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 proprietați 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: $p \land q$ = true în interpretarea I / $p \land q$ = false în interpretarea J)
  - Conform unor **reguli semantice** care determină valoarea de adevăr a unei propoziții compuse din valorile de adevăr ale propozițiilor constituente

Reguli semantice pentru propoziții compuse

$\neg P1 = true$	dacă și numai dacă	$P1 = false$		
$P1 \land P2 = true$	dacă și numai dacă	$P1 = true$	și	$P2 = true$
$P1 \lor P2 = true$	dacă și numai dacă	$P1 = true$	sau	$P2 = true$
$P1 \Rightarrow P2 = true$	dacă și numai dacă	$P1 = false$	sau	$P2 = true$
$P1 \Leftrightarrow P2 = true$	dacă și numai dacă	$P1 = P2$		

Evaluare

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

Exemplu

$E = p \land (q \lor r)$ în interpretarea $I = \{ p = true, q = true, r = false \}$

$E = true \land (true \lor false) = true \land true = true$

Logica propozițională – Cuprins

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

Satisfiabilitate

Propoziția P este **satisfiabilă** ⇔ P este adevărată (true) în cel puțin o interpretare

Propoziția P este **nesatisfiabilă** ⇔ P este falsă (false) în toate interpretările

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

Exemple

- $p \land (q \lor r)$ satisfiabilă conform tabelei alăturate (întrucât este adevărată în 3 interpretări)
- $p \land \neg p$ nesatisfiabilă (se determină similar)

p	q	r	$p \land (q \lor r)$
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)** ⇔ P este adevărată (true) în toate interpretările

Propoziția P este **invalidă** ⇔ P este falsă (false) în cel puțin o interpretare

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

Exemple

- (p ∨ q) ∨ (¬q ∨ r) validă conform tabelei alăturate (întrucât este adevărată în toate interpretările)
- p ∧ (q ∨ r) invalidă conform paginii anterioare (întrucât este falsă în 5 interpretări)

p	q	r	(p ∨ q) ∨ (¬q ∨ 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 Δ **derivă** propoziția P **(Δ ⊨ P)** ⇔ orice interpretare care satisface toate propozițiile din Δ satisface și propoziția P

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

Exemplu

- { p, p ⇒ q} ⊨ q conform tabelei alăturate (Singura interpretare care satisface p și p ⇒ q este prima, și prima interpretare satisface și q)

p	q	p ⇒ 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 ⇒ 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**

⇒ 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

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

P_i = premise (propoziții considerate adevărate)

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

Notații

$\Delta \vDash C$ - derivabilitate logică

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

Reguli de inferență

$$\frac{\begin{array}{c} A \Rightarrow B \\ A \end{array}}{B} \; (ModusPonens)$$

$$\frac{\begin{array}{c} A \Rightarrow B \\ \neg B \end{array}}{\neg A} \; (ModusTollens)$$

$$\frac{\begin{array}{c} A \vee B \\ \neg A \vee C \end{array}}{B \vee C} \; (Rezoluție)$$

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

Regulile fără premise se numesc **axiome**

Reguli de inferență

$$\frac{A \Rightarrow B \quad B \Rightarrow C}{A \Rightarrow C}\ (TranziImp)$$

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

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

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

Regulile fără premise se numesc **axiome**

Proprietăți ale regulilor de inferență

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

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

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

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

- 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)

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

Exemplu

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

$\{ p \Rightarrow q, q \Rightarrow r \} \nvdash_{\text{ModusPonens}} p \Rightarrow r$ (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
- Instanțe 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:

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

Axiome

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

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

1.	$p \Rightarrow q$	*(Premisă)*
2.	$q \Rightarrow r$	*(Premisă)*
3.	$(q \Rightarrow r) \Rightarrow (p \Rightarrow (q \Rightarrow r))$	*(IntroImp)*
4.	$p \Rightarrow (q \Rightarrow r)$	*(ModusPonens 2,3)*
5.	$(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$	*(DistriImp)*
6.	$(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$	*(ModusPonens 4,5)*
7.	$p \Rightarrow r$	*(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)
  - Oprire 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)

