Paradigme de Programare

S.l. dr. ing. Andrei Olaru
slides: Mihnea Muraru si Andrei Olaru

Catedra de Calculatoare

2013 – 2013, semestrul 2

Cursul 9

Logica cu predicate de ordinul I

Cuprins

Introducere

Logică

- formalism simbolic pentru reprezentarea faptelor și raționament.
- se bazează pe ideea de valoare de adevăr – e.g. *Adevărat* sau *Fals*.
- permite realizarea de argumente (argumentare) și demonstrații – deducție, inducție, rezoluție, etc.

Programare logică

- program scris folosind propoziții logice (clauze Horn pentru Prolog);
- mediul de execuție poate folosi propozițiile pentru a demonstra teoreme sau pentru a deduce fapte.
- pentru a înțelege cum funcționează programele scrise într-un limbaj de programare logică trebuie să înțelegem
  - ce sunt propozițiile, ce înseamnă și cum pot fi ele reprezentate;
  - cum funcționează procesele teoretice pe care se bazează mediul de execuție.

Logica propoziţională

- Cadru pentru:
  - descrierea proprietăţilor obiectelor, prin intermediul unui limbaj, cu o semantică asociată;
  - deducerea de noi proprietăţi, pe baza celor existente.
- Expresia din limbaj: propoziţia, corespunzătoare unei afirmaţii, ce poate fi adevărată sau falsă.
- Exemplu: “Profesorul vorbește și studenții ascultă.”
- Accepţii asupra unei propoziţii:
  - secvenţa de simboluri utilizate (abordarea aleasă) sau
  - înţelesul propriu-zis al acesteia, într-o interpretare.
- Valoarea de adevăr a unei propoziţii determinată de valorile de adevăr ale propoziţiilor constituente.

Sintaxă

- 2 categorii de propoziţii
  - simple $\rightarrow$ fapte atomice: “Profesorul vorbește.”, “Studenții ascultă.”
  - compuse $\rightarrow$ relaţii între propoziţii mai simple: “Telefonul sună şi câinele latră.”
- Propoziţii simple: $p, q, r, \ldots$
- Negaţii: $\neg\alpha$
- Conjuncţii: $(\alpha \wedge \beta)$
- Disjuncţii: $(\alpha \vee \beta)$
- Implicaţii: $(\alpha \Rightarrow \beta)$
- Echivalenţe: $(\alpha \Leftrightarrow \beta)$

Semantică

- Scop: dezvoltarea unor mecanisme de prelucrare, aplicabile independent de valoarea de adevăr a propoziţiilor, într-o situaţie particulară.

- Accent pe relaţiile între propoziţiile compuse şi cele constituente.

- Pentru explicitarea legăturilor $\rightarrow$ utilizarea conceptului de interpretare.

Definiția 33.1 (Interpretare).

Mulţime de asocieri între fiecare propoziţie simplă din limbaj şi o valoare de adevăr.

Exemplul 33.2.

Interpretarea I:

- $p^I = false$
- $q^I = true$
- $r^I = false$

Interpretarea J:

- $p^J = true$
- $q^J = true$
- $r^J = true$

- Sub o interpretare *fixată* $\rightarrow$ dependenţa valorii de adevăr a unei propoziţii compuse de valorile de adevăr ale celor constituente
- Negaţie: $(\neg\alpha)^I = \begin{cases} \textit{true} & \text{dacă } \alpha^I = \textit{false} \\ \textit{false} & \text{altfel} \end{cases}$
- Conjuncţie:
  $(\alpha \wedge \beta)^I = \begin{cases} \textit{true} & \text{dacă } \alpha^I = \textit{true} \text{ şi } \beta^I = \textit{true} \\ \textit{false} & \text{altfel} \end{cases}$
- Disjuncţie:
  $(\alpha \vee \beta)^I = \begin{cases} \textit{false} & \text{dacă } \alpha^I = \textit{false} \text{ şi } \beta^I = \textit{false} \\ \textit{true} & \text{altfel} \end{cases}$

Semantică

Propoziții compuse (2)

- Implicaţie:

$$(\alpha \vee \beta)^I = \begin{cases} \textit{false} & \text{dacă } \alpha^I = \textit{true} \text{ şi } \beta^I = \textit{false} \\ \textit{true} & \text{altfel} \end{cases}$$

- Echivalenţă: $(\alpha \vee \beta)^I = \begin{cases} \textit{true} & \text{dacă } \alpha^I = \beta^I \\ \textit{false} & \text{altfel} \end{cases}$

Evaluare

Cum determinăm valoarea de adevăr

Definiția 33.3 (Evaluare).

