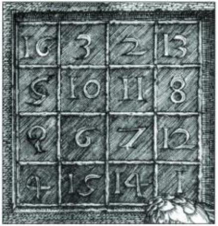
Logică și Structuri Discrete -LSD



Cursul 12 – Logica predicatelor II

dr. ing. Cătălin Iapă
catalin.iapa@cs.upt.ro

Problemă de logică

Gasiti o intrebare la care raspunzi tot timpul cu ”NU” pentru ca nu ai cum sa raspunzi cu ”DA”.

Problemă de logică

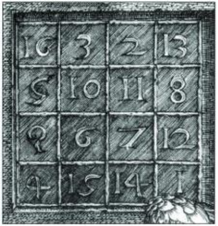
Un profesor de logică se întoarce dintr-o călătorie de afaceri cu 100 de monede pe care să le împartă celor doi copii ai săi. El pune monedele pe o masă cu 60 de monede cu stema în sus, iar restul cu banul în sus. Apoi stinge lumina astfel încât să fie complet întuneric. Îi spune fiului său că poate rearanja monedele de pe masă (le poate muta, le poate învârti etc.) iar în final trebuie să le împartă în două grupuri. Apoi îi spune fiicei sale că poate decide care dintre cele două grupuri este al ei și care este al fratelui ei. Ei vor aprinde apoi din nou lumina și fiecare copil poate păstra doar monedele din grupul său care au banul în sus. (Tata va păstra toate monedele cu stema în sus.)

Când lumina este stinsă, copiii nu pot vedea orientarea monedelor și este imposibil să distingem orientarea pipăind monedele. Băiețelul este hotărât să nu-și lase sora să „câștige” ajungând cu mai multe monede decât el, așa că vrea să împartă monedele în grupuri care să *garanteze* că, indiferent de grupul pe care sora lui îl alege, ea nu va ajunge cu mai multe monede decât el. Ce ar trebui să facă băiețelul?

Problemă de logică

Cum credeți că putem formaliza în Logica Predicatelor răspunsul la întrebarea:

Câte animale a urcat Moise pe arca sa?



Rezoluția în Logica predicatelor

Unificarea

Demonstrații în Logica Predicatelor

Semantica în Logica Predicatelor

Exerciții

Exemplu de rezoluție

$$
\begin{array}{l}
\quad a \\
\wedge (\neg a \vee b) \\
\wedge (\neg b \vee c) \\
\wedge (\neg a \vee \neg b \vee \neg c)
\end{array}
\qquad
\begin{array}{l}
c \text{ pozitiv} \\
c \text{ negat}
\end{array}
$$

Aplicăm rezoluția după c, avem o singură pereche de clauze:
$rez_c(\neg b \vee c,\ \neg a \vee \neg b \vee \neg c) = \neg b \vee \neg a \vee \neg b = \neg a \vee \neg b$

Eliminăm cele două clauze cu c și adăugăm clauza nouă:

$$
\begin{array}{l}
\quad a \\
\wedge (\neg a \vee b) \\
\wedge (\neg a \vee \neg b)
\end{array}
$$

Aplicăm rezoluția după b:

$rez_b(\neg a \vee b, \neg a \vee \neg b) = \neg a \vee \neg a = \neg a$

Eliminăm cele două clauze cu b, adăugăm clauza nouă: $\begin{array}{l} \quad a \\ \wedge \neg a \end{array}$

Aplicăm rezoluția după a: $rez_a(a, \neg a)$ = F (clauza vidă) Deci formula inițială e o contradicție (e nerealizabilă).

Rezoluția în calculul predicatelor

Fie clauzele: A cu $P(\ldots)$ *pozitiv* și B, cu $\neg P(\ldots)$ (*negat*) Exemplu:

A: $P(x, g(y)) \vee P(h(a), z) \vee Q(z)$
B: $\neg P(h(z), t) \vee R(t, z)$

Alegem niște (≥ 1) $P(\ldots)$ din A și niște $\neg P(\ldots)$ din B. aici: toți

Redenumim variabilele comune (nu au legătură între A și B)
A: $P(x, g(y)) \vee P(h(a), z) \vee Q(z)$ B: $\neg P(h(z_2), t) \vee R(t, z_2)$

