q = hrănește(vivi, puiu)
  
- P = mănâncă(vivi, X)  
Q = mănâncă(X, mălai)
  
- P = mănâncă(Canibal, tata(Canibal))  
Q = mănâncă(tata(X), Y)

# Unificare – Exemple

- P = mănâncă(vivi, X)  
Q = hrănește(vivi, puiu)

Nu unifică!  
(nu au același predicat)

- P = mănâncă(vivi, X)  
Q = mănâncă(X, mălai)

- P = mănâncă(Canibal, tata(Canibal))  
Q = mănâncă(tata(X), Y)

# Unificare – Exemple

- P = mănâncă(vivi, X)  
Q = hrănește(vivi, puiu)

Nu unifică!  
(nu au același predicat)

- P = mănâncă(vivi, X)  
Q = mănâncă(X, mălai)

Nu unifică!  
(vivi/X, mălai/X, vivi nu unifică cu mălai)

- P = mănâncă(Canibal, tata(Canibal))  
Q = mănâncă(tata(X), Y)

# Unificare – Exemple

- P = mănâncă(vivi, X)  
Q = hrănește(vivi, puiu)

Nu unifică!  
(nu au același predicat)

- P = mănâncă(vivi, X)  
Q = mănâncă(X, mălai)

Nu unifică!  
(vivi/X, mălai/X, vivi nu unifică cu mălai)

- P = mănâncă(Canibal, tata(Canibal))  
Q = mănâncă(tata(X), Y)

$S = \{ \text{tata}(X) / \text{Canibal}, \text{tata}(\text{Canibal}) / Y \} \rightarrow$   
 $S = \{ \text{tata}(X) / \text{Canibal}, \text{tata}(\text{tata}(X)) / Y \}$

Așa Da

Așa NU

subst(S, P) = mănâncă(tata(X), tata(tata(X)))

subst(S, Q) = mănâncă(tata(X), tata(tata(X)))

subst(S, P) = mănâncă(tata(X), bunicul(X))



# Reguli folosite în algoritmul de unificare

- O **variabilă**  $X$  unifică cu un **termen**  $t$  ( $S = \{t / X\}$ ) dacă și numai dacă
  - $t = X$  sau
  - $t$  nu conține  $X$  (occurs check – pentru a evita legări ciclice)
- **2 constante / 2 funcții / o funcție și o constantă** unifică ( $S = \{ \}$ ) dacă și numai dacă se evaluează la același obiect
- **2 propoziții atomice** unifică dacă și numai dacă sunt aplicații ale aceluiași predicat asupra câte  $n$  termeni care unifică recursiv

**Observație:** Rezoluția lucrează doar cu propoziții atomice și negațiile lor, nu avem nevoie să unificăm și propoziții compuse

# Rezoluție pentru clauze Horn

$$\frac{\begin{array}{l} p_1 \wedge \cdots \wedge p_m \rightarrow p \\ q_1 \wedge \cdots \wedge p' \wedge \cdots \wedge q_n \rightarrow q \\ \text{unificare}(p, p') = S \end{array}}{\text{subst}(S, p_1 \wedge \cdots \wedge p_m, q_1 \wedge \cdots \wedge q_n \rightarrow q)} \quad (\text{Rezoluție})$$

**Observație:** Pentru a simplifica demonstrațiile, propozițiile în Prolog sunt restricționate la clauze Horn

# Comparație LP – LPOI

	Logica propozițională	Logica cu predicate de ordinul întâi
Ontologie		
Sintaxă		
Semantică		
Inferență		
Complexitate		

# Comparație LP – LPOI

	Logica propozițională	Logica cu predicate de ordinul întâi
Ontologie	Fapte (propoziții declarative)	Obiecte, proprietăți, relații
Sintaxă		
Semantică		
Inferență		
Complexitate		

# Comparație LP – LPOI

	Logica propozițională	Logica cu predicate de ordinul întâi
<b>Ontologie</b>	Fapte (propoziții declarative)	Obiecte, proprietăți, relații
<b>Sintaxă</b>	Atomi, conective	Atomi, conective, termeni structurați, variabile cuantificate
<b>Semantică</b>		
<b>Inferență</b>		
<b>Complexitate</b>		

# Comparație LP – LPOI

	Logica propozițională	Logica cu predicate de ordinul întâi
<b>Ontologie</b>	Fapte (propoziții declarative)	Obiecte, proprietăți, relații
<b>Sintaxă</b>	Atomi, conective	Atomi, conective, termeni structurați, variabile cuantificate
<b>Semantică</b>	Interpretări (în tabele de adevăr)	Interpretări (complexe)
<b>Inferență</b>		
<b>Complexitate</b>		

# Comparație LP – LPOI

	Logica propozițională	Logica cu predicate de ordinul întâi
<b>Ontologie</b>	Fapte (propoziții declarative)	Obiecte, proprietăți, relații
<b>Sintaxă</b>	Atomi, conective	Atomi, conective, termeni structurați, variabile cuantificate
<b>Semantică</b>	Interpretări (în tabele de adevăr)	Interpretări (complexe)
<b>Inferență</b>	Rezoluție	Rezoluție + unificare
<b>Complexitate</b>		