Determinarea valorii de adevăr a unei propoziţii, sub o interpretare, prin aplicarea regulilor semantice anterioare.

Exemplul 33.4.

- Interpretarea I:
  - $p^I = false$
  - $q^I = true$
  - $r^I = false$
- Propoziţia: $\phi = (p \wedge q) \vee (q \Rightarrow r)$
  $\phi^I = (false \wedge true) \vee (true \Rightarrow false) = false \vee false = false$

Satisfiabilitate

Valoare de adevăr peste mai multe interpretări

Definiția 33.5 (Satisfiabilitate).

Proprietatea unei propoziţii care este adevărată sub cel puţin o interpretare. Acea interpretare satisface propoziţia.

Exemplul 33.6 (Metoda tabelei de adevăr).

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

Validitate

Definiția 33.7 (Validitate).

Proprietatea unei propoziţii care este adevărată în toate interpretările. Propoziţia se mai numeşte tautologie.

Exemplul 33.8 (Validitate).

Propoziţia $p \vee \neg p$ este adevărată, indiferent de valoarea de adevăr a lui p, deci este validă.

- Verificabilă prin metoda tabelei de adevăr.

Nesatisfiabilitate

Definiția 33.9 (Nesatisfiabilitate).

Proprietatea unei propoziţii care este falsă în toate interpretările. Propoziţia se mai numeşte contradicţie.

Exemplul 33.10 (Nesatisfiabilitate).

Propoziţia $p \Leftrightarrow \neg p$ este falsă, indiferent de valoarea de adevăr a lui p, deci este nesatisfiabilă.

- Verificabilă prin metoda tabelei de adevăr.

Definiția 33.11 (Derivabilitate logică).

Proprietatea unei propoziţii de a reprezenta consecinţa logică a unei mulţimi de alte propoziţii, numite premise. Mulţimea de propoziţii Δ derivă propoziţia ϕ, fapt notat prin $\Delta \models \phi$, dacă şi numai dacă orice interpretare care satisface toate propoziţiile din Δ satisface şi ϕ.

Exemplul 33.12.

- $\{p\} \models p \vee q$
- $\{p, q\} \models p \wedge q$
- $\{p\} \not\models p \wedge q$
- $\{p, p \Rightarrow q\} \models q$

Derivabilitate
Verificare

- Verificabilă prin metoda tabelei de adevăr: toate intrările pentru care premisele sunt adevărate trebuie să inducă adevărul concluziei.

Exemplul 33.13.

Demonstrăm că $\{p, p \Rightarrow q\} \models q$.

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

Singura intrare în care ambele premise, p şi $p \Rightarrow q$, sunt adevărate, precizează şi adevărul concluziei, q.

Formulări echivalente ale derivabilităţii

- $\{\phi_1, \ldots, \phi_n\} \models \phi$

  sau

- Propoziţia $\phi_1 \wedge \ldots \wedge \phi_n \Rightarrow \phi$ este validă

  sau

- Propoziţia $\phi_1 \wedge \ldots \wedge \phi_n \wedge \neg\phi$ este nesatisfiabilă

Inferență

Motivaţie

- Derivabilitate logică → proprietate a propoziţiilor.
- Derivare mecanică (inferenţă) → demers de calcul, în scopul verificării derivabilităţii logice.
- Creşterea exponenţială a numărului de interpretări în raport cu numărul de propoziţii simple.
- De aici, diminuarea valorii practice a metodelor semantice, precum cea a tabelei de adevăr.
- Alternativ, metode sintactice, care manipulează doar reprezentarea simbolică.

Definiția 33.14 (Inferenţă).

Derivarea mecanică a concluziilor unui set de premise.

Definiția 33.15 (Regulă de inferenţă).

Procedură de calcul capabilă să deriveze concluziile unui set de premise. Derivabilitatea mecanică a concluziei ϕ din mulţimea de premise Δ, utilizând regula de inferenţă *inf*, se notează $\Delta \vdash_{inf} \phi$.

Reguli de inferenţă

- Şabloane parametrizate de raţionament, formate dintr-o mulţime de premise şi o mulţime de concluzii.
- exemplu: *Modus Ponens* (MP):

$$\frac{\alpha \Rightarrow \beta \qquad \alpha}{\beta}$$

- exemplu: *Modus Tollens*:

$$\frac{\alpha \Rightarrow \beta \qquad \neg\beta}{\neg\alpha}$$

Definiția 33.16 (Consistenţă (*soundness*)).

Regula de inferenţă determină doar propoziţii care sunt, într-adevăr, consecinţe logice ale premiselor. Echivalent, $\Delta \vdash_{inf} \phi \Rightarrow \Delta \models \phi$.

