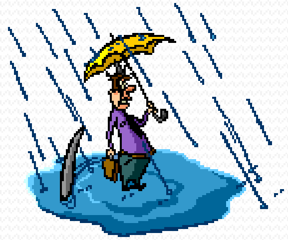
Logică computațională
Curs 8

Lector dr. Pop Andreea-Diana

Logica predicatelor de ordinul I



- ”Afară plouă”
- ”În fiecare zi plouă afară”
- ”Mâine va ploua”
- ”Acum o lună a plouat”

- p $\equiv P(a)$ (a – azi)
- $(\forall x)\ P(x)$ (x –ziua)
- $P(b)$ (b – mâine)
- $P(f(a))$ ($f(x) := x - 30$)

Sistemul axiomatic al logicii predicatelor de ordinul I - Alfabetul

- P=(Σ_{Pr} , F_{Pr} , A_{Pr} , R_{Pr})
  - $\Sigma_{\text{Pr}} = Var \cup Const \cup (\cup^{n}_{j=1} \mathscr{F}_j) \cup (\cup^{n}_{j=1} \mathscr{P}_j) \cup \mathscr{P}_0 \cup Conective \cup Cuantif$
    - $Var = \{ x, y, z, \ldots\}$ – mulţimea *simbolurilor de variabile*
    - $Const = \{ a, b, c, \ldots\}$ – mulţimea *constantelor*
    - $\mathscr{F}_j = \{ f \mid f : D^{j} \to D\}$ – mulţimea *simbolurilor de funcţii* de aritate "*j*"
    - $\mathscr{P}_j = \{ P \mid P: D^{j} \to \{T, F\}\}$ – mulţimea *simbolurilor de predicate* de aritate "*j*"
    - $\mathscr{P}_0 = \{ p, q, r, \ldots\} \cup \{T, F\}$ – mulţimea *variabilelor propoziționale* și a *valorilor de adevăr*
    - $Conective = \{ \neg , \wedge , \vee , \to , \leftrightarrow \}$
    - *Cuantif* ={ $\forall$ (cuantificatorul universal), $\exists$ (cuantificatorul existențial)}

TERM, ATOM, Literal

- ***TERM*** = mulţimea *termenilor*:
  - $Var \subset TERM$
  - $Const \subset TERM$
  - dacă $f \in \mathcal{F}_k$ și $t_1, \ldots, t_k \in TERM$ atunci $f(t_1, \ldots, t_k) \in TERM$
- ***ATOM*** = mulţimea *formulelor atomice* (*atomilor*):
  - $T, F \in ATOM$
  - dacă $P \in \mathcal{P}_k$ și $t_1, \ldots, t_k \in TERM$ atunci $P(t_1, \ldots, t_k) \in ATOM$
- ***Literal*** = un atom sau negația sa

Formule corect construite

- F_{Pr} = mulţimea formulelor predicative bine formate
  - $ATOM \subset F_{\mathrm{Pr}}$
  - dacă $U \in \mathrm{F}_{\mathrm{Pr}}$ și $x \in Var$ astfel încât x nu se află deja sub incidența unui cuantificator (nu este legat), atunci:

    $(\forall x)\ U\ (x) \in \mathrm{F}_{\mathrm{Pr}}$ și $(\exists x)\ U\ (x) \in \mathrm{F}_{\mathrm{Pr}}$
  - dacă U, $V \in \mathrm{F}_{\mathrm{Pr}}$ astfel încât U și V nu conțin aceeași variabilă atât liberă cât și legată, atunci:

    $\neg\ U \in F_{\mathrm{Pr}}$, $U \wedge V \in F_{\mathrm{Pr}}$, $U \vee V \in F_{\mathrm{Pr}}$, $U \rightarrow V \in F_{\mathrm{Pr}}$, $U \leftrightarrow V \in F_{\mathrm{Pr}}$

Axiome

- $A_{\mathrm{Pr}} = \{A_1, A_2, A_3, A_4, A_5\}$ scheme axiomatice
  - A_1: $U \to (V \to U)$
  - A_2: $(U \to (V \to Z)) \to ((U \to V) \to (U \to Z))$
  - A_3: $(U \to V) \to (\neg V \to \neg U)$
  - A_4: $(\forall x)\, U(x) \to U(t)$, unde t este un termen arbitrar
  - A_5: $(U \to V(y)) \to (U \to (\forall x)\, V(x))$, unde y este o variabilă liberă în V care nu apare în U, iar x nu este variabilă liberă nici în U, nici în V

Reguli de inferență

- $R_{\mathrm{Pr}} = \{mp, gen\}$
  - *modus ponens*: $U\ ,\ U \to V \vdash_{mp} V$
  - *regula generalizării*: $U\,(x) \vdash_{gen} (\forall x)\ U\,(x)$

(*x* era o variabilă liberă în *U*)

Definiții

- Variabilele din formulele predicative care se află sub incidenţa unui cuantificator se numesc *variabile legate*, în caz contrar ele se numesc *variabile libere*.

- O *formulă* predicativă se numeşte *închisă*, dacă toate variabilele sale sunt legate, iar în caz contrar se numeşte *deschisă*.

Exemplu – Transformarea unor afirmaţii din limbaj natural în logica predicatelor

- Dacă x şi y sunt întregi nenegativi şi x este mai mare decât y, atunci x^2 este mai mare decât y^2.

D = **N**

Variabilele (din D): x, y

Constantele (din D): 2

Simboluri de Funcții (definite pe $D^n \to D$):

$f : \mathbf{N}^2 \to \mathbf{N}$, $f(x,y) = x^y$

Simboluri de Predicate (definite pe $D^n \to \{T,F\}$):

$P : \mathbf{N}^2 \to \{T,F\}$, $P(x,y) =$ "$x > y$"

Formula Predicativă: $(\forall x)\ (\forall y)\ (P(x, y) \to P(f(x,2), f(y,2)))$

Definiția deducției

- Fie formulele $U_1, U_2, \ldots, U_n$ numite ipoteze şi V formulă propoziţională. Spunem că **V este deductibilă din $U_1, U_2, \ldots, U_n$** şi notăm $U_1, U_2, \ldots, U_{n-1}, U_n \vdash V$, dacă există o secvenţă de formule $(f_1, f_2, \ldots, f_m)$ astfel încât $f_m = V$ şi $\forall i \in \{1, \ldots, m\}$ avem:
  - $f_i \in A_{\mathrm{Pr}}$;
  - $f_i \in \{U_1, U_2, \ldots, U_n\}$;
  - $f_j, f_k \vdash_{mp} f_i$, $j < i$ și $k < i$
  - $f_j \vdash_{gen} f_i$, $j < i$
- Secvenţa $(f_1, f_2, \ldots, f_m)$ se numeşte *deducţia* lui V din $U_1, U_2, \ldots, U_n$.

Definiția teoremei

- O formulă $U \in F_{\text{Pr}}$, astfel încât $\varnothing \vdash U$ (sau $\vdash U$) se numeşte ***teoremă***.

Exerciţiu

$A_1: U \to (V \to U)$

$A_2: (U \to (V \to Z)) \to ((U \to V) \to (U \to Z))$

$A_3: (U \to V) \to (\neg V \to \neg U)$

$A_4: (\forall x)\, U(x) \to U(t),\ldots$

$A_5: (U \to V(y)) \to (U \to (\forall x)\, V(x)),\ldots$

$U,\ U \to V \vdash_{mp} V$

$U(x) \vdash_{gen} (\forall x)\, U(x)$

- $(\forall y)\, P(y) \vdash (\forall x)\, (Q(x) \to P(x))$

$f_1: (\forall y)\, P(y)$ (ip)

$f_2: (\forall y)\, P(y) \to P(x)$ (A_4)

$f_1,\ f_2 \vdash_{mp} f_3: P(x)$

$f_4: P(x) \to (Q(x) \to P(x))$ (A_1)

$f_3,\ f_4 \vdash_{mp} f_5: Q(x) \to P(x)$

$f_5 \vdash_{gen} f_6: (\forall x)\, (Q(x) \to P(x))$

$(f_1, f_2, \ldots, f_6)$ este deducția lui $(\forall x)\, (Q(x) \to P(x))$ din $(\forall y)\, P(y)$, deci $(\forall y)\, P(y) \vdash (\forall x)\, (Q(x) \to P(x))$

Semantica logicii predicatelor de ordinul I

- realizează legătura dintre
  - constantele,
  - simbolurile de funcţii,
  - simbolurile de predicate respectiv

  constantele, funcţiile şi predicatele din conceptualizarea universului modelat

- este furnizat un înţeles în termenii universului modelat pentru orice formulă din limbaj

Definiția interpretării

- O *interpretare* pentru un limbaj L al calculului predicatelor este o pereche $I=<D, m>$, unde :
  - D este o mulţime nevidă numită *domeniu al interpretării*.
  - m este o funcţie care asociază:
    - o valoare fixă $m(c)$ din domeniul D unei constante c.
    - o funcţie $m(f): D^n \to D$ fiecărui simbol de funcţie f de aritate n;
    - un predicat $m(P): D^n \to \{T, F\}$ fiecărui simbol de predicat P de aritate n.