Unificăm (toți odată) doar acei $P(\ldots)$ din A și $\neg P(\ldots)$ din B aleși

$\{P(x, g(y)), P(h(a), z), P(h(z_2), t)\}$ $x \mapsto h(a)$; $z_2 \mapsto a$; $z, t \mapsto g(y)$

Eliminăm pe $P(\ldots)$ și $\neg P(\ldots)$ aleși din $A \vee B$. *Aplicăm substituția* rezultată *din unificare* și *adăugăm noua clauză* la lista clauzelor.

$$Q(g(y)) \vee R(g(y), a)$$

Păstrăm clauzele inițiale, se pot folosi cu alte alegeri de predicate.

Rezoluția: în concluzie

Generăm repetat clauze noi (*rezolvenți*) prin **rezoluție cu unificare**

Dacă repetând obținem *clauza vidă*, formula inițială e *nerealizabilă*.
Dacă *nu mai găsim rezolvenți noi*, formula inițială e *realizabilă*.

Reamintim: am pornit încercând să demonstrăm

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \rightarrow C$$

prin *reducere la absurd*, negând concluzia și arătând că

$$A_1 \wedge A_2 \wedge \ldots \wedge A_n \wedge \neg C \quad \text{e } \textit{contradicție}$$

Metoda rezoluției e *completă* relativ la refutație
pentru orice formulă nerealizabilă, va ajunge la clauza vidă
dar nu poate *determina* realizabilitatea *oricărei formule*
(există formule pentru care rulează la infinit)

Exemplu de aplicare a rezoluției

Reluăm exercițiul formalizat anterior.
Folosim () și nu . pentru a evita greșeli la aplicarea cuantificării.

A_1: $\forall X\,(inv\,(X) \to \exists C\,(cump(X, C) \wedge (act(C) \vee oblig\,(C))))$

A_2: $scadedj \to \forall X\,(act(X) \wedge \neg aur\,(X) \to scade(X))$

A_3: $creștedob \to \forall X\,(oblig\,(X) \to scade(X))$

A_4: $\forall X\,(inv\,(X) \to (\exists C\,(cump(X, C) \wedge scade(C)) \to \neg bucur\,(X)))$

C : $scadedj \wedge creștedob \to$
$\forall X\,(inv\,(X) \wedge bucur\,(X) \to \exists C\,(cump(X, C) \wedge act(C) \wedge aur\,(C)))$

Negăm concluzia *la început,* înainte de a transforma cuantificatorii!

$\neg C$: $\neg(scadedj \wedge creștedob \to$
$\forall X\,(inv\,(X) \wedge bucur\,(X) \to \exists C\,(cump(X, C) \wedge act(C) \wedge aur\,(C))))$

Eliminăm implicația, ducem negația până la predicate

1.Eliminăm implicația: $A \rightarrow B = \neg A \vee B$, $\neg(A \rightarrow B) = A \wedge \neg B$

Orice transformare *într-o formulă* NU afectează ce e în afara ei!

În $\forall x\, A$, transformând oricum *pe A* ($\rightarrow$, $\neg$, ...) NU se schimbă $\forall x$

2. Ducem $\neg$ *înăuntru*: $\neg\forall x P(x) = \exists x\, \neg P(x)$ $\quad \neg\exists x P(x) = \forall x\, \neg P(x)$

A_1: $\forall X\, (inv\, (X) \rightarrow \exists\, C\, (cump(X, C) \wedge (act(C) \vee oblig\, (C))))$
$\quad \forall X\, (\neg inv\, (X) \vee \exists\, C\, (cump(X, C) \wedge (act(C) \vee oblig\, (C))))$

A_2: $scadedj \rightarrow \forall\, X\, (act(X) \wedge \neg aur\, (X) \rightarrow scade(X))$
$\quad \neg scadedj \vee \forall\, X\, (\neg act(X) \vee aur\, (X) \vee scade(X))$

A_3: $creștedob \rightarrow \forall\, X\, (oblig\, (X) \rightarrow scade(X))$
$\quad \neg creștedob \vee \forall\, X\, (\neg oblig\, (X) \vee scade(X))$

A_4: $\forall X\, (inv\, (X) \rightarrow (\exists\, C\, (cump(X, C) \wedge scade(C)) \rightarrow \neg bucur\, (X)))$
$\quad \forall X\, (\neg inv\, (X) \vee \neg\exists\, C\, (cump(X, C) \wedge scade(C)) \vee \neg bucur\, (X))$
$\quad \forall X\, (\neg inv\, (X) \vee \forall\, C\, (\neg cump(X, C) \vee \neg scade(C)) \vee \neg bucur\, (X))$

Eliminăm implicația, ducem negația înăuntru (cont.)