Definiția 33.17 (Completitudine (*completeness*)).

Regula de inferenţă determină toate consecinţele logice ale premiselor. Echivalent, $\Delta \models \phi \Rightarrow \Delta \vdash_{inf} \phi$.

- Ideal, ambele proprietăţi – “nici în plus, nici în minus”.
- Incompletitudinea regulii *Modus Ponens*, din imposibilitatea scrierii oricărei propoziţii ca implicaţie.

Axiome

- Exemplu: verificarea că $\{p \Rightarrow q, q \Rightarrow r\} \vdash p \Rightarrow r$
- Caz în care premisele sunt insuficiente pentru aplicarea regulilor de inferenţă.
- Soluţia: axiome – reguli de inferenţă fără premise.
- Exemplu de set de axiome:

$\alpha \Rightarrow (\beta \Rightarrow \alpha)$	Introducerea "$\Rightarrow$"
$(\alpha \Rightarrow (\beta \Rightarrow \gamma)) \Rightarrow ((\alpha \Rightarrow \beta) \Rightarrow (\alpha \Rightarrow \gamma))$	Distribuirea "$\Rightarrow$"
$(\neg\alpha \Rightarrow \neg\beta) \Rightarrow (\beta \Rightarrow \alpha)$	Inversarea "$\Rightarrow$"

Definiția 33.18 (Demonstraţie).

Secvenţă de propoziţii, finalizată cu o concluzie, conţinând:

- premise
- instanţe ale axiomelor
- rezultate ale aplicării regulilor de inferenţă asupra elementelor precedente din secvenţă.

Definiția 33.19 (Teoremă).

Concluzia cu care se termină o demonstraţie.

Definiția 33.20 (Procedură de demonstrare).

Mecanism constând din (1) o mulţime de reguli de inferenţă și (2) o strategie de control, ce dă ordinea aplicării regulilor.

Demonstraţii

Exemplu

Exemplul 33.21.

Demonstrăm că $\{p \Rightarrow q, q \Rightarrow r\} \vdash p \Rightarrow r$.

1	$p \Rightarrow q$	Premisă
2	$q \Rightarrow r$	Premisă
3	$(q \Rightarrow r) \Rightarrow (p \Rightarrow (q \Rightarrow r))$	Introd. "$\Rightarrow$"
4	$p \Rightarrow (q \Rightarrow r)$	MP (3), (2)
5	$(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$	Distrib "$\Rightarrow$"
6	$(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$	MP (5), (4)
7	$p \Rightarrow r$	MP 6, 1

Demonstraţii

Necesități

- Existenţa unui sistem de inferenţă consistent şi complet, bazat pe:
  - axiomele de mai devreme.
  - regula de inferenţă *Modus Ponens*.

$$\Delta \models \phi \Leftrightarrow \Delta \vdash \phi$$

Rezoluţie

O regulă de inferență mai bună

- Regulă de inferenţă foarte puternică.
- Baza unui demonstrator de teoreme consistent şi complet.
- Spaţiul de căutare mult mai mic ca în abordarea standard (vezi mai sus).
- Se bazează pe lucrul cu propoziţii în forma clauzală.

Forma clauzală

Definiții

Definiția 33.22 (Literal).

Propoziţie simplă sau negaţia ei. E.g. p şi $\neg p$.

Definiția 33.23 (Expresie clauzală).

Literal sau disjuncţie de literali. E.g. $p \vee \neg q \vee r$.

Definiția 33.24 (Clauză).

Mulţime de literali dintr-o expresie clauzală. E.g. $\{p, \neg q, r\}$.

Definiția 33.25 (Forma clauzală – CNF).

Reprezentarea unei propoziţii sub forma unei mulţimi de clauze, implicit legate prin conjuncţii.

Forma clauzală
Exemplu

Exemplul 33.26 (FNC).

Forma clauzală a propoziţiei

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

este

$\{p\}, \{\neg q, r\}, \{\neg p, \neg r\}$.

Forma clauzală
Obținere

- Orice propoziţie convertibilă în această formă astfel:

1. Eliminarea implicaţiilor:
   $$\alpha \Rightarrow \beta \;\rightarrow\; \neg\alpha \vee \beta$$
2. Avansarea negaţiilor până la literali:
   $$\neg(\alpha \wedge \beta) \;\rightarrow\; \neg\alpha \vee \neg\beta,\ \neg(\alpha \vee \beta) \;\rightarrow\; \neg\alpha \wedge \neg\beta,$$
   $$\neg(\neg\alpha) \;\rightarrow\; \alpha$$
3. Distribuirea lui $\vee$ faţă de $\wedge$:
   $$\alpha \vee (\beta \wedge \gamma) \;\rightarrow\; (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$
4. Transformarea expresiilor în clauze:
   $$\phi_1 \vee \ldots \vee \phi_n \;\rightarrow\; \{\phi_1, \ldots, \phi_n\}$$
   $$\phi_1 \wedge \ldots \wedge \phi_n \;\rightarrow\; \{\phi_1\}, \ldots, \{\phi_n\}$$

Forma clauzală
Obținere – Exemplu

Exemplul 33.27.

