

Seminar - Programare Logica si Functionala

Seria 36

2024

1 Algoritmul de unificare



Pana si Grasa stie algoritmul de unificare, aplica *metoda tabelului* (singurul seminar la care au performat chiar si cei de la 362)

Teorie

O *substituție* este o funcție parțială de la variabile la termeni, adică $\sigma : V \rightarrow Trm_{\mathcal{L}}$. Un *unificator* pentru doi termeni t_1 și t_2 este o substituție θ astfel încât $\theta(t_1) = \theta(t_2)$. Un unificator ν pentru t_1 și t_2 este un *cel mai general unificator* dacă pentru orice alt unificator ν' pentru t_1 și t_2 , există o substituție μ astfel încât $\nu' = \nu; \mu$.

Algoritmul de unificare:

	Lista soluție S	Lista de rezolvat R
Inițial	\emptyset	$t_1 \doteq t'_1, \dots, t_n \doteq t'_n$
SCOATE	S	$R', t \doteq t$
	S	R'
DESCOMPUNE	S	$R', f(t_1, \dots, t_n) \doteq f(t'_1, \dots, t'_n)$
	S	$R', t_1 \doteq t'_1, \dots, t_n \doteq t'_n$
REZOLVĂ	S	$R', x \doteq t$ sau $t \doteq x$, x nu apare în t
	$x \doteq t, S[x \leftarrow t]$	$R'[x \leftarrow t]$
Final	S	\emptyset

Algoritmul *se termină normal* dacă $R = \emptyset$ (în acest caz, în S are un unificator pentru termenii din lista inițială R).

Algoritmul este oprit cu concluzia *inexistenței unui unificator* dacă:

- (i) În R există o ecuație de forma $f(t_1, \dots, t_n) \doteq g(t'_1, \dots, t'_k)$ cu $f \neq g$. Simbolurile de constantă se consideră simboluri de funcție de aritate 0.
- (ii) În R există o ecuație de forma $x \doteq t$ sau $t \doteq x$ și variabila x apare în termenul t .

(Exercițiul 1) Considerăm

- x, y, z, u, v variabile,

- a, b, c simboluri de constantă,
- $h, g, (-)^{-1}$ simboluri de funcție de aritate 1,
- $f, *, +$ simboluri de funcție de aritate 2,
- p simbol de funcție de aritate 3.

Aplicați algoritmul de unificare de mai sus pentru a găsi un unificator pentru termenii:

- 1) $p(a, x, h(g(y)))$ și $p(z, h(z), h(u))$
- 2) $f(h(a), g(x))$ și $f(y, y)$
- 3) $p(a, x, g(x))$ și $p(a, y, y)$
- 4) $p(x, y, z)$ și $p(u, f(v, v), u)$
- 5) $f(x, f(x, x))$ și $f(g(y), f(z, g(a)))$
- 6) $x + (y * y)$ și $(y * y) + z$
- 7) $(x * y) * z$ și $u * u^{-1}$
- 8) $x * y$ și $u * u^{-1}$
- 9) $x * y$ și $x * (y * (u * v)^{-1})$
- 10) $x * y$ și $y * (u * v)^{-1}$
- 11) $f(g(x), x)$ și $f(y, y)$
- 12) $p(x, z, z)$ și $p(y, y, b)$
- 13) $p(a, u, h(x))$ și $p(y, f(y, z), z)$
- 14) $f(x, f(b, x))$ și $f(f(y, a), f(b, f(z, z)))$
- 15) $p(x, b, x)$ și $p(y, y, c)$
- 16) $f(x, y), f(h(x), x)$ și $f(x, b)$
- 17) $f(x, f(x, g(y))), f(u, z)$ și $f(g(y), y)$
- 18) $f(f(x, y), x), f(g(y), z)$ și $f(u, h(z))$
- 19) $f(f(x, y), x), f(v, u)$ și $f(u, h(z))$

Demonstrație:

1.