Notații

pentru interpretarea I=<D, m>:

- $|I| = D$ este domeniul interpretării I
- $I|x| = m(x)$ unde x este constantă, simbol de funcţie sau simbol de predicat.
- As(I) mulţimea funcţiilor de asignare de variabile peste domeniul interpretării I.

  O funcţie $a \in$ As(I) este definită astfel a:$Var \rightarrow |I|$.
- $[a]_x=\{a' \mid a' \in$ As(I) şi $a'(y)=a(x)$, pentru orice $y \neq x\}$.

Definiția funcției de evaluare

Fie o interpretare I şi $a \in \mathrm{As}(I)$. Se defineşte inductiv ***funcţia de evaluare*** ν^I_a:

- $\nu^I_a(x) = a(x)$, $x \in Var$;
- $\nu^I_a(c) = \mathrm{I}|c|$, $c \in Const$;
- $\nu^I_a(f(t_1, \ldots, t_n)) = I|f|(\nu^I_a(t_1), \ldots, \nu^I_a(t_n))$, $f \in \mathcal{F}_k$, $n>0$;
- $\nu^I_a(P(t_1, \ldots, t_n)) = I|P|(\nu^I_a(t_1), \ldots, \nu^I_a(t_n))$, $P \in \mathcal{P}_k$, $n>0$;
- $\nu^I_a(\neg A) = \neg\, \nu^I_a(A)$; $\nu^I_a(A \wedge B) = \nu^I_a(A) \wedge \nu^I_a(B)$
- $\nu^I_a(A \vee B) = \nu^I_a(A) \vee \nu^I_a(B)$; $\nu^I_a(A \rightarrow B) = \nu^I_a(A) \rightarrow \nu^I_a(B)$
- $\nu^I_a((\exists x)A(x)) = T$ dacă şi numai dacă $\nu^I_{a'}(A(x)) = T$ pentru o funcţie $a' \in [a]_x$
- $\nu^I_a((\forall x)A(x)) = T$ dacă şi numai dacă $\nu^I_{a'}(A(x)) = T$ pentru orice funcţie $a' \in [a]_x$

Concepte semantice

- *O formulă A* este *realizabilă (consistentă)* dacă şi numai dacă există o interpretare I şi o funcţie $a \in \mathrm{As}(I)$ astfel încât $\nu^I{}_a(A)=T$. În caz contrar formula se numeşte *nerealizabilă (inconsistentă)*.
- Formula A este *adevărată în interpretarea I* dacă şi numai dacă pentru orice funcţie $a \in \mathrm{As}(I)$ de asignare avem $\nu^I{}_a(A)=\mathrm{T}$ şi notăm $\models_I A$, iar I se numeşte *model* al lui A.
- Interpretarea I se numește *anti-model* al formulei predicative A dacă A este evaluată ca falsă în I, adică: $\forall\ a \in \mathrm{As}(I)$ are loc $\nu^I{}_a(A)=F$.
- *Formula* A este *validă (tautologie)* dacă şi numai dacă A este adevărată în orice interpretare şi se notează: $\models A$.
- Două *formule A* şi *B* sunt *logic echivalente* dacă $\nu^I{}_a(A)= \nu^I{}_a(B)$ pentru orice interpretare I şi funcţie a de asignare. Notaţie $A \equiv B$.
- O mulţime S de formule *implică logic* o formulă A dacă toate modelele mulţimii (adică modelele conjuncţiei formulelor din S) sunt modele ale formulei. Spunem că A este o *consecinţă logică* a mulţimii de formule S şi notăm $S \models A$.
- O *mulţime de formule* predicative este *consistentă* dacă formula obținută prin conjuncţia elementelor sale este consistentă, adică are cel puţin un model.
- O *mulţime de formule* este *inconsistentă* dacă nu există nici un model pentru formula obținută prin conjuncţia elementelor sale.

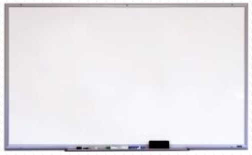
Observații

- Evaluarea unei formule *A* închise depinde doar de interpretarea în care se evaluează formula, notându-se ν^{I} (A).
- Dacă interpretarea are domeniu finit:
  - o formulă cuantificată *universal* este înlocuită cu *conjuncția* instanțelor acesteia folosind toate elementele domeniului de interpretare
  - o formulă cuantificată *existențial* este înlocuită cu *disjuncția* instanțelor acesteia folosind toate elementele domeniului de interpretare

Exerciţiu:

Evaluaţi într-o interpretare cu domeniu finit şi una cu domeniu infinit:

- $U = (\forall x)(P(x) \vee Q(x)) \rightarrow (\forall x)P(x) \vee (\forall x)Q(x)$



Exercițiu (domeniu infinit):

$U = (\forall x)(P(x) \vee Q(x)) \rightarrow (\forall x)P(x) \vee (\forall x)Q(x)$

$I = <D,m>$

$D = \mathbf{R}$

Sunt constante (a,b,c)? Nu

Sunt simboluri de funcții? Nu

Sunt simboluri de Predicate? Da: P, Q – ambele au aritatea 1

m(P):$\mathbf{R} \rightarrow \{T,F\}$, m(P)(x)="x<0"

m(Q):$\mathbf{R} \rightarrow \{T,F\}$, m(Q)(x)="x>0"

$v^I(U) = v^I((\forall x)(P(x) \vee Q(x)) \rightarrow (\forall x)P(x) \vee (\forall x)Q(x)) =$

$= v^I((\forall x)(P(x) \vee Q(x))) \rightarrow v^I((\forall x)P(x) \vee (\forall x)Q(x)) =$

$= v^I((\forall x)(P(x) \vee Q(x))) \rightarrow v^I((\forall x)P(x)) \vee v^I((\forall x)Q(x)) =$

= "$\forall x \in \mathbf{R}$, x<0 sau x>0" $\rightarrow$ "$\forall x \in \mathbf{R}$, x<0" $\vee$ "$\forall x \in \mathbf{R}$, x>0" =

= F $\rightarrow$ F $\vee$ F = F $\rightarrow$ F = T, deci I este un model

Exercițiu (domeniu finit):

$U = (\forall x)(P(x) \vee Q(x)) \rightarrow (\forall x)P(x) \vee (\forall x)Q(x)$

$I = <D,m>$

D = {România, Norvegia}

Sunt constante (a,b,c)? Nu

Sunt simboluri de funcții? Nu

Sunt simboluri de Predicate? Da: P, Q – ambele au aritatea 1

m(P):{România, Norvegia} $\rightarrow$ {T,F}, m(P)(x)="x se califică la Mondială"

m(Q):{România, Norvegia} $\rightarrow$ {T,F}, m(Q)(România)=T, m(Q)(Norvegia)=F

$v^I(U) = v^I((\forall x)(P(x) \vee Q(x)) \rightarrow (\forall x)P(x) \vee (\forall x)Q(x)) =$

$= v^I((\forall x)(P(x) \vee Q(x))) \rightarrow v^I((\forall x)P(x) \vee (\forall x)Q(x))=$

$= v^I((\forall x)(P(x) \vee Q(x))) \rightarrow v^I((\forall x)P(x)) \vee v^I((\forall x)Q(x))=$

= (m(P)(România) $\vee$ m(Q)(România)) $\wedge$ (m(P)(Norvegia) $\vee$ m(Q)(Norvegia)) $\rightarrow$ m(P)(România) $\wedge$ m(P)(Norvegia) $\vee$ m(Q)(România) $\wedge$ m(Q)(Norvegia) = (T $\vee$T) $\wedge$ (T $\vee$ F) $\rightarrow$ (T $\wedge$T) $\vee$ (T $\wedge$ F) = T $\wedge$ **T** $\rightarrow$T $\vee$ F=T $\rightarrow$T=T

Deci și I este model

Echivalenţe logice în calculul predicatelor

- Legile de expansiune

$(\forall x)\, A(x) \equiv (\forall x)\, A(x) \wedge A(t)$, t – termen oarecare $t \neq x$

$(\exists x)\, A(x) \equiv (\exists x)\, A(x) \vee A(t)$, t – termen oarecare $t \neq x$