Transformăm propoziţia $p \wedge (q \Rightarrow r)$ în formă clauzală.

1.	$p \wedge (\neg q \vee r)$	Eliminare implicații
2.	$\{p\}, \{\neg q, r\}$	Formare clauze

Exemplul 33.28.

Transformăm propoziţia $\neg(p \wedge (q \Rightarrow r))$ în formă clauzală.

1.	$\neg(p \wedge (\neg q \vee r))$	Eliminare implicații
2.	$\neg p \vee \neg(\neg q \vee r)$	Împinge negații (1)
3.	$\neg p \vee (q \wedge \neg r)$	Împinge negații (2,3)
4.	$(\neg p \vee q) \wedge (\neg p \vee \neg r)$	Distribuire $\vee$
5.	$\{\neg p, q\}, \{\neg p, \neg r\}$	Formare clauze

Rezoluţie

Principiu de bază

- Ideea:

$$\frac{\{p \Rightarrow q\} \quad \{\neg p \Rightarrow r\}}{\{q, r\}}$$

- "Anularea" lui p
- p adevărată $\rightarrow \neg p$ falsă $\rightarrow r$ adevărată
- p falsă $\rightarrow q$ adevărată
- Cel puţin una dintre q şi r adevărată
- Forma generală:

$$\frac{\{p_1, \ldots, r, \ldots, p_m\} \quad \{q_1, \ldots, \neg r, \ldots, q_n\}}{\{p_1, \ldots, p_m, q_1, \ldots, q_n\}}$$

Rezoluţie

Cazuri speciale

- Clauza vidă $\rightarrow$ indicator de contradicţie între premise

$$\frac{\begin{array}{l}\{\neg p\}\\ \{p\}\end{array}}{\{\}}$$

- Mai mult de 2 rezolvenţi posibili (se alege doar unul):

$$\frac{\begin{array}{l}\{p, q\}\\ \{\neg p, \neg q\}\end{array}}{\begin{array}{l}\{p, \neg p\}\ \mathit{sau}\\ \{q, \neg q\}\end{array}}$$

Rezoluţie

Alte reguli de inferență – cazuri particulare ale rezoluției

- *Modus Ponens*:

$$\frac{\begin{array}{l} p \Rightarrow q \\ p \end{array}}{q} \qquad \sim \qquad \frac{\begin{array}{l} \{\neg p, q\} \\ \{p\} \end{array}}{\{q\}}$$

- *Modus Tollens*

$$\frac{\begin{array}{l} p \Rightarrow q \\ \neg q \end{array}}{\neg p} \qquad \sim \qquad \frac{\begin{array}{l} \{\neg p, q\} \\ \{\neg q\} \end{array}}{\{\neg p\}}$$

- Tranzitivitatea implicaţiei:

$$\frac{\begin{array}{l} p \Rightarrow q \\ q \Rightarrow r \end{array}}{p \Rightarrow r} \qquad \sim \qquad \frac{\begin{array}{l} \{\neg p, q\} \\ \{\neg q, r\} \end{array}}{\{\neg p, r\}}$$

Rezoluţie

Observații

- Demonstrarea nesatisfiabilităţii $\rightarrow$ derivarea clauzei vide.
- Demonstrarea derivabilităţii concluziei ϕ din premisele $\phi_1, \ldots, \phi_n \rightarrow$ demonstrarea nesatisfiabilităţii propoziţiei $\phi_1 \wedge \ldots \wedge \phi_n \wedge \neg\phi$.
- Demonstrarea validităţii propoziţiei $\phi \rightarrow$ demonstrarea nesatisfiabilităţii propoziţiei $\neg\phi$.
- Rezoluţia $\rightarrow$ incompletă generativ, i.e. concluziile nu pot fi derivate direct, răspunsul fiind dat în raport cu o "întrebare" fixată.

Rezoluţie

Exemplu

Exemplul 33.29.

Demonstrăm că $\{p \Rightarrow q, q \Rightarrow r\} \vdash p \Rightarrow r$, i.e. mulţimea $\{p \Rightarrow q, q \Rightarrow r, \neg(p \Rightarrow r)\}$ conţine o contradicţie.