$\neg C$: $\neg(scadedj \wedge creștedob \rightarrow$
$\forall X\,(inv(X) \wedge bucur(X) \rightarrow \exists C\,(cump(X, C) \wedge act(C) \wedge aur(C))))$

$\neg C$: $scadedj \wedge creștedob \wedge$
$\neg\forall\, X\,(inv(X) \wedge bucur(X) \rightarrow \exists C\,(cump(X, C) \wedge act(C) \wedge aur(C)))$

$scadedj \wedge creștedob \wedge$
$\exists X\,(inv(X) \wedge bucur(X) \wedge \neg\exists C\,(cump(X, C) \wedge act(C) \wedge aur(C)))$

$scadedj \wedge creștedob \wedge$
$\exists X\,(inv(X) \wedge bucur(X) \wedge \forall C\,(\neg cump(X, C) \vee \neg act(C) \vee \neg aur(C)))$

Redenumim: nume unice la variabile cuantificate

3. Dăm *nume unice* variabilelor cuantificate în fiecare formulă, pentru a putea elimina ulterior cuantificatorii. De exemplu:

$\forall x\, P(x) \vee \forall x\, \exists y\, Q(x, y)$ devine $\forall x\, P(x) \vee \forall z\, \exists y\, Q(z, y)$

Nu e nevoie în exemplul nostru:

A_1: $\forall X\, (\neg inv\, (X) \vee \exists\, C\, (cump(X, C) \wedge (act(C) \vee oblig\, (C))))$

A_2: $\neg scadedj \vee \forall X\, (\neg act(X) \vee aur\, (X) \vee scade(X))$

A_3: $\neg \text{creștedob} \vee \forall X\, (\neg oblig\, (X) \vee scade(X))$

A_4: $\forall X\, (\neg inv\, (X) \vee \forall C\, (\neg cump(X, C) \vee \neg scade(C)) \vee \neg bucur\, (X))$

$\neg C$: $scadedj \wedge \text{creștedob} \wedge$
$\exists X\, (inv(X) \wedge bucur(X) \wedge \forall C\, (\neg cump(X, C) \vee \neg act(C) \vee \neg aur(C)))$

Skolemizare: eliminăm cuantificatorii existențiali

4. *Skolemizare*: În $\forall x_1 \ldots \forall x_n \exists y$, alegerea lui y *depinde* de $x_1, \ldots x_n$; introducem o nouă *funcție Skolem* $y = g(x_1, \ldots, x_n)$, $\exists y$ dispare

A_1: $\forall X\, (\neg inv\,(X) \vee \exists C\, (cump(X, C) \wedge (act(C) \vee oblig\,(C))))$

C din $\exists$ depinde de X $\Rightarrow$ C devine *o nouă funcție* $f\,(X)$, $\exists C$ dispare

$\forall X\, (\neg inv\,(X) \vee (cump(X, f\,(X)) \wedge (act(f\,(X)) \vee oblig\,(f\,(X)))))$

Atenție! fiecare cuantificator $\exists$ primește o *nouă funcție* Skolem!

Pentru $\exists y$ în *exteriorul* oricărui $\forall$, alegem o nouă *constantă Skolem*

$\neg C$: $scadedj \wedge creștedob \wedge \exists X\, (inv\,(X) \wedge bucur\,(X)$
$\wedge \forall C\, (\neg cump(X, C) \vee \neg act(C) \vee \neg aur\,(C)))$

X devine o nouă *constantă* b (nu depinde de nimic), $\exists X$ dispare

$scadedj \wedge creștedob \wedge inv\,(b) \wedge bucur\,(b)$
$\wedge \forall C\, (\neg cump(b, C) \vee \neg act(C) \vee \neg aur\,(C))$