- Legile infinite ale lui DeMorgan

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

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

- Legile de interschimbare a cuantificatorilor

$(\exists x)\, (\exists y)\, A(x, y) \equiv (\exists y)\, (\exists x)\, A(x , y)$

$(\forall x)\, (\forall y)\, A(x, y) \equiv (\forall y)\, (\forall x)\, A(x , y)$

Legi de extragere

- Legile de extragere a cuantificatorilor în faţa formulei

$A \vee (\exists x)\, B(x) \equiv (\exists x)\, (A \vee B(x))$

$A \vee (\forall x)\, B(x) \equiv (\forall x)\, (A \vee B(x))$

$A \wedge (\exists x)\, B(x) \equiv (\exists x)\, (A \wedge B(x))$

$A \wedge (\forall x)\, B(x) \equiv (\forall x)\, (A \wedge B(x))$

unde formula A nu conţine pe x ca variabilă liberă sau legată

$(\exists x)\, A(x) \vee B \equiv (\exists x)\, (A(x) \vee B)$

$(\forall x)\, A(x) \vee B \equiv (\forall x)\, (A(x) \vee B)$

$(\exists x)\, A(x) \wedge B \equiv (\exists x)\, (A(x) \wedge B)$

$(\forall x)\, A(x) \wedge B \equiv (\forall x)\, (A(x) \wedge B)$

unde formula B nu conţine pe x ca variabilă liberă sau legată

Legile distributivităţii

- $\exists$ faţă de $\vee$:

  $(\exists x)\ (A(x) \vee B(x)) \equiv (\exists x)\ A(x) \vee (\exists x)\ B(x)$

- $\forall$ faţă de $\wedge$:

  $(\forall x)\ (A(x) \wedge B(x)) \equiv (\forall x)\ A(x) \wedge (\forall x)\ B(x)$

Semidistributivităţi

- $\exists$ faţă de $\wedge$:

  $\models (\exists x)\ (A(x) \wedge B(x)) \rightarrow (\exists x)\ A(x) \wedge (\exists x)\ B(x)$

- $\forall$ faţă de $\vee$:

  $\models (\forall x)\ A(x) \vee (\forall x)\ B(x) \rightarrow (\forall x)\ (A(x) \vee B(x))$

Forme normale ale formulelor predicative

- O formulă U predicativă este în ***forma normală prenexă*** dacă ea este de forma $(Q_1x_1)\ldots(Q_nx_n)\ M$, unde $Q_i, i=1,\ldots,n$ sunt cuantificatori logici, iar M nu conţine cuantificatori. Secvenţa se numeşte *prefixul formulei U*, iar M este *matricea formulei U*.
- O formulă U predicativă este în *forma normală prenexă conjunctivă* dacă ea este în formă normală prenexă, iar matricea este în FNC.

- **Teoremă:**

  Orice formulă din calculul predicatelor poate fi transformată într-o forma normală prenexă logic echivalentă cu ea.

Algoritmul de aducere la forma normală prenexă

Pas 1: Se înlocuiesc conectivele $\rightarrow$ şi $\leftrightarrow$ folosind $\neg$, $\wedge$, $\vee$.

Pas 2: Se aplică legile finite şi infinite ale lui DeMorgan astfel încât cuantificatorii să nu fie precedaţi de negaţie.

Pas 3: Se redenumesc variabilele legate astfel încât ele să fie distincte.

Pas 4: Se utilizează echivalenţele logice care reprezintă *legile de extragere a cuantificatorilor în faţa formulei.*

!!! Ordinea de extragere a cuantificatorilor este arbitrară.

Forma normală Skolem (**Pas 5**)

- Fie U o formulă predicativă, iar $U^P = (Q_1x_1)\ldots(Q_nx_n)\ M$ una dintre formele sale normale prenexe.
- Formulei U îi corespunde o formulă în *forma normală Skolem* notată U^S care se obţine astfel: pentru fiecare cuantificator existenţial Q_r din prefix se aplică următoarea transformare:
  - dacă înaintea simbolului Q_r nu apare niciun cuantificator universal, atunci se alege o constantă notată **a,** diferită de toate constantele care apar în M şi se înlocuiesc toate apariţiile variabilei x_r în M cu **a**. Se şterge (Q_rx_r) din prefixul formulei.
  - dacă înaintea simbolului Q_r apar cuantificatorii universali $Q_{s_1},\ldots,Q_{s_m}$, unde $1 \leq s_1 < \ldots < s_m < r$, atunci alegem un simbol f de funcţie de m variabile, diferit de celelalte simboluri de funcţii şi se înlocuieşte fiecare apariţie a variabilei în M cu $f\,(x_{s_1},\ldots, x_{s_m})$. Se şterge (Q_rx_r) din prefixul formulei.
- Constantele şi funcţiile folosite pentru a înlocui variabilele existenţiale se numesc *constante Skolem* şi *funcţii Skolem*.

Exemplu

$(\exists x)\ (\exists y)\ (\forall z)\ (\exists t)\ (\forall s)\ (\exists u)\ (\forall w)\ \mathrm{P}(x, y, z, t, s, u, w)$

$x \leftarrow a$

$y \leftarrow b$

$t \leftarrow f\ (z)$

$u \leftarrow g\ (z,s)$

Forma Skolem: $(\forall z)\ (\forall s)\ (\forall w)\ \mathrm{P}(a, b, z, f\ (z), s, g(z,s), w)$

Forma normală clauzală

- Formulei U îi corespunde o formulă în *forma normală Skolem fără cuantificatori* notată U^{Sq} care se obţine prin eliminarea cuantificatorilor universali din U^{S}. (**Pas 6**)
- Formulei U îi corespunde o formulă în *forma normală clauzală* notată U^{C} care se obţine din U^{Sq} prin aducerea la FNC. (**Pas 7**)

- Obs.: Transformările utilizate în procesul de Skolemizare nu păstrează echivalenţa logică, dar păstrează inconsistenţa

Teoremă

Fie $U_1, U_2, .., U_n, V$ formule predicative.

- V inconsistentă **ddacă** V^P inconsistentă **ddacă** V^S inconsistentă **ddacă** V^{Sq} inconsistentă **ddacă** V^C inconsistentă.
- $\{U_1, U_2, .., U_n\}$ inconsistentă **ddacă** $\{U_1^C, U_2^C, .., U_n^C\}$ inconsistentă.

Exercițiu: Aduceți la Forma normală Clauzală

- $U = (\forall x)(P(x) \vee Q(x)) \rightarrow (\forall x)P(x) \vee (\forall x)Q(x)$

Pas 1: Se înlocuiesc conectivele $\rightarrow$ şi $\leftrightarrow$ folosind $\neg$, $\wedge$, $\vee$.

$U \equiv \neg\, (\forall x)(P(x) \vee Q(x)) \vee (\forall x)P(x) \vee (\forall x)Q(x)$

Pas 2: Se aplică legile finite şi infinite ale lui DeMorgan astfel încât cuantificatorii să nu fie precedaţi de negaţie.

$U \equiv (\exists x)\, (\neg P(x) \wedge \neg\, Q(x)) \vee (\forall x)P(x) \vee (\forall x)Q(x)$

Pas 3: Se redenumesc variabilele legate astfel încât ele să fie distincte

$U \equiv (\exists x)\, (\neg P(x) \wedge \neg\, Q(x)) \vee (\forall y)P(y) \vee (\forall z)Q(z)$

Pas 4: Se utilizează echivalenţele logice care reprezintă *legile de extragere a cuantificatorilor în faţa formulei*. (Forma Normală Prenexă)

Exerciţiu: Aduceţi la Forma normală Clauzală

- $U = (\forall x)(P(x) \vee Q(x)) \rightarrow (\forall x)P(x) \vee (\forall x)Q(x)$

$U \equiv (\exists x)\,(\neg P(x) \wedge \neg\, Q(x)) \vee (\forall y)P(y) \vee (\forall z)Q(z)$

Pas 4: Se utilizează echivalenţele logice care reprezintă *legile de extragere a cuantificatorilor în faţa formulei*. (Forma Normală Prenexă)

$U \equiv U^{P_1} = (\exists x)\,(\forall y)\,(\forall z)\,((\neg P(x) \wedge \neg\, Q(x)) \vee P(y) \vee Q(z))$

$U \equiv U^{P_2} = (\exists x)\,(\forall z)\,(\forall y)\,((\neg P(x) \wedge \neg\, Q(x)) \vee P(y) \vee Q(z))$

$U \equiv U^{P_3} = (\forall y)\,(\exists x)(\forall z)\,((\neg P(x) \wedge \neg\, Q(x)) \vee P(y) \vee Q(z))$