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 $\rightarrow$ clauza vidă

Teorema 33.30 (Rezoluţiei).

Rezoluţia propoziţională este consistentă şi completă, i.e. $\Delta \models \phi \Leftrightarrow \Delta \vdash_{rez} \phi$.

- Terminarea garantată a procedurii de aplicare a rezoluţiei: număr finit de clauze $\rightarrow$ număr finit de concluzii.

Logica cu predicate de ordinul I

Logica cu predicate de ordinul I

First Order Logic (FOL) – Context

- Extensie a logicii propoziţionale, cu explicitarea:
  - obiectelor din universul problemei;
  - relaţiilor dintre acestea.
- Logica propoziţională:
  - p: “Andrei este prieten cu Bogdan.”
  - q: “Bogdan este prieten cu Andrei.”
  - $p \Leftrightarrow q$

  $\rightarrow$ Opacitate în raport cu obiectele şi relaţiile referite.
- FOL:
  - Generalizare: $prieten(x, y)$: “x este prieten cu y.”
  - $\forall x.\forall y.(prieten(x, y) \Leftrightarrow prieten(y, x))$

  $\rightarrow$ Aplicare pe cazuri particulare.

  $\rightarrow$ Transparenţă în raport cu obiectele şi relaţiile referite.

Sintaxă
Simboluri utilizate

- Constante: obiecte particulare din universul discursului: *c*, *d*, *andrei*, *bogdan*, ...
- Variabile: obiecte generice: *x*, *y*, ...
- Simboluri funcţionale: *succesor*, $+$, ...
- Simboluri relaţionale (predicate): relaţii *n*-are peste obiectele din universul discursului: $prieten = \{(andrei, bogdan), (bogdan, andrei), \ldots\}$, $impar = \{1, 3, \ldots\}$, ...
- Conectori logici: $\neg$, $\wedge$, ...
- Cuantificatori: $\forall$, $\exists$

Sintaxă
Termeni

- Termeni (obiecte):
  - Constante;
  - Variabile;
  - Aplicaţii de funcţii: $f(t_1, \ldots, t_n)$, unde f este un simbol funcţional n-ar şi $t_1, \ldots, t_n$ sunt termeni.
    Exemple:
    - *succesor*(4): succesorul lui 4, şi anume 5.
    - $+(2, x)$: aplicaţia funcţiei de adunare asupra numerelor 2 şi x, şi, totodată, suma lor.

Sintaxă
Atomi

- Atomi (relaţii): atomul $p(t_1, \ldots, t_n)$, unde p este un predicat n-ar şi $t_1, \ldots, t_n$ sunt termeni.
  Exemple:
  - $impar(3)$
  - $varsta(ion, 20)$
  - $=(+(2, 3), 5)$

- Propoziţii (fapte) – dacă x variabilă, A atom, și α și β propoziții, atunci o propoziție are forma:
  - Fals, adevărat: $\perp$, $\top$
  - Atomi: A
  - Negaţii: $\neg\alpha$
  - Conectori: $\alpha \wedge \beta$, $\alpha \Rightarrow \beta$, ...
  - Cuantificări: $\forall x.\alpha$, $\exists x.\alpha$

Exemplul 34.1.

"Dan este prieten cu sora Ioanei"

$$\underbrace{\mathit{prieten}(\underbrace{\mathit{dan}}_{\text{termen}}, \underbrace{\mathit{sora}(\overbrace{\mathit{ioana}}^{\text{termen}})}_{\text{termen}})}_{\text{atom/propoziţie}}$$

- Simplificare: legarea tuturor variabilelor, prin cuantificatori universali sau existenţiali
- Domeniul de vizibilitate al unui cuantificator $\rightarrow$ restul propoziţiei (v. simbolul λ în Calculul Lambda)

Definiția 34.2 (Interpretare).

O interpretare constă din:

- Un domeniu nevid, D
- Pentru fiecare constantă c, un element $c^I \in D$
- Pentru fiecare simbol funcţional, n-ar f, o funcţie $f^I : D^n \to D$
- Pentru fiecare predicat n-ar p, o funcţie $p^I : D^n \to \{false, true\}$.

Semantică
Elemente

- Atom:
  $(p(t_1, \ldots, t_n))^I = p^I(t_1^I, \ldots, t_n^I)$
- Negaţie, conectori, implicații: v. logica propoziţională
- Cuantificare universală:
  $$(\forall x.\alpha)^I = \begin{cases} \mathit{false} & \text{dacă } \exists d \in D \,.\, \alpha^I_{[d/x]} = \mathit{false} \\ \mathit{true} & \text{altfel} \end{cases}$$
- Cuantificare existenţială:
  $$(\exists x.\alpha)^I = \begin{cases} \mathit{true} & \text{dacă } \exists d \in D \,.\, \alpha^I_{[d/x]} = \mathit{true} \\ \mathit{false} & \text{altfel} \end{cases}$$

Exemplul 34.3.

