

PARADIGME DE PROGRAMARE

Curs 9

Logica propozițională. Logica cu predicate de ordinul întâi.

1

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

2

2

Logica propozițională – Cuprins

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

3

3

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

4

4

Logica propozițională – Cuprins

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

5

5

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
 $(\neg P)$ | $(P \wedge Q)$ | $(P \vee Q)$ | $(P \Rightarrow Q)$ | $(P \Leftrightarrow Q)$

6

6

Logica propozițională – Cuprins

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

7

7

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 constituente

8

8

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	$P_1 = P_2$		

9

9

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 $I = \{ p = \text{true}, q = \text{true}, r = \text{false} \}$

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

10

10

Logica propozițională – Cuprins

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

11

11

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 alăturată (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

12

12

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

13

13

Logica propozițională – Cuprins

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

14

14

Derivabilitate

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

15

15

Logica propozițională – Cuprins

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

16

16

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)

17

17

Logica propozițională – Cuprins

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

18

18

Inferență

Inferență = derivarea prin calcul a concluziilor unei mulțimi de premise

Regulă de inferență

$$P_1, P_2, \dots, P_n$$

P_i = premise (propoziții considerate adevărate)

$$\frac{}{C} \quad (\text{id_regulă})$$

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

Notații

$\Delta \models C$ - derivabilitate logică

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

19

19

Reguli de inferență

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

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

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

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

Regulile fără premise se numesc **axiome**

20

20

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)

21

21

Proprietăți Modus Ponens

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

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

Exemplu

- $\{p \Rightarrow q, q \Rightarrow r\} \models p \Rightarrow r$ (se poate verifica prin tabela de adevăr)
- $\{p \Rightarrow q, q \Rightarrow r\} \not\vdash_{\text{Modus Ponens}} p \Rightarrow r$ (nu am informație despre vreo propoziție atomică)

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

22

22

Logica propozițională – Cuprins

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

23

23

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

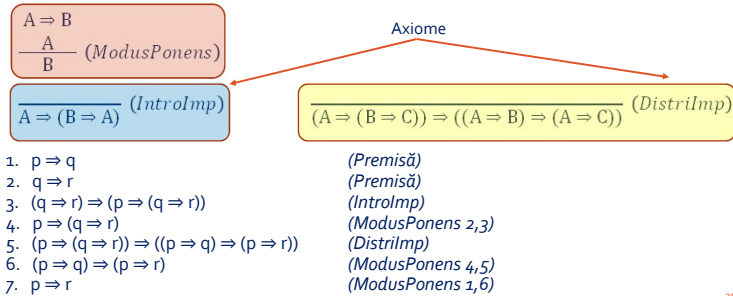
- Aceleași elemente de mai sus +
- La premise se adaugă $\neg C$ (negația teoremei, singurul posibil neadevăr, cauza pentru care o propoziție falsă a fost determinată ca fiind adevărată)

24

24

Demonstrație cu Modus Ponens - Exemplu

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



25

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.

26

Logica propozițională – Cuprins

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

27

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 numită **forma clauzală** (acest lucru este mereu posibil).

28

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 $p \wedge (q \vee \neg r) \wedge (\neg p \vee r)$

29

29

Rezoluție în forma clauzală

Forma simplă

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

Forma generală

$$\frac{\{q_1, \dots, p, \dots, q_m\} \quad \{r_1, \dots, \neg p, \dots, r_n\}}{\{q_1, \dots, q_m, r_1, \dots, r_n\}} \text{ (Rezoluție)}$$

Contradicție

$$\frac{\{p\} \quad \{\neg p\}}{\{\}} \text{ (Rezoluție)}$$

Rezolvent vid: indică existența unei contradicții între premise

30

30

Cazuri particulare ale Rezoluției

Modus Ponens

$$\frac{\{ \neg p, q \} \quad \frac{A \Rightarrow B \quad A}{B} \text{ (ModusPonens)}}{\{p\}} \text{ (Rezoluție)}$$

Modus Tollens

$$\frac{\{ \neg p, q \} \quad \frac{A \Rightarrow B \quad \neg B}{\neg A} \text{ (ModusTollens)}}{\{ \neg q \}} \text{ (Rezoluție)}$$