Forma normală prenex. Eliminăm cuantificatorii universali

5.Ducem *cuantificatorii universali în față*: *forma normală prenex*

A_4: $\forall X\,(\neg inv(X) \vee \forall C\,(\neg cump(X, C) \vee \neg scade(C)) \vee \neg bucur(X))$

$\forall X\,\forall C\,(\neg inv(X) \vee \neg cump(X, C) \vee \neg scade(C) \vee \neg bucur(X))$

6. Eliminăm cuantificatorii universali
(devin impliciți, o variabilă poate fi înlocuită cu orice termen).

A_1: $(\neg inv(X) \vee (cump(X, f(X)) \wedge (act(f(X)) \vee oblig(f(X)))))$

A_2: $\neg scadedj \vee \neg act(X) \vee aur(X) \vee scade(X)$

A_3: $\neg creștedob \vee \neg oblig(X) \vee scade(X)$

A_4: $\neg inv(X) \vee \neg cump(X, C) \vee \neg scade(C) \vee \neg bucur(X)$

$\neg C$: $scadedj \wedge creștedob \wedge inv(b) \wedge bucur(b)$
$\wedge(\neg cump(b, C) \vee \neg act(C) \vee \neg aur(C))$

Forma clauzală

7. Ducem *conjuncția în exteriorul* disjuncției (distributivitate) și scriem fiecare clauză separat (*formă clauzală*, CNF)

(1) $\neg inv(X) \vee cump(X, f(X))$

(2) $\neg inv(X) \vee act(f(X)) \vee oblig(f(X))$

(3) $\neg scadedj \vee \neg act(X) \vee aur(X) \vee scade(X)$

(4) $\neg creștedob \vee \neg oblig(X) \vee scade(X)$

(5) $\neg inv(X) \vee \neg cump(X, C) \vee \neg scade(C) \vee \neg bucur(X)$

(6) $scadedj$

(7) $creștedob$

(8) $inv(b)$

(9) $bucur(b)$

(10) $\neg cump(b, C) \vee \neg act(C) \vee \neg aur(C)$

Generăm rezolvenți până la clauza vidă

Căutăm predicate $P(\ldots)$ și $\neg P(\ldots)$ și unificăm, obținând rezolvenții:

(11) $\neg act(X) \vee aur(X) \vee scade(X)$ (3, 6)

(12) $\neg cump(b, C) \vee \neg act(C) \vee scade(C)$ (10, 11, $X = C$)

(13) $\neg oblig(X) \vee scade(X)$ (4, 7)

Când unificăm, redenumim clauzele să nu aibă variabile comune:
(13) $\neg oblig(Y) \vee scade(Y)$ vom unifica cu (2), redenumim X

(14) $\neg inv(X) \vee act(f(X)) \vee scade(f(X))$ (2, 13, $Y = X$)

(15) $\neg cump(b, f(X)) \vee \neg inv(X) \vee scade(f(X))$ (12, 14, $C = f(X)$)

(16) $\neg cump(b, C) \vee \neg scade(C) \vee \neg bucur(b)$ (5, 8, $X = b$)

(17) $\neg cump(b, C) \vee \neg scade(C)$ (9, 16)

(18) $\neg cump(b, f(X)) \vee \neg inv(X)$ (15, 17, $C = f(X)$)

(19) $\neg inv(b)$ (1, 18, $X = b$)

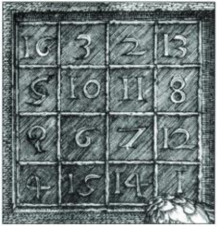
(20) $\varnothing$ (contradicție = succes în reducerea la absurd) (8, 19)

Ce putem face cu ce am învățat?

Putem traduce (*formaliza*) din limbaj natural în logica predicatelor

Putem *demonstra teoreme* prin reducere la absurd:

- negăm concluzia
- transformăm în *formă clauzală* (conjuncție $\wedge$ de disjuncții $\vee$)
- prin metoda *rezoluției* găsim o *contradicție* (clauza vidă)



Rezoluția în Logica predicatelor
Unificarea
Demonstrații în Logica Predicatelor
Semantica în Logica Predicatelor
Exerciții

Cum se face unificarea?