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

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 | ¬p (propoziții atomice sau negațiile lor)

Clauză = mulțime de literali aflați în relația de disjuncție

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

Clauză Horn = clauză în care un singur literal este nenegat

- { p, ¬q, ¬r } echivalent cu $p \lor \neg q \lor \neg r$ și cu $q \land r \Rightarrow p$

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

- { {p}, {q, ¬r}, {¬p, r} } echivalent cu $p \land (q \lor \neg r) \land (\neg p \lor r)$

Rezoluție în forma clauzală

Forma simplă

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

Forma generală

$$\frac{\{\mathrm{q}_1,\ \ldots, p, \ldots q_m\}\quad \{\mathrm{r}_1,\ \ldots, \neg p, \ldots r_n\}}{\{\mathrm{q}_1\ \ldots\ q_m, r_1\ \ldots r_n\}}\ (Rezoluție)$$

Contradicție

$$\frac{\{\mathrm{p}\}\quad \{\neg \mathrm{p}\}}{\{\,\}}\ (Rezoluție)$$

Rezolvent vid: indică existența unei contradicții între premise

Cazuri particulare ale Rezoluției

Modus Ponens

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

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

Modus Tollens

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

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

Tranzitivitatea implicației

$$\frac{A \Rightarrow B \quad B \Rightarrow C}{A \Rightarrow C} \ (TranziImp)$$

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

Diferența de notație (A, B vs. p, q) indică faptul că în cazul formei clauzale 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 ¬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 ¬P în premise ← reducere la absurd
  - Derivez clauza vidă
- **P validă**
  - Pornesc cu premisa ¬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

∀X,Y.((iepure(X) ∧ țestoasă(Y)) ⇒ mai_rapid(X,Y)).	*Iepurii sunt mai rapizi decât broaștele țestoase.*
iepure(bugs).	*Bugs e iepure.*
țestoasă(leo).	*Leo e broască țestoasă.*
mai_rapid(bugs, leo).	*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)
- **Cuantificatori:** $\forall$, $\exists$
- **Egalitate:** =

Asociativi la dreapta

Sintaxă

termen = constantă | variabilă | funcție(termen,… termen)

propoziție atomică = predicat(termen,… termen) | termen = termen

propoziție = propoziție atomică | (propoziție) | ¬propoziție |
propoziție conectivă_binară propoziție | cuantificator var_1 … 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 ($c \in D$, $f : D^n \longrightarrow D$)
  - Predicatelor li se asociază o valoare de adevăr (true sau false) ($p : D^n \longrightarrow \{true, false\}$)

Reguli semantice

Cele de la logica propozițională (pentru conective) +

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

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

Logica cu predicate de ordinul întâi – Cuprins

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

Exemple

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

Exemple

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

Exemple

1. Vrabia mălai visează.

   $\forall X \bullet (vrabie(X) \Rightarrow visează(X,mălai))$
2. Unele vrăbii visează mălai.

   $\exists X \bullet (vrabie(X) \wedge visează(X, mălai))$
3. Nu toate vrăbiile visează mălai.
4. Nicio vrabile nu visează mălai.
5. Numai vrăbiile visează mălai.
6. Toate și numai vrăbiile visează mălai.
7. Vivi este o vrabie care visează mălai.

Exemple

1. Vrabia mălai visează.

   $\forall X \bullet (\text{vrabie}(X) \Rightarrow \text{visează}(X,\text{mălai}))$
2. Unele vrăbii visează mălai.

   $\exists X \bullet (\text{vrabie}(X) \wedge \text{visează}(X, \text{mălai}))$
3. Nu toate vrăbiile visează mălai.

   $\neg(\forall X \bullet (\text{vrabie}(X) \Rightarrow \text{visează}(X, \text{mălai})))$ sau $\exists X \bullet (\text{vrabie}(X) \wedge \neg\text{visează}(X, \text{mălai}))$
4. Nicio vrabile nu visează mălai.
5. Numai vrăbiile visează mălai.
6. Toate și numai vrăbiile visează mălai.
7. Vivi este o vrabie care visează mălai.