1. "Vrabia mălai visează."

Exemple

Cu cuantificatori

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. "Unele vrăbii visează mălai."

Exemple

Cu cuantificatori

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. "Unele vrăbii visează mălai."
   $\exists x.(vrabie(x) \wedge viseaza(x, malai))$

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. "Unele vrăbii visează mălai."
   $\exists x.(vrabie(x) \wedge viseaza(x, malai))$
3. "Nu toate vrăbiile visează mălai."

Exemple

Cu cuantificatori

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. "Unele vrăbii visează mălai."
   $\exists x.(vrabie(x) \wedge viseaza(x, malai))$
3. "Nu toate vrăbiile visează mălai."
   $\exists x.(vrabie(x) \wedge \neg viseaza(x, malai))$

Exemplul 34.3.

1. “Vrabia mălai visează.” $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. “Unele vrăbii visează mălai.”
   $\exists x.(vrabie(x) \wedge viseaza(x, malai))$
3. “Nu toate vrăbiile visează mălai.”
   $\exists x.(vrabie(x) \wedge \neg viseaza(x, malai))$
4. “Nicio vrabie nu visează mălai.”

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. "Unele vrăbii visează mălai."
   $\exists x.(vrabie(x) \wedge viseaza(x, malai))$
3. "Nu toate vrăbiile visează mălai."
   $\exists x.(vrabie(x) \wedge \neg viseaza(x, malai))$
4. "Nicio vrabie nu visează mălai."
   $\forall x.(vrabie(x) \Rightarrow \neg viseaza(x, malai))$

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. "Unele vrăbii visează mălai."
   $\exists x.(vrabie(x) \wedge viseaza(x, malai))$
3. "Nu toate vrăbiile visează mălai."
   $\exists x.(vrabie(x) \wedge \neg viseaza(x, malai))$
4. "Nicio vrabie nu visează mălai."
   $\forall x.(vrabie(x) \Rightarrow \neg viseaza(x, malai))$
5. "Numai vrăbiile visează mălai."

Exemple

Cu cuantificatori

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. "Unele vrăbii visează mălai."
   $\exists x.(vrabie(x) \wedge viseaza(x, malai))$
3. "Nu toate vrăbiile visează mălai."
   $\exists x.(vrabie(x) \wedge \neg viseaza(x, malai))$
4. "Nicio vrabie nu visează mălai."
   $\forall x.(vrabie(x) \Rightarrow \neg viseaza(x, malai))$
5. "Numai vrăbiile visează mălai."
   $\forall x.(viseaza(x, malai) \Rightarrow vrabie(x))$

Exemple

Cu cuantificatori

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. "Unele vrăbii visează mălai."
   $\exists x.(vrabie(x) \wedge viseaza(x, malai))$
3. "Nu toate vrăbiile visează mălai."
   $\exists x.(vrabie(x) \wedge \neg viseaza(x, malai))$
4. "Nicio vrabie nu visează mălai."
   $\forall x.(vrabie(x) \Rightarrow \neg viseaza(x, malai))$
5. "Numai vrăbiile visează mălai."
   $\forall x.(viseaza(x, malai) \Rightarrow vrabie(x))$
6. "Toate şi numai vrăbiile visează mălai."

Exemple

Cu cuantificatori

Exemplul 34.3.

1. "Vrabia mălai visează." $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
2. "Unele vrăbii visează mălai."
   $\exists x.(vrabie(x) \wedge viseaza(x, malai))$
3. "Nu toate vrăbiile visează mălai."
   $\exists x.(vrabie(x) \wedge \neg viseaza(x, malai))$
4. "Nicio vrabie nu visează mălai."
   $\forall x.(vrabie(x) \Rightarrow \neg viseaza(x, malai))$
5. "Numai vrăbiile visează mălai."
   $\forall x.(viseaza(x, malai) \Rightarrow vrabie(x))$
6. "Toate şi numai vrăbiile visează mălai."
   $\forall x.(viseaza(x, malai) \Leftrightarrow vrabie(x))$

Cuantificatori

Greşeli frecvente

- $\forall x.(vrabie(x) \Rightarrow viseaza(x, malai))$
  $\rightarrow$ corect: “Toate vrăbiile visează mălai.”

- $\forall x.(vrabie(x) \wedge viseaza(x, malai))$
  $\rightarrow$ greşit: “Toţi sunt vrăbii care visează mălai.”