În logica predicatelor, rezoluția *unifică* un literal cu negatul lui:

$$A \vee P(t_1, t_2, \ldots, t_n) \quad \text{și} \quad B \vee \neg P(t_1', t_2', \ldots, t_n')$$

dacă putem unifica ("potrivi") argumentele lui P și $\neg P$: t_1 cu t_1',...

O *substituție* e o *funcție* care asociază unor *variabile* niște *termeni*:

$$\{x_1 \mapsto t_1, \ \ldots, x_n \mapsto t_n\}$$

Doi termeni se pot *unifica* dacă există o substituție care îi face egali

(o asemenea substituție se numește *unificator*)

$$f(x, g(y, z), t)\{x \mapsto h(z), y \mapsto h(b), t \mapsto u\} = f(h(z), g(h(b), z), u)$$

Termenul T cu substituția σ se notează uzual postfix: $T\sigma$

Substituția găsită se aplică predicatelor care rămân (rezolventul):

$$\frac{A \vee P(t_1, t_2, \ldots, t_n) \qquad B \vee \neg P(t_1', t_2', \ldots, t_n')}{A\sigma \vee B\sigma}$$

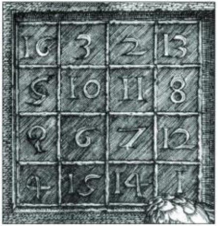
Reguli de unificare

O *variabilă* x poate fi unificată cu orice *termen* t (substituție) dacă x *nu apare* în t (altfel, substituind obținem un termen infinit)

deci nu: x cu $f(h(y), g(x, z))$; dar trivial, putem unifica x cu x

Doi *termeni* $f(\ldots)$ pot fi unificați doar dacă au aceeași funcție, și *argumentele* (termeni) pot fi unificate (poziție cu poziție)

Două *constante* (funcții cu 0 arg.) ⇒ unificate dacă sunt identice



Rezoluția în Logica predicatelor
Unificarea
Demonstrații în Logica Predicatelor
Semantica în Logica Predicatelor
Exerciții

Cât de generală e o demonstrație?

rezoluție:
$$\frac{\neg boy(x) \vee \neg girl(x) \vee child(x) \qquad boy(c)}{\neg girl(c) \vee child(c)}$$

Demonstrația e făcută *fără* a ține cont (sau înțelege) *semnificația* predicatelor *boy*, *child*, *good*, etc.:

puteau fi $P(x)$, $Q(x)$, $R(x)$, . . .

Demonstrația e valabilă pentru *orice predicate* care satisfac ipotezele.

Recapitulăm: sintaxa logicii predicatelor

Formulele logicii predicatelor sunt definite *structural recursiv*:

Termenii

variabilă v sau constantă c
$f(t_1, \cdot \cdot \cdot, t_n)$ cu f funcție n-ară și $t_1, \cdot \cdot \cdot, t_n$ *termeni*

Formule (well-formed formulasformule bine formate):

$P(t_1, \cdot \cdot \cdot, t_n)$	cu P predicat n-ar; $t_1, \cdot \cdot \cdot, t_n$ termeni
$\neg\alpha$	unde α este o formulă unde α,
$\alpha \rightarrow \beta$	β sunt formule
$\forall v\, \alpha$	cu v variabilă, α formulă: *cuantificare universală*
$t_1 = t_2$	cu t_1, t_2 termeni (în logica de ord. I cu egalitate)

În loc de propoziții avem *predicate* (peste *termeni*).

Sintaxă și semantică

Ca în logica propozițională (și pentru orice limbaj), deosebim

sintaxa = *forma*, *regulile* după care construim ceva (aici, formule)

semantica = *înțelesul* construcțiilor de limbaj

La fel ca în logica propozițională, lucrăm cu

deducția (demonstrația): procedeu pur sintactic

implicația / consecința logică (consecința semantică):

interpretăm formula (*înțelesul* ei, valoarea de adevăr)

Ne interesează corespondența dintre aceste două aspecte.

Reguli și ce înseamnă aplicarea lor

Regulile discutate sunt *sintactice*:
manipulează forma (*simboluri*, nu înțelesul lor).