Exemple

1. Vrabia mălai visează.

   $\forall X \bullet (\text{vrabie}(X) \Rightarrow \text{visează}(X,\text{mălai}))$
2. Unele vrăbii visează mălai.

   $\exists X \bullet (\text{vrabie}(X) \wedge \text{visează}(X, \text{mălai}))$
3. Nu toate vrăbiile visează mălai.

   $\neg(\forall X \bullet (\text{vrabie}(X) \Rightarrow \text{visează}(X, \text{mălai})))$ sau $\exists X \bullet (\text{vrabie}(X) \wedge \neg\text{visează}(X, \text{mălai}))$
4. Nicio vrabile nu visează mălai.

   $\neg(\exists X \bullet (\text{vrabie}(X) \wedge \text{visează}(X, \text{mălai})))$ sau $\forall X \bullet (\text{vrabie}(X) \Rightarrow \neg\text{visează}(X, \text{mălai}))$
5. Numai vrăbiile visează mălai.

6. Toate și numai vrăbiile visează mălai.

7. Vivi este o vrabie care visează mălai.

Exemple

1. Vrabia mălai visează.
   $\forall X \bullet (vrabie(X) \Rightarrow visează(X,mălai))$
2. Unele vrăbii visează mălai.
   $\exists X \bullet (vrabie(X) \land visează(X, mălai))$
3. Nu toate vrăbiile visează mălai.
   $\neg(\forall X \bullet (vrabie(X) \Rightarrow visează(X, mălai)))$ sau $\exists X \bullet (vrabie(X) \land \neg visează(X, mălai))$
4. Nicio vrabile nu visează mălai.
   $\neg(\exists X \bullet (vrabie(X) \land visează(X, mălai)))$ sau $\forall X \bullet (vrabie(X) \Rightarrow \neg visează(X, mălai))$
5. Numai vrăbiile visează mălai.
   $\forall X \bullet (visează(X, mălai) \Rightarrow vrabie(X))$
6. Toate și numai vrăbiile visează mălai.

7. Vivi este o vrabie care visează mălai.

Exemple

1. Vrabia mălai visează.

   $\forall X \bullet (vrabie(X) \Rightarrow visează(X,mălai))$

2. Unele vrăbii visează mălai.

   $\exists X \bullet (vrabie(X) \land visează(X, mălai))$

3. Nu toate vrăbiile visează mălai.

   $\neg(\forall X \bullet (vrabie(X) \Rightarrow visează(X, mălai)))$ sau $\exists X \bullet (vrabie(X) \land \neg visează(X, mălai))$

4. Nicio vrabile nu visează mălai.

   $\neg(\exists X \bullet (vrabie(X) \land visează(X, mălai)))$ sau $\forall X \bullet (vrabie(X) \Rightarrow \neg visează(X, mălai))$

5. Numai vrăbiile visează mălai.

   $\forall X \bullet (visează(X, mălai) \Rightarrow vrabie(X))$

6. Toate și numai vrăbiile visează mălai.

   $\forall X \bullet (visează(X, mălai) \Leftrightarrow vrabie(X))$

7. Vivi este o vrabie care visează mălai.

Exemple

1. Vrabia mălai visează.

   $\forall X \bullet (\text{vrabie}(X) \Rightarrow \text{visează}(X,\text{mălai}))$

2. Unele vrăbii visează mălai.

   $\exists X \bullet (\text{vrabie}(X) \land \text{visează}(X, \text{mălai}))$

3. Nu toate vrăbiile visează mălai.

   $\neg(\forall X \bullet (\text{vrabie}(X) \Rightarrow \text{visează}(X, \text{mălai})))$ sau $\exists X \bullet (\text{vrabie}(X) \land \neg\text{visează}(X, \text{mălai}))$