# Comparație LP – LPOI

	Logica propozițională	Logica cu predicate de ordinul întâi
<b>Ontologie</b>	Fapte (propoziții declarative)	Obiecte, proprietăți, relații
<b>Sintaxă</b>	Atomi, conective	Atomi, conective, termeni structurați, variabile cuantificate
<b>Semantică</b>	Interpretări (în tabele de adevăr)	Interpretări (complexe)
<b>Inferență</b>	Rezoluție	Rezoluție + unificare
<b>Complexitate</b>	Decidabilă dar NP-completă	Semidecidabilă



# Rezumat

Satisfiabilitate

Validitate

Derivabilitate

Inferență

Reguli de inferență

Regulă consistentă

Regulă completă

Demonstrație

Strategii de control

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

Validitate

Derivabilitate

Inferență

Reguli de inferență

Regulă consistentă

Regulă completă

Demonstrație

Strategii de control

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

Derivabilitate

Inferență

Reguli de inferență

Regulă consistentă

Regulă completă

Demonstrație

Strategii de control

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

Inferență

Reguli de inferență

Regulă consistentă

Regulă completă

Demonstrație

Strategii de control

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

Reguli de inferență

Regulă consistentă

Regulă completă

Demonstrație

Strategii de control

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

Regulă consistentă

Regulă completă

Demonstrație

Strategii de control

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

**Regulă consistentă:** determină doar concluzii care derivă logic din premise

Regulă completă

Demonstrație

Strategii de control

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

**Regulă consistentă:** determină doar concluzii care derivă logic din premise

**Regulă completă:** determină toate concluziile care derivă logic din premise

Demonstrație

Strategii de control

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare



# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

**Regulă consistentă:** determină doar concluzii care derivă logic din premise

**Regulă completă:** determină toate concluziile care derivă logic din premise

**Demonstrație:** secvență de propoziții adevărate (premise, axiome, concluzii ale regulilor)

Strategii de control

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

**Regulă consistentă:** determină doar concluzii care derivă logic din premise

**Regulă completă:** determină toate concluziile care derivă logic din premise

**Demonstrație:** secvență de propoziții adevărate (premise, axiome, concluzii ale regulilor)

**Strategii de control:** forward chaining, backward chaining

Formă clauzală

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

**Regulă consistentă:** determină doar concluzii care derivă logic din premise

**Regulă completă:** determină toate concluziile care derivă logic din premise

**Demonstrație:** secvență de propoziții adevărate (premise, axiome, concluzii ale regulilor)

**Strategii de control:** forward chaining, backward chaining

**Formă clauzală:** conjuncție de disjuncții de literali

Clauză Horn

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

**Regulă consistentă:** determină doar concluzii care derivă logic din premise

**Regulă completă:** determină toate concluziile care derivă logic din premise

**Demonstrație:** secvență de propoziții adevărate (premise, axiome, concluzii ale regulilor)

**Strategii de control:** forward chaining, backward chaining

**Formă clauzală:** conjuncție de disjuncții de literali

**Clauză Horn:** clauză cu un singur literal nenegat

Demonstrabile prin rezoluție

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

**Regulă consistentă:** determină doar concluzii care derivă logic din premise

**Regulă completă:** determină toate concluziile care derivă logic din premise

**Demonstrație:** secvență de propoziții adevărate (premise, axiome, concluzii ale regulilor)

**Strategii de control:** forward chaining, backward chaining

**Formă clauzală:** conjuncție de disjuncții de literali

**Clauză Horn:** clauză cu un singur literal nenegat

**Demonstrabile prin rezoluție:** validitate, derivabilitate, nesatisfiabilitate

Nedemonstrabile prin rezoluție

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

**Regulă consistentă:** determină doar concluzii care derivă logic din premise

**Regulă completă:** determină toate concluziile care derivă logic din premise

**Demonstrație:** secvență de propoziții adevărate (premise, axiome, concluzii ale regulilor)

**Strategii de control:** forward chaining, backward chaining

**Formă clauzală:** conjuncție de disjuncții de literali

**Clauză Horn:** clauză cu un singur literal nenegat

**Demonstrabile prin rezoluție:** validitate, derivabilitate, nesatisfiabilitate

**Nedemonstrabile prin rezoluție:** invaliditate, satisfiabilitate

Unificare

# Rezumat

**Satisfiabilitate:**  $P = \text{true}$  în cel puțin o interpretare

**Validitate:**  $P = \text{true}$  în toate interpretările

**Derivabilitate:**  $P = \text{true}$  în toate interpretările care satisfac premisele ( $\Delta \models P$ )

**Inferență:** derivare prin calcul a concluziilor unei mulțimi de premise ( $\Delta \vdash_{\text{id\_regulă}} P$ )

**Reguli de inferență:** Modus Ponens, Modus Tollens, Rezoluție, axiome

**Regulă consistentă:** determină doar concluzii care derivă logic din premise

**Regulă completă:** determină toate concluziile care derivă logic din premise

**Demonstrație:** secvență de propoziții adevărate (premise, axiome, concluzii ale regulilor)

**Strategii de control:** forward chaining, backward chaining

**Formă clauzală:** conjuncție de disjuncții de literali

**Clauză Horn:** clauză cu un singur literal nenegat

**Demonstrabile prin rezoluție:** validitate, derivabilitate, nesatisfiabilitate

**Nedemonstrabile prin rezoluție:** invaliditate, satisfiabilitate

**Unificare:** cea mai generală substituție conform căreia 2 expresii devin identice