$U \equiv U^{P_4} = (\forall z)\,(\exists x)\,(\forall y)\,((\neg P(x) \wedge \neg\, Q(x)) \vee P(y) \vee Q(z))$

$U \equiv U^{P_5} = (\forall y)\,(\forall z)\,(\exists x)\,((\neg P(x) \wedge \neg\, Q(x)) \vee P(y) \vee Q(z))$

$U \equiv U^{P_6} = (\forall z)\,(\forall y)\,(\exists x)\,((\neg P(x) \wedge \neg\, Q(x)) \vee P(y) \vee Q(z))$

Exercițiu: Aduceți la Forma normală Clauzală

- $U = (\forall x)(P(x) \vee Q(x)) \rightarrow (\forall x)P(x) \vee (\forall x)Q(x)$

$U \equiv U^{P_3} \equiv (\forall y)\,(\exists x)(\forall z)\,((\neg P(x) \wedge \neg\, Q(x)) \vee P(y) \vee Q(z))$

Pas 5: Eliminarea cuantificatorilor $\exists$ **(**Forma normală Skolem)

(se păstrează doar inconsistența)

$x \leftarrow f(y)$

$U \not\equiv U^{S_3} \equiv (\forall y)\,(\forall z)\,((\neg P(f(y)) \wedge \neg\, Q(f(y))) \vee P(y) \vee Q(z))$

Pas 6: Eliminarea cuantificatorilor $\forall$ **(**Forma normală Skolem fără cuantificatori)

$U \not\equiv U^{Sq_3} \equiv (\neg P(f(y)) \wedge \neg\, Q(f(y))) \vee P(y) \vee Q(z)$

Pas 7: Aducerea la Forma Normală Clauzală **(**distributivitatea $\vee$ față de $\wedge$)

$U \not\equiv U^{C_3} \equiv (\neg P(f(y)) \vee P(y) \vee Q(z)) \wedge (\neg\, Q(f(y)) \vee P(y) \vee Q(z))$

Sumar

Pas 1: Se înlocuiesc conectivele $\rightarrow$ şi $\leftrightarrow$ folosind $\neg$, $\wedge$, $\vee$.

Pas 2: Se aplică legile finite şi infinite ale lui DeMorgan astfel încât cuantificatorii să nu fie precedaţi de negaţie.

Pas 3: Se redenumesc variabilele legate astfel încât ele să fie distincte.

Pas 4: Se utilizează echivalenţele logice care reprezintă *legile de extragere a cuantificatorilor în faţa formulei*.

Pas 5: Eliminarea cuantificatorilor $\exists$ **(**Forma normală Skolem)

Pas 6: Eliminarea cuantificatorilor $\forall$ **(**Forma normală Skolem fără cuantificatori)

Pas 7: Aducerea la Forma Normală Clauzală **(**distributivitatea $\vee$ față de $\wedge$)

Proprietăţile logicii predicatelor de ordinul I

- **Teorema de completitudine și corectitudine**

  Fie S o mulţime de formule predicative, iar A o formulă predicativă.

  - **completitudinea**: dacă $S \models A$ atunci $S \vdash A$.
  - **corectitudinea**: dacă $S \vdash A$ atunci $S \models A$.

- **Teorema respingerii**

  Fie S o mulţime de formule predicative, iar A o formulă predicativă. Dacă $S \cup \{\neg A\}$ este inconsistentă atunci $S \vdash A$.

- **Teorema de deducţie**

  Fie S o mulţime de formule predicative, iar A o formulă predicativă. Dacă $S \cup \{A\} \vdash B$ atunci $S \vdash A \rightarrow B$.

Teorema lui Church 1936

- Problema validităţii unei formule în calculul predicatelor de ordinul I este *nedecidabilă*. Mulţimea formulelor valide din acest sistem logic este recursiv numărabilă, adică există o procedură *P* care, având ca intrare o formulă *A* din limbaj, are următorul comportament:
  - dacă formula *A* este validă, *P* se termină şi furnizează răspunsul corespunzător;
  - dacă formula *A* nu este validă, *P* se termină cu răspunsul corespunzător *sau* execuţia procedurii nu se încheie niciodată.

- Calculul predicatelor este **semi-decidabil**.

Logică computațională
Curs 9

Lector dr. Pop Andreea-Diana

Metoda tabelelor semantice în calculul predicatelor

- introdusă de Smullyan
- se bazează pe considerații semantice
- încearcă să construiască modelele unei formule date (FND)
- $\vdash U$ prin respingere, $\neg U$ nu are modele
- ideea:
  - descompunerea formulei inițiale în subformule
  - până la nivel de literali

Clase de formule (1)

- clasa α - formule de tip conjunctiv

$$A \wedge B$$

$$\neg (A \vee B)$$

$$\neg (A \rightarrow B)$$

- clasa β - formule de tip disjunctiv

$$A \vee B$$

$$\neg (A \wedge B)$$

$$A \rightarrow B$$

Clase de formule (2)

- clasa γ - formule cuantificate universal

$$(\forall x)\, A(x)$$

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

- clasa δ - formule cuantificate existențial

$$(\exists x)\, A(x)$$

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

Reguli de descompunere a formulelor (1)

- regula α

$$\begin{array}{c} A \wedge B \\ | \\ A \\ | \\ B \end{array} \qquad \begin{array}{c} \neg(A \vee B) \\ | \\ \neg A \\ | \\ \neg B \end{array} \qquad \begin{array}{c} \neg(A \to B) \\ | \\ A \\ | \\ \neg B \end{array}$$

- regula β

$$\begin{array}{ccc} & A \vee B & \\ \swarrow & & \searrow \\ A & & B \end{array}$$

$$\begin{array}{ccc} & \neg(A \wedge B) & \\ \swarrow & & \searrow \\ \neg A & & \neg B \end{array}$$

$$\begin{array}{ccc} & A \to B & \\ \swarrow & & \searrow \\ \neg A & & B \end{array}$$

Reguli de descompunere a formulelor (2)

- regula γ

$$
\begin{array}{c}
(\forall x)\, A(x) \\
\mid \; c_1, c_2, \ldots, c_n \text{ – toate constantele existente pe ramură} \\
A(c_1) \\
\mid \\
A(c_2) \\
\mid \\
\vdots \\
\mid \\
A(c_n) \\
\mid \\
(\forall x)\, A(x) \\
\text{copie formulă}
\end{array}
\qquad
\begin{array}{c}
\neg (\exists x)\, A(x) \\
\mid \\
\neg A(c_1) \\
\mid \\
\neg A(c_2) \\
\mid \\
\vdots \\
\mid \\
\neg A(c_n) \\
\mid \\
\neg (\exists x)\, A(x) \text{ – copie formulă}
\end{array}
$$

- regula δ

$$
\begin{array}{c}
(\exists x)\, A(x) \\
\mid \; a \text{ – constantă nou introdusă} \\
A(a)
\end{array}
\qquad
\begin{array}{c}
\neg (\forall x)\, A(x) \\
\mid \\
\neg A(a)
\end{array}
$$

Arborele binar de descompunere a unei formule

Având o formulă *U*, ei i se poate asocia o tabelă semantică, care este de fapt un arbore binar ce conţine în nodurile sale formule şi se construieşte astfel:

- rădăcina arborelui este etichetată cu formula *U*;
- fiecare ramură a arborelui care conţine o formulă va fi extinsă cu subarborele corespunzător regulii de descompunere care se aplică formulei;
- extinderea unei ramuri se *încheie* în două situaţii:
  a) dacă pe ramură apare o formulă şi negaţia sa;
  b) dacă au fost descompuse toate formulele de pe acea ramură sau prin aplicarea regulilor de descompunere nu se mai obțin formule noi pe acea ramură

Tipuri de ramuri

- O *ramură* a tabelei se numeşte *închisă* (simbolizată prin ⊗) dacă ea conţine o formulă şi negaţia ei, în caz contrar, dacă este completă, *ramura* se numeşte *deschisă* (simbolizată prin⊙).
- O *ramură* a tabelei se numeşte *completă* dacă ea este fie *închisă*, fie *toate formulele* de pe acea ramură au fost *descompuse*.

Tipuri de tabele semantice

- O *tabelă* se numeşte *închisă* dacă toate ramurile sale sunt închise. Dacă o tabelă are cel puţin o ramură deschisă, atunci ea se numeşte *deschisă*.
- O *tabelă* se numeşte *completă* dacă toate ramurile ei sunt complete.