4. Nicio vrabile nu visează mălai.

   $\neg(\exists X \bullet (\text{vrabie}(X) \land \text{visează}(X, \text{mălai})))$ sau $\forall X \bullet (\text{vrabie}(X) \Rightarrow \neg\text{visează}(X, \text{mălai}))$

5. Numai vrăbiile visează mălai.

   $\forall X \bullet (\text{visează}(X, \text{mălai}) \Rightarrow \text{vrabie}(X))$

6. Toate și numai vrăbiile visează mălai.

   $\forall X \bullet (\text{visează}(X, \text{mălai}) \Leftrightarrow \text{vrabie}(X))$

7. Vivi este o vrabie care visează mălai.

   $\text{vrabie}(\text{vivi}) \land \text{visează}(\text{vivi}, \text{mălai})$

Greșeli frecvente

- În general:
  - $\forall$ se folosește împreună cu $\Rightarrow$
  - $\exists$ se folosește împreună cu $\wedge$

- $\forall X \bullet (vrabie(X) \wedge visează(X, mălai))$
  ?

- $\exists X \bullet (vrabie(X) \Rightarrow visează(X, mălai))$
  ?

Greșeli frecvente

- În general:
  - $\forall$ se folosește împreună cu $\Rightarrow$
  - $\exists$ se folosește împreună cu $\wedge$

- $\forall X \bullet (vrabie(X) \wedge visează(X, mălai))$
  înseamnă că toată lumea e vrabie și toată lumea visează mălai

- $\exists X \bullet (vrabie(X) \Rightarrow visează(X, mălai))$
  este o propoziție adevărată și dacă există cineva care nu e vrabie

Proprietăți cuantificatori și conective

- **(Ne)Comutativitate**

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

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

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

  - $\exists X \bullet \forall Y \bullet$visează(X,Y) inseamnă că ....
  - $\forall Y \bullet \exists X \bullet$visează(X,Y) inseamnă că ....

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

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

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

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

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

Proprietăți cuantificatori și conective

- **(Ne)Comutativitate**

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

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

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

  - $\exists X \bullet \forall Y \bullet$visează(X,Y) inseamnă că există cineva care visează la toată lumea
  - $\forall Y \bullet \exists X \bullet$visează(X,Y) inseamnă că la oricine visează măcar cineva

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

$\forall X \bullet p = \neg(\exists X \bullet \neg p)$		$p \lor q = \neg(\neg p \land \neg q)$
$\exists X \bullet p = \neg(\forall X \bullet \neg p)$		$p \land q = \neg(\neg p \lor \neg q)$
$\neg(\forall X \bullet p) = \exists X \bullet \neg p$		$\neg(p \lor q) = \neg p \land \neg q$
$\neg(\exists X \bullet p) = \forall X \bullet \neg p$		$\neg(p \land q) = \neg p \lor \neg q$

Definiții moștenite de la logica propozițională

- **Evaluare:** determinarea valorii de adevăr a lui P (într-o interpretare) prin aplicarea regulilor semantice
- **Satisfiabilitate:** P = true în cel puțin o interpretare
- **Validitate:** P = true în toate interpretările
- **Derivabilitate:** P = 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 ⇒ 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), ¬fricos(X) } echivalent cu prieten(X,porthos) ∨ ¬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**
   $\neg(a \vee b) = \neg a \wedge \neg b$ $\neg(\forall X \bullet p) = \exists X \bullet \neg p$ $\neg(\neg p) = p$
   $\neg(a \wedge b) = \neg a \vee \neg b$ $\neg(\exists X \bullet p) = \forall X \bullet \neg p$

3. **Redenumește variabilele cuantificate a.î. să nu existe cuantificatori diferiți care folosesc același nume de variabilă**
   (ex: $\forall X \bullet p(X) \wedge \forall X \bullet q(X) \vee \exists X \bullet r(X)$ devine $\forall X \bullet p(X) \wedge \forall Y \bullet q(Y) \vee \exists Z \bullet r(Z)$)