Tranzitivitatea implicației

$$\frac{\{ \neg p, q \} \quad \frac{A \Rightarrow B \quad B \Rightarrow C}{A \Rightarrow C} \text{ (TranzImp)}}{\{ \neg q, r \}} \text{ (Rezoluție)}$$

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

31

31

Demonstrații bazate pe Rezoluție

Fie P o propoziție în formă clauzală. Pentru a demonstra

- P nesatisfiabilă
- P derivabilă din premise
- P validă

32

32

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**

33

33

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**

34

34

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

35

35

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) |

36

36

Test

Completați definițiile Haskell de mai jos astfel încât să obțineți în final TDA-ul Card (carte de joc, unde o carte se reprezintă printr-un rang (Rank) (de la 2 la As) și o culoare (Suit) (treflă, caro, etc.)).

```
data Suit =
data Rank =
data Card =
```

Apoi definiți funcția `winner :: [Card] -> Card` care întoarce cartea câștigătoare într-o rundă de joc, știind că o rundă e câștigată de cartea cea mai mare care are aceeași culoare cu prima carte.

Ex: Pentru lista `[2 ♣, A ♠, 4 ♣, 8 ♥]` winner este `4 ♣`.

Obs: Respectați sintaxa Haskell, nu încercați să puneți asemenea simboluri în cod.

37

37

Logica cu predicate de ordinul întâi – Cuprins

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

38

38

Ontologie

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

Exemple

$\forall X, Y. ((iepure(X) \wedge \text{testoasă}(Y)) \Rightarrow \text{mai_rapid}(X, Y)).$ *Iepurii sunt mai rapizi decât broaștele țestoase.*
iepure(bugs). *Bugs e iepure.*
testoasă(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

39

39

Logica cu predicate de ordinul întâi – Cuprins

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

40

40

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 } Asociativi la dreapta
- **Egalitate:** =

41

41

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 $\text{var}_1 \dots \text{var}_n$, propoziție
- Termenii reprezintă obiecte
 - Propozițiile reprezintă proprietăți și relații

42

Logica cu predicate de ordinul întâi – Cuprins

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

43

43

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. $\text{prop}_{\{d/X\}} = \text{true}$

44

44

Logica cu predicate de ordinul întâi – Cuprins

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

45

45

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.

46

46

Exemple



1. Vrăbia mălai visează.
 $\forall X \bullet (\text{vrăbie}(X) \Rightarrow \text{visează}(X, \text{mălai}))$
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.

47

47

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.
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.

48

48

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.
6. Toate și numai vrăbiile visează mălai.
7. Vivi este o vrăbie care visează mălai.

49

49

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.
6. Toate și numai vrăbiile visează mălai.
7. Vivi este o vrăbie care visează mălai.

50

50

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.

51

51

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.

52

52

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

53

53

Greșeli frecvente

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

54

54

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

55

55

Proprietăți cuantificatori și conective

- **(Ne)Comutativitate**
 - $\forall X \bullet \forall Y$ este totuna cu $\forall Y \bullet \forall X$ și se scrie prescurtat $\forall X, Y$.
 - $\exists X \bullet \exists Y$ este totuna cu $\exists Y \bullet \exists X$ și se scrie prescurtat $\exists X, Y$.
 - $\exists X \bullet \forall Y$ NU este totuna cu $\forall Y \bullet \exists X$
 - $\exists X \bullet \forall Y \bullet visează(X, Y)$ înseamnă că
 - $\forall Y \bullet \exists X \bullet visează(X, Y)$ înseamnă că
- **Dualitate (se pot scrie unul/una în funcție de altul/alta)**
 - $\forall X \bullet p = \neg(\exists X \bullet \neg p)$ $p \vee q = \neg(\neg p \wedge \neg q)$
 - $\exists X \bullet p = \neg(\forall X \bullet \neg p)$ $p \wedge q = \neg(\neg p \vee \neg q)$
 - $\neg(\forall X \bullet p) = \exists X \bullet \neg p$ $\neg(p \vee q) = \neg p \wedge \neg q$
 - $\neg(\exists X \bullet p) = \forall X \bullet \neg p$ $\neg(p \wedge q) = \neg p \vee \neg q$

56

56

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 $\forall Y.\exists X$

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

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

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

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

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

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

57

57

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)

58

58

Logica cu predicate de ordinul întâi – Cuprins

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

59

59

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

60

60

Logica cu predicate de ordinul întâi – Cuprins

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

61

61

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

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

62

62

Logica cu predicate de ordinul întâi – Cuprins

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

63

63

Algoritm de aducere la forma clauzală (I)