Observaţii:

- Procesul de construire a unei tabele semantice este unul *nedeterminist* deoarece regulile de descompunere se pot aplica în orice ordine şi la un moment dat se pot alege mai multe ramuri pentru extindere. Astfel unei formule i se pot asocia mai multe tabele semantice, dar acestea sunt echivalente.
- Pentru a obţine tabele semantice *cât mai simple* (mai puţin ramificate) se recomandă:
  - utilizarea regulilor de tip α înaintea regulilor de tip β care realizează o ramificare;
  - utilizarea regulilor de tip δ (care introduc constante noi) înaintea regulilor de tip γ care utilizează toate constantele de pe ramura respectivă;

Observaţii (2):

- formulele de pe aceeaşi ramură a unei tabele semantice sunt *legate* între ele prin conectiva logică $\wedge$, iar *ramificarea* corespunde conectivei logice $\vee$.
- tabela semantică asociată unei formule propoziţionale este o reprezentare grafică a *formei* sale *normale disjunctive*. Fiecare ramură reprezintă un *cub* (conjuncţia tuturor literalilor de pe acea ramură), iar arborele este *disjuncţia* tuturor *ramurilor* sale.
- Unei formule *consistente* i se asociază o *tabelă completă deschisă*, iar fiecare *ramură deschisă* a tabelei furnizează cel puţin un *model* pentru formula respectivă.
- O *tabelă semantică închisă* asociată unei formule indică faptul că formula este *inconsistentă*, adică nu există nicio interpretare în care formula să fie adevărată

Teorema de corectitudine şi completitudine a metodei tabelelor semantice

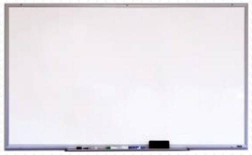
- O formulă U este teoremă (tautologie) dacă şi numai dacă există o tabelă semantică închisă pentru formula $\neg U$.

Teoremă

- $U_1, U_2, \ldots, U_n \vdash Y$ (echivalent cu $U_1, U_2, \ldots, U_n \models Y$) dacă şi numai dacă există o tabelă semantică închisă pentru formula $U_1 \wedge U_2 \wedge \ldots \wedge U_n \wedge \neg Y$.

Exemple

$$\overset{?}{\models} (\exists x)\,(A(x) \land B(x)) \rightarrow (\exists x)\,A(x) \land (\exists x)\,B(x)$$



$\neg((\exists x)\,(A(x) \wedge B(x)) \rightarrow (\exists x)\,A(x) \wedge (\exists x)\,B(x))$ (1) √

$\propto$ (1)

$(\exists x)\,(A(x) \wedge B(x))$ (2) √

$\neg((\exists x)\,A(x) \wedge (\exists x)\,B(x))$ (3)√

δ (2), a – constantă nouă

$A(a) \wedge B(a)$ (4) √

$\propto$ (4)

$\boxed{A(a)}$

$\boxed{B(a)}$

β (3)

$\neg(\exists x)\,A(x)$ (5)√	$\neg(\exists x)\,B(x)$ (6)√
$\boxed{\neg A(a)}$ γ (5), a - constantă existentă	$\boxed{\neg B(a)}$ γ (6), a - constantă existentă
$\neg(\exists x)\,A(x)$ (5’) (copia)	$\neg(\exists x)\,B(x)$ (6’) (copia)
$\otimes$	$\otimes$

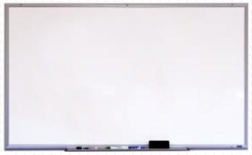
TCC

Deci tabela semntică este închisă $\Rightarrow$ formula este tautologie

Exemple

$$\overset{?}{\models} (\exists x)\,(A(x) \wedge B(x)) \rightarrow (\exists x)\,A(x) \wedge (\exists x)\,B(x)$$

$$\overset{?}{\models} (\forall x)\,(A(x) \vee B(x)) \rightarrow (\forall x)\,A(x) \vee (\forall x)\,B(x)$$



$\neg((\forall x)(A(x) \vee B(x)) \to (\forall x) A(x) \vee (\forall x) B(x))$ (1)√

$\propto$ (1)

$(\forall x)(A(x) \vee B(x))$ (2)√

$\neg((\forall x) A(x) \vee (\forall x) B(x))$ (3)√

$\propto$ (3)

$\neg(\forall x) A(x)$ (4)√

$\neg (\forall x) B(x)$ (5)√

δ (4), a- constantă nouă

$\boxed{\neg A(a)}$

δ (5), b - constantă nouă

$\boxed{\neg B(b)}$

γ (2), a,b- constante existente

$A(a) \vee B(a)$(6)√

$A(b) \vee B(b)$(7)√

$(\forall x)(A(x) \vee B(x))$ (2')

β (6)

$\boxed{A(a)}$ $B(a)$

β (7)

$A(b)$ $\boxed{B(b)}$

⊗ TCC ⊙ ⊗

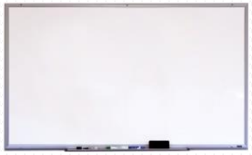
Deci tabela semntică este deschisă ⇒ formula nu este tautologie

Exemple

$$\overset{?}{\models} (\exists x)\,(A(x) \wedge B(x)) \rightarrow (\exists x)\,A(x) \wedge (\exists x)\,B(x)$$

$$\overset{?}{\models} (\forall x)\,(A(x) \vee B(x)) \rightarrow (\forall x)\,A(x) \vee (\forall x)\,B(x)$$

$$\overset{?}{\models} (\exists y)\,(\forall x)\,P(x, y)$$



$\neg(\exists y)\ (\forall x)\ P(x, y)$ (1)√

γ (1), a - constantă implicită

$\neg\ (\forall x)\ P(x, a)$ (2)√

$\neg(\exists y)\ (\forall x)\ P(x, y)$ (1')√

δ (2), b - constantă nouă

$\neg\ P(b, a)$

γ (1'), b - constantă existentă

$\neg\ (\forall x)\ P(x, b)$ (3)√

$\neg(\exists y)\ (\forall x)\ P(x, y)$ (1'')

δ (3), c - constantă nouă

$\neg\ P(c, b)$

γ (1''), c - constantă existentă

…

Deci am intrat în ciclu infinit, deci nu putem decide tipul formulei (pp. nu are loc – identificăm un anti-model)

Semi-decidabilitatea calculului predicativ

- Pentru cazul logicii predicatelor de ordinul I, arborele poate fi infinit datorită combinării regulilor de tip γ şi δ.

- Dacă arborele asociat negaţiei unei formule predicative este *finit*, atunci *se poate* decide dacă formula respectivă este o tautologie sau nu, dar dacă arborele este *infinit*, *nu* se poate decide nimic asupra validităţii formulei.

Substituţii

Definiţie: O *substituţie* este o funcţie definită pe mulţimea variabilelor, *Var* cu valori în mulţimea termenilor, *TERM*. Se notează cu $\theta=[x_1\leftarrow t_1,\ldots, x_k\leftarrow t_k]$, reprezentând o mulţime finită de înlocuiri de variabile cu termeni. $x_1,\ldots, x_k$ sunt variabile distincte, iar $t_1,\ldots, t_k$ sunt termeni, astfel încât $\forall i = 1,\ldots,k$, $t_i \neq x_i$ și x_i nu este *subtermen* al lui t_i.

- $dom(\theta) = \{x_1,\ldots, x_k\}$ se numeşte domeniul substituţiei θ.
- ε – substituția vidă

- $\varphi, \xi, \psi, \eta, \theta, \lambda$

Aplicarea substituției

$\theta=[x_1 \leftarrow t_1,\ldots, x_k \leftarrow t_k]$ asupra formulei U se definește recursiv:

- $\theta(x_i) = t_i$, $x_i \in dom(\theta)$; $\theta(x) = x$, $x \notin dom(\theta)$;
- $\theta(c) = c$, c – constantă;
- $\theta(f(t_1,\ldots, t_n)) = f(\theta(t_1),\ldots, \theta(t_n))$, $f \in \mathcal{F}_n$;
- $\theta(P(t_1,\ldots, t_n)) = P(\theta(t_1),\ldots, \theta(t_n))$, $P \in \mathcal{P}_n$;
- $\theta(\neg U) = \neg\, \theta(U)$;
- $\theta(U \wedge V) = \theta(U) \wedge \theta(V)$;
- $\theta(U \vee V) = \theta(U) \vee \theta(V)$;
- $\theta(U \rightarrow V) = \theta(U) \rightarrow \theta(V)$;
- $\theta(U \leftrightarrow V) = \theta(U) \leftrightarrow \theta(V)$.