4. **Forma prenex: Mută toți cuantificatorii la începutul propoziției, în aceeași ordine**
   (ex: $\forall X \bullet p(X) \wedge \forall Y \bullet q(Y) \vee \exists Z \bullet r(Z)$ devine $\forall X \bullet \forall Y \bullet \exists Z \bullet (p(X) \wedge q(Y) \vee r(Z))$)

Algoritm de aducere la forma clauzală (II)

5. **Skolemizare: Elimină cuantificatorii existențiali prin înlocuire cu**
   - O **constantă** - dacă acest ∃ nu era precedat de niciun ∀
   - O **funcție** de toate variabilele cuantificate universal care îl precedau - altfel
     (ex: $\forall X \bullet \forall Y \bullet \exists Z \bullet (p(X) \land q(Y) \lor r(Z))$ devine $\forall X \bullet \forall Y \bullet (p(X) \land q(Y) \lor r(f(X,Y)))$)

6. **Elimină cuantificatorii universali (ce rămâne e automat cuantificat universal)**
   (ex: $\forall X \bullet \forall Y \bullet (p(X) \land q(Y) \lor r(f(X,Y)))$ devine $p(X) \land q(Y) \lor r(f(X,Y))$)

7. **Distribuie ∨ prin ∧**
   $(a \land b) \lor c$ devine $(a \lor c) \land (b \lor c)$

8. **Transformă expresiile în clauze**

Aducere la forma clauzală – Exemplu (I)

"Oricine rezolvă toate laboratoarele este apreciat de cineva."

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

1. **Elimină implicațiile**
   $\forall X \bullet (\neg(\forall Y \bullet \neg \text{laborator}(Y) \vee \text{rezolvă}(X,Y)) \vee \exists Y \bullet \text{apreciază}(Y,X))$

2. **Mută negațiile spre interior**
   $\forall X \bullet (\exists Y \bullet \neg(\neg \text{laborator}(Y) \vee \text{rezolvă}(X,Y)) \vee \exists Y \bullet \text{apreciază}(Y,X))$
   $\forall X \bullet (\exists Y \bullet (\text{laborator}(Y) \wedge \neg \text{rezolvă}(X,Y)) \vee \exists Y \bullet \text{apreciază}(Y,X))$

3. **Redenumește variabilele cuantificate**
   $\forall X \bullet (\exists Y \bullet (\text{laborator}(Y) \wedge \neg \text{rezolvă}(X,Y)) \vee \exists Z \bullet \text{apreciază}(Z,X))$

4. **Mută cuantificatorii la început**
   $\forall X \bullet \exists Y \bullet \exists Z \bullet ((\text{laborator}(Y) \wedge \neg \text{rezolvă}(X,Y)) \vee \text{apreciază}(Z,X))$

Aducere la forma clauzală – Exemplu (II)

"Oricine rezolvă toate laboratoarele este apreciat de cineva."

$$\boxed{\forall X \bullet \exists Y \bullet \exists Z \bullet ((laborator(Y) \land \neg rezolvă(X,Y)) \lor apreciază(Z,X))}$$

1. **Elimină cuantificatorii existențiali**
   $\forall X \bullet ((laborator(f_y(X)) \land \neg rezolvă(X,f_y(X))) \lor apreciază(f_z(X),X))$
2. **Elimină cuantificatorii universali**
   $(laborator(f_y(X)) \land \neg rezolvă(X,f_y(X))) \lor apreciază(f_z(X),X)$
3. **Distribuie ∨ prin ∧**
   $(laborator(f_y(X)) \lor apreciază(f_z(X),X)) \land (\neg rezolvă(X,f_y(X)) \lor apreciază(f_z(X),X))$
4. **Transformă expresiile în clauze**
   $\{ laborator(f_y(X)) , apreciază(f_z(X),X) \}$
   $\{ \neg rezolvă(X,f_y(X)) , apreciază(f_z(X),X) \}$

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 propoziții = **găsirea celei mai generale substituții**

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

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

Utilitate

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

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

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 = { tata(X) / Canibal, tata(Canibal) / Y } →
  S = { tata(X) / Canibal, tata(tata(X)) / Y }
  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))