- $\exists x.(vrabie(x) \wedge viseaza(x, malai))$
  $\rightarrow$ corect: “Unele vrăbii visează mălai.”

- $\exists x.(vrabie(x) \Rightarrow viseaza(x, malai))$
  $\rightarrow$ greşit: adevărată şi dacă există cineva care nu este vrabie.

Cuantificatori
Proprietăţi

- **Necomutativitate**:
  - $\forall x.\exists y.viseaza(x,y) \rightarrow$ "Toţi visează la ceva anume."
  - $\exists x.\forall y.viseaza(x,y) \rightarrow$ "Există cineva care visează la orice."
- **Dualitate**:
  - $\neg(\forall x.\alpha) \equiv \exists x.\neg\alpha$
  - $\neg(\exists x.\alpha) \equiv \forall x.\neg\alpha$

Aspecte legate de propoziţii

Analoage logicii propoziţionale

- Satisfiabilitate
- Validitate
- Derivabilitate
- Inferenţă
- Demonstraţie

Forme normale

Definiții (1)

Definiția 34.4 (Literal).

Atom sau negaţia lui. Exemplu: $prieten(x,y)$, $\neg prieten(x,y)$.

Definiția 34.5 (Expresie clauzală).

Literal sau disjuncţie de literali.
Exemplu: $prieten(x,y) \vee \neg doctor(x)$.

Definiția 34.6 (Clauză).

Mulţime de literali dintr-o expresie clauzală. Exemplu: $\{prieten(x,y), \neg doctor(x)\}$.

Forme normale
Definiții (2)

Definiția 34.7 (Forma clauzală / Forma normală conjunctivă – FNC).

Reprezentarea unei propoziţii sub forma unei mulţimi de clauze, implicit legate prin conjuncţii.

Definiția 34.8 (Forma normală implicativă – FNI).

Reprezentarea unei propoziţii sub forma unei mulţimi de clauze, implicit legate prin conjuncţii, în care fiecare clauză are forma grupată
$\{\neg A_1, \ldots, \neg A_m, B_1, \ldots, B_n\}$,
corespunzătoare implicaţiei
$(A_1 \wedge \ldots \wedge A_m) \Rightarrow (B_1 \vee \ldots \vee B_n)$, unde A_i şi B_j sunt atomi.

Definiția 34.9 (Clauză Horn).

Clauză în care un singur literal este în formă pozitivă:
$\{\neg A_1, \ldots, \neg A_n, A\}$,
corespunzătoare implicaţiei
$A_1 \wedge \ldots \wedge A_n \Rightarrow A$.

Exemplul 34.10 (Clauze Horn).

Transformarea propoziţiei
$vrabie(x) \vee ciocarlie(x) \Rightarrow pasare(x)$ în forme normale, utilizând clauze Horn:

- FNC: $\{\neg vrabie(x), pasare(x)\}, \{\neg ciocarlie(x), pasare(x)\}$
- FNI: $vrabie(x) \Rightarrow pasare(x), ciocarlie(x) \Rightarrow pasare(x)$

Conversia propoziţiilor în FNC (1)

Eliminare implicații, împingere negații, redenumiri

1. Eliminarea implicaţiilor ($\not\Rightarrow$)
2. Împingerea negaţiilor până în fața literalilor ($\overrightarrow{\neg}$)
3. Redenumirea variabilelor cuantificate pentru obţinerea unicităţii de nume (R):

$$\forall x.p(x) \wedge \forall x.q(x) \vee \exists x.r(x) \rightarrow \forall x.p(x) \wedge \forall y.q(y) \vee \exists z.r(z)$$

4. Deplasarea cuantificatorilor la începutul expresiei, conservându-le ordinea (forma normală *prenex*) (P):

$$\forall x.p(x) \wedge \forall y.q(y) \vee \exists z.r(z) \rightarrow \forall x.\forall y.\exists z.(p(x) \wedge q(y) \vee r(z))$$

Conversia propoziţiilor în FNC (2)

Skolemizare

5. Eliminarea cuantificatorilor existenţiali (skolemizare) (S):

   - Dacă nu este precedat de cuantificatori universali: înlocuirea apariţiilor variabilei cuantificate printr-o constantă:
   $$\exists x.p(x) \rightarrow p(c_x)$$
   - Dacă este precedat de cuantificatori universali: înlocuirea apariţiilor variabilei cuantificate prin aplicaţia unei funcţii unice asupra variabilelor anterior cuantificate universal:
   $$\forall x.\forall y.\exists z.(p(x) \wedge q(y) \vee r(z))$$
   $$\rightarrow \forall x.\forall y.(p(x) \wedge q(y) \vee r(f_z(x, y)))$$