Compunerea substituțiilor

$\theta_1=[x_1 \leftarrow t_1,\ldots, x_k \leftarrow t_k]$ și $\theta_2=[y_1 \leftarrow s_1,\ldots, y_k \leftarrow s_k]$

$$\theta = \theta_1 \circ \theta_2 = [x_i \leftarrow \theta_2(\, t_i \,) \mid x_i \in dom(\theta_1), x_i \neq \theta_2(\, t_i \,)] \cup$$
$$[y_i \leftarrow s_j \mid y_i \in dom(\theta_2) \setminus dom(\theta_1)]$$

- Obs.: Nu întotdeauna compunerea unor substituții este o substituție.

Exemple

$\theta_1=[x_1 \leftarrow t_1,\ldots, x_k \leftarrow t_k]$ și $\theta_2=[y_1 \leftarrow s_1,\ldots, y_k \leftarrow s_k]$

$$\theta = \theta_1 \circ \theta_2 = [x_i \leftarrow \theta_2(t_i) \mid x_i \in dom(\theta_1), x_i \neq \theta_2(t_i)] \cup$$
$$[y_i \leftarrow s_j \mid y_i \in dom(\theta_2) \setminus dom(\theta_1)]$$

$\theta_1=[x \leftarrow y, z \leftarrow a, u \leftarrow f(y)]$ și $\theta_2=[y \leftarrow a, t \leftarrow g(b), v \leftarrow w]$

$$\theta = \theta_1 \circ \theta_2 = [x \leftarrow a, z \leftarrow a, u \leftarrow f(a), y \leftarrow a, t \leftarrow g(b), v \leftarrow w]$$

$\theta_3=[x \leftarrow y, z \leftarrow a, u \leftarrow f(t)]$ și $\theta_4=[y \leftarrow a, t \leftarrow g(u), v \leftarrow w]$

$\theta = \theta_3 \circ \theta_4 =$

$= [x \leftarrow a, z \leftarrow a, u \leftarrow f(g(u)), y \leftarrow a, t \leftarrow g(u), v \leftarrow w]$ – nu este o substituție

Proprietăți ale operației de compunere

- Element neutru: ε – substituția vidă:

$$\varepsilon\theta = \theta\varepsilon = \theta, \forall\theta \text{ – substituție}$$

- Asociativitatea: $\theta_1(\theta_2\theta_3) = (\theta_1\theta_2)\theta_3 = \theta_1\theta_2\theta_3$

- În general compunerea nu este comutativă

Unificatori

- O substituţie θ se numeşte *unificator* al termenilor t_1 şi t_2 dacă $\theta(t_1) = \theta(t_2)$. Termenul $\theta(t_1)$ se numeşte *instanţa comună* a termenilor unificaţi.
- Un *unificator al mulţimii* de formule $\{U_1, U_2, \ldots, U_n\}$ este o substituţie θ cu proprietatea: $\theta(U_1) = \ldots = \theta(U_n)$.

- *Cel mai general unificator* ***(mgu)*** este un unificator μ cu proprietatea că orice alt unificator θ se obţine din compunerea lui μ cu o altă substituţie λ: $\theta = \mu\,\lambda$.

Algoritm pentru determinarea celui mai general unificator a doi literali (1)

Date de intrare: l_1=$P_1(t_{1_1}, t_{1_2},\ldots,t_{1_n})$ şi l_2=$P_2(t_{2_1}, t_{2_2},\ldots,t_{2_k})$ doi literali

Date de ieşire: *mgu*(l_1, l_2) sau "l_1, l_2 nu sunt unificabili"

dacă ($P_1 \neq P_2$) // simbolurile predicative sunt diferite

atunci *scrie "l_1, l_2 nu sunt unificabili"*; STOP;

sf_dacă

dacă($n \neq k$) // aritate diferită pentru acelaşi simbol predicativ

atunci *scrie "l_1, l_2 nu sunt unificabili"*; STOP;

sf_dacă

$\theta \leftarrow \varepsilon$; // iniţializare cu substituţia vidă

Algoritm pentru determinarea celui mai general unificator a doi literali (2)

cât_timp ($\theta(l_1) \neq \theta(l_2)$)

Din $\theta(l_1), \theta(l_2)$ se determină cele mai din stânga simbol de funcţie, constantă sau variabilă diferite şi notăm cu t_1 şi t_2 termenii lor corespunzători.

dacă (*niciunul dintre* t_1 *şi* t_2 *nu este variabilă* sau *unul este subtermenul celuilalt*)

atunci *scrie "l_1, l_2 nu sunt unificabili"*; STOP;

sf_dacă

dacă (t_1 *este variabilă*)

atunci $\lambda = [t_1 \leftarrow t_2]$;

altfel $\lambda = [t_2 \leftarrow t_1]$;

sf_dacă

$\theta \leftarrow \theta\,\lambda$;

dacă (θ *nu este substituție*)

atunci *scrie "l_1, l_2 nu sunt unificabili"*; STOP;

sf_dacă

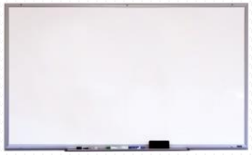
sf_cât_timp

scrie "*l_1 şi l_2 sunt unificabili,mgu(l_1, l_2)=* " θ

Sf_algoritm

Exerciții

- $P(a, x, g(f(y)))$ și $Q(f(y), f(z), g(z))$ — Nu au același simbol de predicat
- $P(a, x, g(f(y)), b)$ și $P(f(y), f(z), g(z))$ — Nu au aceeași aritate
- $P(a, x, g(f(y)))$ și $P(f(y), f(z), g(z))$ — $\xcancel{[a \leftarrow f(y)]}$
- $P(x, g(f(a)), x)$ și $P(f(y), z, h(y, f(y)))$



$A_1=P(x, g(f(a)), x)$ și $A_2=P(f(y), z, h(y, f(y)))$

$\theta_1 = [x \leftarrow f(y)]$

$\theta_1(A_1) = P(f(y), g(f(a)), f(y))$

$\theta_1(A_2) = P(f(y), z, h(y, f(y)))$

$\theta_2 = [z \leftarrow g(f(a))]$

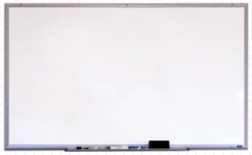
$\theta_2(\theta_1(A_1)) = P(f(y), g(f(a)), f(y))$

$\theta_2(\theta_1(A_2)) = P(f(y), g(f(a)), h(y, f(y)))$

Nu sunt unificabili pentru că nu avem variabilă care să se substituie (*f* și *h* sunt ambele simboluri de funcții)

Exerciții

- $\xcancel{P(a, x, g(f(y))) \text{ și } Q(f(y), f(z), g(z))}$ Nu au același simbol de predicat
- $\xcancel{P(a, x, g(f(y)), b) \text{ și } P(f(y), f(z), g(z))}$ Nu au aceeași aritate
- $\xcancel{P(a, x, g(f(y))) \text{ și } P(f(y), f(z), g(z))}$ $\xcancel{[a \leftarrow f(y)]}$
- $\xcancel{P(x, g(f(a)), x) \text{ și } P(f(y), z, h(y, f(y)))}$
- $P(a, h(x,u), f(g(y))) \text{ și } P(z, h(z, u), f(u))$



$A_3 = P(a, h(x,u), f(g(y)))$ și $A_4 = P(z, h(z, u), f(u))$

$\theta_1 = [z \leftarrow a]$

$\theta_1(A_3) = P(a, h(x,u), f(g(y)))$

$\theta_1(A_4) = P(a, h(a, u), f(u))$

$\theta_2 = [x \leftarrow a]$

$\theta_2(\theta_1(A_3)) = P(a, h(a,u), f(g(y)))$

$\theta_2(\theta_1(A_4)) = P(a, h(a, u), f(u))$

$\theta_3 = [u \leftarrow g(y)]$

$\theta_3(\theta_2(\theta_1(A_3)) = P(a, h(a, g(y)), f(g(y)))$

$\theta_3(\theta_2(\theta_1(A_4)) = P(a, h(a, g(y)), f(g(y)))$

Deci $\theta_3(\theta_2(\theta_1(A_3)) = \theta_3(\theta_2(\theta_1(A_4))$, deci A_3 și A_4 sunt unificabile și $mgu(A_3,A_4)= \theta_1 \circ \theta_2 \circ \theta_3 = [z \leftarrow a, x \leftarrow a, u \leftarrow g(y)]$