Așa Da → subst(S,P) = mănâncă(tata(X), tata(tata(X)))

Așa NU → 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 în forma cea mai generală

$$\frac{\begin{array}{c}\{\mathrm{q}_1, \ \dots, \boldsymbol{p}, \dots \mathrm{q}_\mathrm{m}\} \\ \{\mathrm{r}_1, \ \dots, \boldsymbol{\neg p'}, \dots \mathrm{r}_\mathrm{n}\} \\ \mathrm{unificare}(p, p') = \mathrm{S}\end{array}}{\mathrm{subst}(\mathrm{S}, \{\mathrm{q}_1 \ \dots \ \mathrm{q}_\mathrm{m}, \mathrm{r}_1 \ \dots \mathrm{r}_\mathrm{n}\})} \ (Rezoluție)$$

Exemplu

$$\frac{\begin{array}{c}\{\mathrm{iepure}(\mathrm{leo}), \mathrm{cal}(\mathrm{rocinante})\} \\ \{\neg\mathrm{iepure}(\mathrm{X}), \mathrm{mănâncă}(\mathrm{X}, \mathrm{morcov}), \mathrm{mănâncă}(\mathrm{X}, \mathrm{varză})\} \\ \mathrm{unificare}(\mathrm{iepure}(\mathrm{leo}), \mathrm{iepure}(\mathrm{X})) = \{\mathrm{leo}/\mathrm{X}\}\end{array}}{\{\mathrm{cal}(\mathrm{rocinante}), \mathrm{mănâncă}(\mathrm{leo}, \mathrm{morcov}), \mathrm{mănâncă}(\mathrm{leo}, \mathrm{varză})\}} \ (Rezoluție)$$

Rezoluție pentru clauze Horn

$$\frac{\begin{array}{c} \mathrm{p}_1 \land \cdots \land \mathrm{p}_\mathrm{m} \rightarrow \boldsymbol{p} \\ \mathrm{q}_1 \land \cdots \land \boldsymbol{p'} \land \cdots \land \mathrm{q}_\mathrm{n} \rightarrow \mathrm{q} \\ \mathrm{unificare}(p, p') = \mathrm{S} \end{array}}{\mathrm{subst}(S, \mathrm{p}_1 \land \cdots \land \mathrm{p}_\mathrm{m}, \mathrm{q}_1 \land \cdots \land \mathrm{q}_\mathrm{n} \rightarrow \mathrm{q})} \ (Rezoluție)$$

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
Ontologie	Fapte (propoziții declarative)	Obiecte, proprietăți, relații
Sintaxă	Atomi, conective	Atomi, conective, termeni structurați, variabile cuantificate
Semantică		
Inferență		
Complexitate		

Comparație LP – LPOI

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

Comparație LP – LPOI

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

Comparație LP – LPOI

	Logica propozițională	Logica cu predicate de ordinul întâi
Ontologie	Fapte (propoziții declarative)	Obiecte, proprietăți, relații
Sintaxă	Atomi, conective	Atomi, conective, termeni structurați, variabile cuantificate
Semantică	Interpretări (în tabele de adevăr)	Interpretări (complexe)
Inferență	Rezoluție	Rezoluție + unificare
Complexitate	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 = 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 = true în cel puțin o interpretare
Validitate: P = 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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = 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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \models P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regulă} P$)
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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \models P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regulă} P$)
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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = 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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \vDash P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regulă} P$)
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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \models P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regulă} P$)
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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \models P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regulă} P$)
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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \vDash P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regulă} P$)
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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \vDash P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regul\breve{a}} P$)
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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \vDash P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regulă} P$)
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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \models P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regulă} P$)
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 = true în cel puțin o interpretare
Validitate: P = true în toate interpretările
Derivabilitate: P = true în toate interpretările care satisfac premisele ($\Delta \vDash P$)
Inferență: derivare prin calcul a concluziilor unei mulțimi de premise ($\Delta \vdash_{id\_regulă} P$)
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 propoziții devin identice