1. Elimină implicațiile

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

2. Mută negațiile spre interior

$\neg(a \vee b) = \neg a \wedge \neg b$ $\neg(\forall X.p) = \exists X.\neg p$ $\neg(\neg p) = p$
 $\neg(a \wedge b) = \neg a \vee \neg b$ $\neg(\exists X.p) = \forall X.\neg p$

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

64

64

Algoritm de aducere la forma clauzală (II)

5. **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: $\forall X. \forall Y. \exists Z. (p(X) \wedge q(Y) \vee r(Z))$ devine $\forall X. \forall Y. (p(X) \wedge q(Y) \vee r(f(X,Y)))$)
6. **Elimină cuantificatorii universali (ce rămâne e automat cuantificat universal)**
(ex: $\forall X. \forall Y. (p(X) \wedge q(Y) \vee r(f(X,Y)))$ devine $p(X) \wedge q(Y) \vee r(f(X,Y))$)
7. **Distribuie \vee prin \wedge**
($a \wedge b) \vee c$ devine $(a \vee c) \wedge (b \vee c)$)
8. **Transformă expresiile în clauze**

65

65

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

66

66

Aducere la forma clauzală – Exemplu (II)

“Oricine rezolvă toate laboratoarele este apreciat de cineva.”

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

1. **Elimină cuantificatorii existențiali**
 $\forall X. ((\text{laborator}(f_1(X)) \wedge \neg \text{rezolvă}(X, f_1(X))) \vee \text{apreciază}(f_2(X), X))$
2. **Elimină cuantificatorii universali**
($\text{laborator}(f_1(X)) \wedge \neg \text{rezolvă}(X, f_1(X))$) \vee $\text{apreciază}(f_2(X), X)$
3. **Distribuie \vee prin \wedge**
($\text{laborator}(f_1(X)) \vee \text{apreciază}(f_2(X), X)$) \wedge ($\neg \text{rezolvă}(X, f_1(X)) \vee \text{apreciază}(f_2(X), X)$)
4. **Transformă expresiile în clauze**
{ $\text{laborator}(f_1(X))$, $\text{apreciază}(f_2(X), X)$ }
{ $\neg \text{rezolvă}(X, f_1(X))$, $\text{apreciază}(f_2(X), X)$ }

67

67

Logica cu predicate de ordinul întâi – Cuprins

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

68

68

Unificare

Unificarea a 2 expresii = găsirea celei mai generale substituții

$$S = \{e_1 / var_{1f}, e_2 / var_{2f}, \dots, e_n / var_{nf}\}$$

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

Utilitate

$$\frac{\{-cal(X), m\ddot{a}n\ddot{a}nc\ddot{a}(X, ov\ddot{a}z)\}}{\{cal(X)\}} \xrightarrow{\text{Rezoluție}} m\ddot{a}n\ddot{a}nc\ddot{a}(X, ov\ddot{a}z)$$

$$\frac{\{-cal(X), m\ddot{a}n\ddot{a}nc\ddot{a}(X, ov\ddot{a}z)\}}{\{cal(rocinante)\}} \xrightarrow{\text{Rezoluție}} ?$$

Mulțumită unificării, putem conchide și că Rocinante mănâncă ovăz.

69

69

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)

70

70

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)

71

71

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)

72

72

Unificare – Exemple

- P = mănâncă(vivi, X)
Q = hrănește(vivi, puiv) 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 }
Așa Da →	subst(S,P) = mănâncă(tata(X), tata(tata(X)))
Așa NU →	subst(S,Q) = mănâncă(tata(X), tata(tata(X)))
	subst(S,P) = mănâncă(tata(X), bunicul(X))

73

73

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

74

74

Rezoluție în forma cea mai generală

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

Exemplu

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

75

75

Rezoluție pentru clauze Horn

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

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

76

76

Comparație LP – LPOI

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

77

77

Comparație LP – LPOI

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

78

78

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		

79

79

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		

80

80

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		

81

81

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

82

82

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

83

83

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

84

84

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

85

85

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

86

86

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

87

87

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

88

88

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

89

89

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

90

90

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

91

91

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

92

92

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

93

93

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

94

94

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

95

95

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

96

96

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

97