Regulile lui deMorgan: $\neg(a \vee b) = \neg a \wedge \neg b$, $\neg(a \wedge b) = \neg a \vee \neg b$
Înlocuim o formă cu alta.
Rezultatul: formulele sunt echivalente

Regulă: Dacă un literal L e singur într-o clauză:
ștergem toate clauzele din care apare
ștergem $\neg L$ din toate clauzele

Rezultatul: dacă formula era realizabilă, rămâne realizabilă

Axiomele calculului predicatelor

A1: $\alpha \rightarrow (\beta \rightarrow \alpha)$ (A1-A3 din logica propozițională)
A2: $(\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))$
A3: $(\neg\beta \rightarrow \neg\alpha) \rightarrow (\alpha \rightarrow \beta)$
A4: $\forall x\,(\alpha \rightarrow \beta) \rightarrow (\forall x\alpha \rightarrow \forall x\beta)$
A5: $\forall x\alpha \rightarrow \alpha[x \leftarrow t]$ dacă x poate fi substituit* cu t în α
A6: $\alpha \rightarrow \forall x\alpha$ dacă x nu apare liber în α

*Definim: putem *substitui* variabila x cu termenul t în $\forall y\phi$ dacă:
x nu apare liber în ϕ (substituția nu are efect) sau
x se poate substitui cu t în ϕ și y nu apare în t
(nu putem substitui variabile legate)

În logica cu egalitate, adăugăm si
A7: $x = x$
A8: $x = y \rightarrow \alpha = \beta$
unde β se obține din α înlocuind oricâte din aparițiile lui x cu y .

Regula de inferență: e suficient *modus ponens*: $\dfrac{A \qquad A \rightarrow B}{B}$

Deducție

Fie H o mulțime de formule. O *deducție* (demonstrație) din H e un șir de formule $A_1, A_2, \cdots, A_n$, astfel ca $\forall i \in \overline{1, n}$

1. A_i este o *axiomă*, sau
2. A_i este o *ipoteză* (o formulă din H), sau
3. A_i rezultă prin *modus ponens* din A_j, A_k anterioare ($j, k < i$)

Spunem că A_n *rezultă* din H (e *deductibil*, e o *consecință*).

Notăm:

$$H \vdash A_n$$

Alte reguli de inferență

$$\frac{\forall x \;\; \phi(x)}{\phi(c)}$$

instanțiere universală (*vezi A*5)

unde c e o constantă arbitrară (nu apare anterior în demonstrație)
Dacă ϕ e valabil pentru orice x , atunci și pentru o valoare arbitrară c.

$$\frac{\phi(c)}{\forall x \;\; \phi(x)}$$

generalizare universală (*vezi A*6)

unde c e o valoare arbitrară (*nu apare în ipoteze*)
Dacă ϕ e valabilă pentru o valoare arbitrară, e valabilă pentru orice x .

$$\frac{\exists x \;\; \phi(x)}{\phi(c)}$$

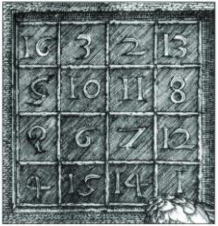
instanțiere existențială

Dacă există o valoare cu proprietatea ϕ, o instanțiem (cu un nume nou).

$$\frac{\phi(c)}{\exists x \;\; \phi(x)}$$

generalizare existențială

Dacă ϕ e adevărată pentru o valoare, există o valoare care o face adevărată



Rezoluția în Logica predicatelor
Unificarea
Demonstrații în Logica Predicatelor
Semantica în Logica Predicatelor
Exerciții

Semantica

Definim noțiunile:

univers

interpretare

model

consecință semantică

Cum interpretăm o formulă ?

Intuitiv, găsim un *înțeles* pentru fiecare simbol din formulă:

O *interpretare* (*structură*) I în logica predicatelor constă din:

- o mulțime nevidă U numită *universul* sau *domeniul* lui I (mulțimea valorilor pe care le pot lua variabilele)
- pentru orice simbol de constantă c, o valoare $c_I \in U$
- pentru orice simbol de funcție n-ară f, o funcție $f_I : U^n \to U$
- pentru orice simbol de predicat n-ar P, o submulțime $P_I \subseteq U^n$. (o *relație* n-ară pe U)

Deci, dăm o *interpretare* fiecărui simbol din formulă.