Logică computațională
Curs 10

Lector dr. Pop Andreea-Diana

Sistem formal (axiomatic) asociat Rezoluției predicative

- $\mathrm{Res}^{\mathrm{Pr}} = (\Sigma_{\mathrm{Res}}^{\mathrm{Pr}}, F_{\mathrm{Res}}^{\mathrm{Pr}}, A_{\mathrm{Res}}^{\mathrm{Pr}}, R_{\mathrm{Res}}^{\mathrm{Pr}})$
  - $\Sigma_{\mathrm{Res}}^{\mathrm{Pr}} = \Sigma_{\mathrm{Pr}} \setminus \{ \forall, \exists, \wedge, \rightarrow, \leftrightarrow \}$ – **alfabetul**
  - $F_{\mathrm{Res}}^{\mathrm{Pr}} \cup \{\square\}$ – **mulţimea formulelor bine-formate**
    - $F_{\mathrm{Res}}^{\mathrm{Pr}}$ mulţimea tuturor clauzelor ce se pot forma folosind alfabetul $\Sigma_{\mathrm{Res}}^{\mathrm{Pr}}$
    - $\square$ **- clauza vidă** care nu conţine nici un literal, simbolizează inconsistenţa
  - $A_{\mathrm{Res}}^{\mathrm{Pr}} = \emptyset$ – **mulțimea axiomelor**
  - $R_{\mathrm{Res}}^{\mathrm{Pr}} = \{res^{\mathrm{Pr}}, fact\}$ **mulţimea regulilor de inferenţă** care conţine:

Reguli de inferenţă predicative

- **regula rezoluţiei predicative**:

  $A \vee l_1, B \vee \neg l_2 \vdash^{Pr}_{res} \theta(A) \vee \theta(B),$

  unde $\theta = mgu(l_1, l_2)$ și $A, B \in F^{Pr}_{Res}$

  - $C_1 = A \vee l_1$, $C_2 = B \vee \neg l_2$ clauzele care rezolvă,

    dacă literalii l_1 și l_2 sunt unificabili

  - Rezolventul binar $C_3 = \mathrm{Res}^{Pr}_{\theta}(C_1, C_2) = \theta(A) \vee \theta(B)$

- **regula factorizării**:

  $C \vdash_{fact} C'$, C' – factor al lui C

  unde $C = l_1 \vee l_2 \vee \ldots \vee l_k \vee l_{k+1} \vee \ldots \vee l_n$,

  $\lambda = mgu(l_1, l_2, \ldots, l_k)$

  $C' = \lambda(l_k) \vee \lambda(l_{k+1}) \vee \ldots \vee \lambda(l_n)$

Exemple

$C_1 = P(x) \vee \underline{Q(y,b)}$

$C_2 = \underline{\neg\, Q(z,z)} \vee R(g(z))$

$\mathrm{Rez}_{\theta}^{\mathrm{Pr}}(C_1, C_2) = \theta(\, P(x)\,) \vee \;\theta(\, R(g(z))) = P(x) \vee R(g(b))$

$\theta = [y \leftarrow z] \circ [z \leftarrow b] = [y \leftarrow b, z \leftarrow b]$

$C_3 = \underline{P(x) \vee P(y) \vee P(a)} \vee R(g(z),x)$

$\lambda = [x \leftarrow a, y \leftarrow a]$

$\mathrm{Fact}_{\lambda}\,(C_3) = \lambda\,(\underline{P(a)}) \vee \lambda\,(\, R(g(z),a)) = \underline{P(a)} \vee R(g(z),a)$

Teoremă

Fie $U_1, U_2, \ldots, U_n$ şi V formule predicative.

- $\vdash V$ dacă şi numai dacă $(\neg V)^C \vdash_{res}^{\Pr} \square$
- $U_1, U_2, \ldots, U_n \vdash V$ dacă şi numai dacă
  $$\{U_1{}^C, U_2{}^C, \ldots, U_n{}^C, (\neg V)^C\} \vdash_{res} \square$$

Observație: Variabilele din clauze distincte se recomandă să fie distincte.

Algoritmul rezoluției predicative:

Date de intrare: $U_1, U_2, \ldots, U_n$, V – formule predicative

Date de ieșire: ”are loc $U_1, U_2, \ldots, U_n \vdash V$” sau ”nu are loc $U_1, U_2, \ldots, U_n \vdash V$”

Se construiește $S = \{ U_1^C, U_2^C, \ldots, U_n^C, (\neg V)^C \}$

Repetă

Se selectează literalii l_1, l_2 și clauzele C_1, C_2 astfel încât sunt clauze sau factori ai unor clauze din S

Fie $l_1 \in C_1$ și $\neg\, l_2 \in C_2$, respectiv $l_1, l_2 \in C_1$

Dacă l_1 și l_2 sunt unificabili cu $\theta = mgu(l_1, l_2)$

Atunci

$C = \mathrm{Res}^{\mathrm{Pr}}_{\theta}(C_1, C_2)$ respectiv $C = fact_{\theta}(C_1)$

Algoritmul rezoluției predicative - continuare

Dacă $C = \square$

Atunci Scrie ”are loc $U_1,U_2,\ldots,U_n \vdash V$”; **STOP**

Altfel $S = S \cup \{C\}$

Sfârșit_dacă

Sfârșit_dacă

Până când nu se mai pot deriva noi rezolvenți **sau** un număr fixat de iterații au fost executate

Dacă nu se mai pot deriva noi rezolvenți

Atunci Scrie ”nu are loc $U_1,U_2,\ldots,U_n \vdash V$”

Altfel Scrie ”nu se poate decide dacă are loc sau nu $U_1,U_2,\ldots,U_n \vdash V$”

Sfârșit_dacă

Sfârșit algoritm

Strategii și rafinări ale rezoluției predicative

- Strategii:
  - Strategia eliminării !unificarea, factorizarea
  - Strategia saturării pe nivele
  - Strategia mulțimii suport

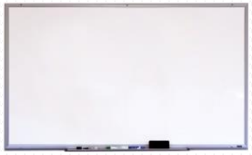
- Rafinări:
  - Rezoluția blocării
  - Rezoluția liniară
    - input
    - unit

Exemple

$\vdash (\exists\, x)(\forall\, y)\, (\mathrm{P}(x,y) \rightarrow \mathrm{R}(x)) \rightarrow ((\forall\, x)(\forall\, y)\mathrm{P}(x,y) \rightarrow (\exists\, z)\mathrm{R}(z))$

$\vdash (\exists y)\, (\exists\, x)\, (P(x, y) \wedge P(y, y)\,) \vee (\exists y)\, (\exists x)\, (\neg\, P(y, x) \wedge \neg\, P(y, y))$

$\vdash (\exists x)\, (\forall y)\, (\, P(x) \wedge \neg\, P(y)\,) \vee \neg(\exists\, z)P(z)$



Exemplu (1) obținerea mulțimii de clauze

$\models^{?} (\exists\, x)(\forall\, y)\,(P(x,y)\rightarrow R(x)) \rightarrow ((\forall\, x)(\forall\, y)P(x,y)\rightarrow(\exists\, z)R(z)) \overset{\text{ITD}}{\Rightarrow}$

$(\exists\, x)(\forall\, y)\,(P(x,y)\rightarrow R(x)) \models^{?} (\forall\, x)(\forall\, y)P(x,y)\rightarrow(\exists\, z)R(z) \overset{\text{ITD}}{\Rightarrow}$

$(\exists\, x)(\forall\, y)\,(P(x,y)\rightarrow R(x)),\ (\forall\, x)(\forall\, y)P(x,y) \models^{?} (\exists\, z)R(z)$

$U_1 = (\exists\, x)(\forall\, y)\,(P(x,y)\rightarrow R(x)) \equiv (\exists\, x)(\forall\, y)\,(\neg P(x,y)\vee R(x)) = U_1^{\mathrm{P}}$

$x\leftarrow a\ U_1^{\mathrm{S}}=(\forall\, y)\,(\neg P(a,y)\vee R(a))$

$U_1^{\mathrm{Sq}} = \neg P(a,y)\vee R(a) = U_1^{\mathrm{C}} = C_1$

$U_2=(\forall\, x)(\forall\, y)P(x,y)$

$U_2^{\mathrm{C}} = P(x,y) = U_2^{\mathrm{C}} = C_2$

$\neg V=\neg((\exists\, z)R(z))$

$\neg V \equiv (\forall z)\,\neg\, R(z)$

$(\neg V)^{\mathrm{Sq}} = \neg\, R(z) = (\neg V)^{\mathrm{C}} = C_3$

$S=\{\neg P(a,y)\vee R(a),\ P(x,y),\ \neg\, R(z)\}$

Exemplu (1) aplicarea metodei rezoluției

$S = \{\neg P(a,y) \vee R(a), P(x,y), \neg R(z)\}$

$\text{Rez}_{\theta}^{Pr}(C_1, C_2) = R(a) = C_4$

$\theta = [x \leftarrow a]$

$\text{Rez}_{\lambda}^{Pr}(C_4, C_3) = \square$

$\lambda = [z \leftarrow a]$

($\overset{\text{TCC}}{\Rightarrow}$ S este inconsistentă)

