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

$$\begin{tikzcd} & A \vee B \arrow[dl, dash] \arrow[dr, dash] & \\ A & & B \end{tikzcd}$$

$$\begin{tikzcd} & \neg(A \wedge B) \arrow[dl, dash] \arrow[dr, dash] & \\ \neg A & & \neg B \end{tikzcd}$$

$$\begin{tikzcd} & A \to B \arrow[dl, dash] \arrow[dr, dash] & \\ \neg A & & B \end{tikzcd}$$

Reguli de descompunere a formulelor (2)

- regula γ

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

$|$ $c_1, c_2, \ldots, c_n$ – toate constantele existente pe ramură

$A(c_1)$

$|$

$A(c_2)$

$|$

$\vdots$

$|$

$A(c_n)$

$|$

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

copie formulă

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

$|$

$\neg\, A(c_1)$

$|$

$\neg\, A(c_2)$

$|$

$\vdots$

$|$

$\neg\, A(c_n)$

$|$

$\neg\, (\exists\, x)\, A(x)$ – copie formulă

- regula δ

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

$|$ a – constantă nou introdusă

$A(a)$

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

$|$

$\neg\ A(a)$

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

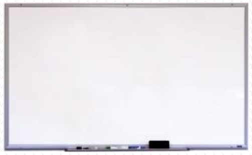
- O formulă U este teoremă (tautologie) dacă şi numai dacă există o tabelă semantică închisă pentru formula $\neg U$.

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

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



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

γ (5), a - constantă existentă

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

$\neg(\exists x)\,A(x)$ (5') (copia)

⊗

$\neg(\exists x)\,B(x)$ (6)√

γ (6), a - constantă existentă

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

$\neg(\exists x)\,B(x)$ (6') (copia)

⊗

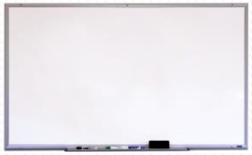
TCC

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

Exemple

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

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



$\neg((\forall x)\,(A(x) \vee B(x)) \rightarrow (\forall x)\,A(x) \vee (\forall x)\,B(x))$ (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 ⊙ ⊗

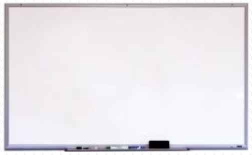
Deci tabela semntică este deschisă ⇒ formula nu este tautologie

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

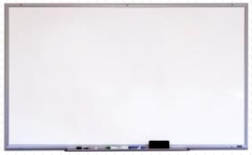
sf_cât_timp

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

Sf_algoritm

Exerciții

- $\xcancel{P(a, x, g(f(y))) \text{ și } Q(f(y), f(z), g(z))}$ 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
- $\xcancel{P(a, x, g(f(y))) \text{ și } P(f(y), f(z), g(z))}$ $\xcancel{[a \leftarrow f(y)]}$
- $P(x, g(f(a)), x) \text{ și } P(f(y), z, h(y, f(y)))$



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

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

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

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

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

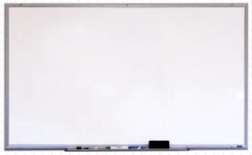
$\theta_2(\theta_1(A_1)) = P(f(y), g(f(a)), f(y))$

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

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

Exerciții

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



$A_3 = P(a, h(x,u), f(g(y)))$ 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 $mgu(A_3,A_4) = \theta_1 \circ \theta_2 \circ \theta_3 = [z \leftarrow a, x \leftarrow a, u \leftarrow g(y)]$