O interpretare *nu* dă valori variabilelor (vezi ulterior: atribuire).

Exemple de interpretări

$\forall x\ \forall y\ \forall z. P(x, y) \wedge P(y, z) \rightarrow P(x, z)$ tranzitivitate

De exemplu:

universul U = numere reale;

predicatul P: relația $\leq$

$\exists e \forall x\ \neg A(x, e)$

existența mulțimii vide:

predicatul $A(x, y)$ e $x \in y$

Implicația logică (consecința semantică)

Fie H o mulțime de formule și C o formulă.
Notăm $I \vDash H$ dacă I e un model pentru fiecare formulă din H.

Spunem că H *implică* C ($H \vDash C$) dacă pentru orice interpretare I,

$$I \vDash H \text{ implică } I \vDash C$$

(C e adev. în orice interpretare care satisface toate ipotezele din H)

Consistență și completitudine

La fel ca în logica propozițională
- *deducția* (demonstrația) se face pur sintactic
- *consecința/ implicația logică* e o noțiune semantică,
  considerând *interpretări* și valori de adevăr.

Calculul predicatelor de ordinul I este *consistent* și *complet*:

$$H \vdash C \text{ dacă și numai dacă } H \models C$$

Concluzia C se poate *deduce* (demonstra) $\vdash$ din ipotezele H dacă și numai dacă ea e o *consecință semantică* $\models$ a ipotezelor H (e adevărată în orice *interpretare* care satisface toate ipotezele)

Dar: relația de implicație logică e doar *semidecidabilă*
- dacă o formulă e o tautologie, ea poate fi demonstrată
- dar dacă nu e, încercarea de a o demonstra (sau o refuta) poate continua la nesfârșit

Există logici mai bogate decât logica predicatelor

Principiul *inducției* matematice e (în ciuda numelui)
o *regulă de deducție* în *teoria aritmetică* a numerelor naturale

$$\forall P[P(0) \land \forall k \in \mathbb{N}.P(k) \to P(k+1)] \to \forall n \in \mathbb{N}\ P(n)$$

formulă în logica de *ordinul 2* (cuantificare peste predicate)

Logica are limitări

Teoria numerelor naturale cu adunare (aritmetica Presburger) e *decidabilă* (orice putem *exprima* despre adunarea numerelor naturale *e demonstrabil*).

Dar: nu putem exprima divizibilitate, numere prime, etc.

Aritmetica lui Peano (cu adunare și înmulțire) e mai bogată dar e *nedecidabilă*: sunt afirmații despre care *nu se poate decide* dacă sunt adevărate sau nu.

Teoremele de incompletitudine ale lui Gödel

Prima teoremă de incompletitudine:

Orice sistem logic *consistent* care poate exprima aritmetica elementară e *incomplet*

i.e., se pot scrie afirmații care nu pot fi nici demonstrate nici infirmate în acel sistem

Demonstrație: codificând formule și demonstrații ca numere construim un număr care exprimă că formula sa e nedemonstrabilă

A doua teoremă de incompletitudine:

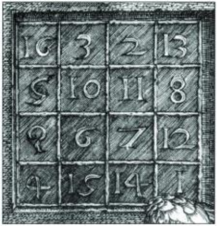
Consistența unui sistem logic capabil să exprime aritmetica elementară *nu poate fi demonstrată* în cadrul acelui sistem.

dar ar putea fi eventual demonstrată în alt sistem logic

Exerciţiu

Formalizaţi în logica predicatelor:

1. Toti cainii latra noaptea.
2. Oricine are o pisica nu o sa aibe nici un soarece.
3. Cei care adorm greu nu au nimic care latra noaptea.
4. Ionut are fie o pisica, fie un caine.
5. (Concluzia) Daca Ionut adoarme greu, atunci Ionut nu are nici un soarece.



Vă mulțumesc!

Bibliografie

Conținutul cursului se bazează pe materialele de anii trecuți de la cursul de LSD, predat de conf. dr. ing.Marius Minea și ș.l. dr. ing. Casandra Holotescu (http://staff.cs.upt.ro/~marius/curs/lsd/index.html)

Intrebarile de logica de la inceputul cursului au fost preluate din cursul Introduction to Logic de la Stanford University (https://www.coursera.org/learn/logic-introduction)