$\Rightarrow$ (pe baza teoremei din curs) că $U_1, U_2 \vdash V$, și din raționamentul prin respingere (sau aplicând de 2 ori TD)

$\vdash (\exists\, x)(\forall\, y)\,(P(x,y) \rightarrow R(x)) \rightarrow ((\forall\, x)(\forall\, y)P(x,y) \rightarrow (\exists\, z)R(z))$

Exemplu (2)
obținerea mulțimii de clauze

$\overset{?}{\vdash} (\exists y)(\exists x)(P(x,y) \wedge P(y,y)) \vee (\exists y)(\exists x)(\neg P(y,x) \wedge \neg P(y,y)) \overset{\text{not.}}{=} U$

$\neg U \equiv \neg\, ((\exists y)(\exists x)(P(x,y) \wedge P(y,y)) \vee (\exists y)(\exists x)(\neg P(y,x) \wedge \neg P(y,y)))$

$\neg U \equiv (\forall y)(\forall x)(\neg P(x,y) \vee \neg P(y,y)) \wedge (\forall y)(\forall x)(P(y,x) \vee P(y,y))$

$\neg U \equiv (\forall y)(\forall x)(\neg P(x,y) \vee \neg P(y,y)) \wedge (\forall t)(\forall z)(P(t,z) \vee P(t,t))$

$\neg U \equiv (\forall y)(\forall x)(\forall t)(\forall z)((\neg P(x,y) \vee \neg P(y,y)) \wedge (P(t,z) \vee P(t,t)))$

$\neg U^{Sq} = (\neg P(x,y) \vee \neg P(y,y)) \wedge (P(t,z) \vee P(t,t)) = \neg \mathrm{U}^{\mathrm{C}}$

$C_1 = \neg P(x,y) \vee \neg P(y,y)$

$C_2 = P(t,z) \vee P(t,t)$

$S = \{C_1, C_2\}$

Exemplu (2) aplicarea metodei rezoluției

$S=\{ C_1, C_2\}$

$\mathrm{Rez}_{\theta}^{\mathrm{Pr}}(C_1, C_2) = \neg P(x, t) \vee P(t, z) = C_3$

$\theta = [y \leftarrow t]$

$\mathrm{Fact}_{\lambda}^{\mathrm{Pr}}(\mathrm{C}_2) = P(t, t) = C_4$

$\lambda = [\mathrm{z} \leftarrow t]$

$\mathrm{Fact}_{\xi}^{\mathrm{Pr}}(C_1) = \neg P(y, y) = C_5$

$\xi = [\mathrm{x} \leftarrow \mathrm{y}]$

$\mathrm{Rez}_{\theta}^{\mathrm{Pr}}(C_4, C_5) = \square$, deci, pe baza teoremei din curs $\Rightarrow \vdash U$

Exemplu (3)
obținerea mulțimii de clauze

$$\overset{?}{\vdash} (\exists x)(\forall y)(P(x) \wedge \neg P(y)) \vee \neg(\exists z)P(z) \overset{not.}{=} U$$

$$\neg U \equiv \neg ((\exists x)(\forall y)(P(x) \wedge \neg P(y)) \vee \neg(\exists z)P(z))$$

$$\neg U \equiv (\forall x)(\exists y)(\neg P(x) \vee P(y)) \wedge (\exists z)P(z)$$

$$\neg U \equiv (\exists z)(\forall x)(\exists y)\ ((\neg P(x) \vee P(y)) \wedge P(z)) = (\neg U)^{P}$$

$$z \leftarrow a,\ y \leftarrow f(x)$$

$$(\neg U)^{S} = (\forall x)((\neg P(x) \vee P(f(x))) \wedge P(a))$$

$$(\neg U)^{Sq} = (\neg P(x) \vee P(f(x))) \wedge P(a) = (\neg U)^{C}$$

$$C_1 = \neg P(x) \vee P(f(x))$$

$$C_2 = P(a)$$

$$S = \{C_1, C_2\}$$

Exemplu (3) aplicarea metodei rezoluției

$S=\{ C_1, C_2\}$

$\mathrm{Rez}_{\theta}^{\mathrm{Pr}}(C_1, C_2) = P(f(a)) = C_3$

$\theta=[x\leftarrow a]$

$\mathrm{Rez}_{\lambda}^{\mathrm{Pr}}(C_1, C_3) = P(f(f(a))) = C_4$

$\lambda=[x\leftarrow f(a)]$

Ciclu infinit, deci nu putem decide dacă formula este sau nu teoremă.

Completitudinea și corectitudinea

- Toate rafinările şi strategiile rezolutive păstrează completitudinea şi corectitudinea.
- Combinarea lor poate impune prea multe restricții și deși mulțimea inițială de clauze este inconsistentă, s-ar putea să nu se poată deriva clauza vidă.
- sunt complete:
  - rezoluţia generală + strategia eliminării
  - rezoluţia generală + strategia mulţimii suport
  - rezoluţia generală + strategia mulţimii suport + strategia eliminării
  - rezoluția liniară + strategia eliminării
  - rezoluţia liniară + strategia mulţimii suport
- nu sunt complete:
  - rezoluția blocării + strategia eliminării
  - rezoluția blocării + strategia mulţimii suport
  - rezoluția blocării + rezoluția liniară
  - rezoluţia unitară
  - rezoluția de intrare

Completitudinea rezoluției de intrare

Definiții:

- O *clauză* se numește *pozitivă* dacă aceasta conține literali pozitivi.
- O *clauză* se numește *negativă* dacă aceasta conține doar literali negativi.
- O clauză se numește *clauză Horn* dacă aceasta conține un singur literal pozitiv, ceilalți fiind negativi.

Teoremă:

- Rezoluția de intrare este completă pe o mulțime de clauze Horn, cu o clauză negativă ca și clauză de vârf (PROLOG).

  - H_i: $l_1 \wedge l_2 \wedge \ldots \wedge l_n \rightarrow v$, $i \in \{1, \ldots, k\}$
  - C: $n_1 \wedge n_2 \wedge \ldots \wedge n_m$?

Limbajul Prolog - ”o privire pe furiș”

- tată(B,C):-bărbat(B),părinte(B,C).
- mamă(M,C):-femeie(M),părinte(M,C).
- fiu(P,S):-bărbat(S),părinte(P,S).
- Fiică(P,D):-femeie(D),părinte(P,D).
- frate(A,F):-părinte(P,A),părinte(P,F).

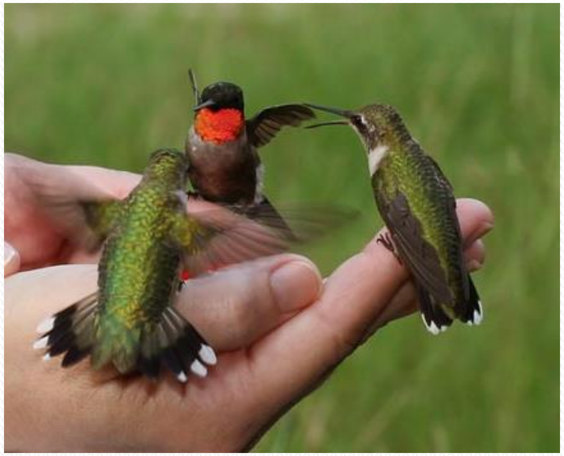
- părinte(”Petru”,”Vlad”)
- părinte(”Petru”,”Bogdan”)

- ? frate(“Vlad”,“Bogdan”)?

Problemă de modelare

- Toate păsările colibri sunt bogat colorate.
- Păsările mari nu se hrănesc cu miere.
- Păsările care nu se hrănesc cu miere nu sunt bogat colorate.
- Piky este o pasăre colibri.

- Toate păsările colibri sunt păsări mici.



Transformarea din limbaj natural în limbaj logic

”pasărea x este bogat colorată” $\stackrel{\text{not.}}{=} B(x)$

”pasărea x se hrănește cu miere” $\stackrel{\text{not.}}{=} H(x)$

”pasărea x este o parsăre colibri” $\stackrel{\text{not.}}{=} C(x)$

”pasărea x este mică” $\stackrel{\text{not.}}{=} M(x)$

Constantă: Piky $\stackrel{\text{not.}}{=} a$

$U_1=(\forall \mathrm{x})\ (C(x)\rightarrow B(x))$

$U_2=(\forall \mathrm{x})\ (\neg M(x)\rightarrow\neg H(x))$

$U_3=(\forall \mathrm{x})\ (\neg H(x)\rightarrow\neg B(x))$

$U_4=C(a)$

$V=(\forall x)(C(x)\rightarrow M(x))$