Conversia propoziţiilor în FNC (3)

Cuantificatori universali, Distribuire $\vee$, Clauze

6. Eliminarea cuantificatorilor universali, consideraţi, acum, impliciţi ($\not\forall$):

$$\forall x.\forall y.(p(x) \wedge q(y) \vee r(f_z(x,y))) \rightarrow p(x) \wedge q(y) \vee r(f_z(x,y))$$

7. Distribuirea lui $\vee$ faţă de $\wedge$ ($\vee/\wedge$):

$$\alpha \vee (\beta \wedge \gamma) \rightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$$

8. Transformarea expresiilor în clauze (C).

Conversia propoziţiilor în FNC – Exemplu

Exemplul 34.11.

"Cine rezolvă toate laboratoarele este apreciat de cineva."

$$\forall x.(\forall y.(lab(y) \Rightarrow rezolva(x,y)) \Rightarrow \exists y.apreciaza(y,x))$$

$\not\Rightarrow$	$\forall x.(\neg\forall y.(\neg lab(y) \vee rezolva(x,y)) \vee \exists y.apreciaza(y,x))$
$\overrightarrow{\neg}$	$\forall x.(\exists y.\neg(\neg lab(y) \vee rezolva(x,y)) \vee \exists y.apreciaza(y,x))$
$\overrightarrow{\neg}$	$\forall x.(\exists y.(lab(y) \wedge \neg rezolva(x,y)) \vee \exists y.apreciaza(y,x))$
R	$\forall x.(\exists y.(lab(y) \wedge \neg rezolva(x,y)) \vee \exists z.apreciaza(z,x))$
P	$\forall x.\exists y.\exists z.((lab(y) \wedge \neg rezolva(x,y)) \vee apreciaza(z,x))$
S	$\forall x.((lab(f_y(x)) \wedge \neg rezolva(x,f_y(x))) \vee apreciaza(f_z(x),x))$
$\not\forall$	$(lab(f_y(x)) \wedge \neg rezolva(x,f_y(x))) \vee apreciaza(f_z(x),x)$
$\vee/\wedge$	$(lab(f_y(x)) \vee apr(f_z(x),x)) \wedge (\neg rez(x,f_y(x)) \vee apr(f_z(x),x))$
C	$\{lab(f_y(x)), apr(f_z(x),x)\}, \{\neg rez(x,f_y(x)), aprc(f_z(x),x)\}$

Unificare
Motivaţie

- Rezoluţie:

$$\frac{\begin{array}{l}\{prieten(x, mama(y)), doctor(x)\}\\ \{\neg prieten(mama(z), z)\}\end{array}}{?}$$

- Cum aplicăm rezoluţia?
- Soluţia: unificare (vezi sinteza de tip – Def. 26.5, 26.6, 26.8)
- MGU (Most General Unifier):
  $S = \{x \leftarrow mama(z), z \leftarrow mama(y)\}$
- Forma comună a celor doi atomi:
  $prieten(mama(mama(y)), mama(y))$
- Rezolvent: $doctor(mama(mama(y)))$

Unificare
Observații

- Problemă NP-completă;
- Posibile legări ciclice;
- Exemplu:
  $prieten(x, mama(x))$ şi $prieten(mama(y), y)$
  MGU: $S = \{x \leftarrow mama(y), y \leftarrow mama(x)\}$
  $\Rightarrow x \leftarrow mama(mama(x)) \rightarrow$ imposibil!
- Soluţie: verificarea apariţiei unei variabile în valoarea la care a fost legată (*occurrence check*);

Unificare
Rolul în rezoluție

- Rezoluţia pentru clauze Horn:

$$\frac{\begin{array}{l} A_1 \wedge \ldots \wedge A_m \Rightarrow A \\ B_1 \wedge \ldots \wedge A' \wedge \ldots \wedge B_n \Rightarrow B \\ unificare(A, A') = S \end{array}}{subst(S, A_1 \wedge \ldots \wedge A_m \wedge B_1 \wedge \ldots \wedge B_n \Rightarrow B)}$$

- $unificare(\alpha, \beta) \rightarrow$ substituţia sub care unifică propoziţiile α şi β;

- $subst(S, \alpha) \rightarrow$ propoziţia rezultată în urma aplicării substituţiei S asupra propoziţiei α.

Sfârșitul cursului 9

Ce am învățat

· Bazele logicii propoziționale și cu predicate de ordinul I, Sintaxă și Semantică, Forme normale, Demonstrații, Axiome, Rezoluția ca regulă de inferență, Unificare.