S	R	
\emptyset	$p(a, x, h(g(y))) = p(z, h(z), h(u))$	DESCOMPUNE
\emptyset	$a \doteq z, x \doteq h(z), h(g(y)) \doteq h(u)$	REZOLVĂ
$z \doteq a$	$x \doteq h(a), h(g(y)) \doteq h(u)$	REZOLVĂ
$z \doteq a, x \doteq h(a)$	$h(g(y)) \doteq h(u)$	DESCOMPUNE
$z \doteq a, x \doteq h(a)$	$g(y) \doteq u$	REZOLVĂ
$z \doteq a, x \doteq h(a), u \doteq g(y)$	\emptyset	

- $\nu = \{z/a, x/h(a), u/g(y)\}$ este cgu.

2.

S	R	
\emptyset	$f(h(a), g(x)) \doteq f(y, y)$	DESCOMPUNE
\emptyset	$y \doteq h(a), y \doteq g(x)$	REZOLVĂ
$y \doteq h(a)$	$g(x) \doteq h(a)$	EȘEC

- Nu există unificator.

3.

S	R	
\emptyset	$p(a, x, g(x)) \dot{=} p(a, y, y)$	DESCOMPUNE
\emptyset	$a \dot{=} a, x \dot{=} y, y \dot{=} g(x)$	SCOATE
\emptyset	$x \dot{=} y, y \dot{=} g(x)$	REZOLVĂ
$x \dot{=} y$	$y \dot{=} g(y)$	EȘEC

- Nu există unificator.

4.

S	R	
\emptyset	$p(x, y, z) \dot{=} p(u, f(v, v), u)$	DESCOMPUNE
\emptyset	$x \dot{=} u, y \dot{=} f(v, v), z \dot{=} u$	REZOLVĂ
$x \dot{=} u$	$y \dot{=} f(v, v), z \dot{=} u$	REZOLVĂ
$y \dot{=} f(v, v), x \dot{=} u$	$z \dot{=} u$	REZOLVĂ
$z \dot{=} u, y \dot{=} f(v, v), x \dot{=} u$		

- $\nu = \{z/u, y/f(v, v), x/u\}$ este cgu.

5.

S	R	
\emptyset	$f(x, f(x, x)) \dot{=} f(g(y), f(z, g(a)))$	DESCOMPUNE
\emptyset	$x \dot{=} g(y), f(x, x) \dot{=} f(z, g(a))$	REZOLVĂ
$x \dot{=} g(y)$	$f(g(y), g(y)) \dot{=} f(z, g(a))$	DESCOMPUNE
$x \dot{=} g(y)$	$z \dot{=} g(y), g(y) \dot{=} g(a)$	REZOLVĂ
$z \dot{=} g(y), x \dot{=} g(y)$	$g(y) \dot{=} g(a)$	DESCOMPUNE
$z \dot{=} g(y), x \dot{=} g(y)$	$y \dot{=} a$	REZOLVĂ
$y \dot{=} a, z \dot{=} g(a), x \dot{=} g(a)$		

- $\nu = \{x/g(a), z/g(a), y/a\}$ este cgu.

6.

S	R	
\emptyset	$x + (y * y) \dot{=} (y * y) + z$	DESCOMPUNE
\emptyset	$x \dot{=} y * y, z \dot{=} y * y$	REZOLVĂ
$x \dot{=} y * y$	$z \dot{=} y * y$	REZOLVĂ
$z \dot{=} y * y, x \dot{=} y * y$		

- $\nu = \{x/y * y, z/y * y\}$ este cgu.

7.

S	R	
\emptyset	$(x * y) * z \dot{=} u * u^{-1}$	DESCOMPUNE
\emptyset	$u \dot{=} x * y, z \dot{=} u^{-1}$	REZOLVĂ
$u \dot{=} x * y$	$z \dot{=} (x * y)^{-1}$	REZOLVĂ
$z \dot{=} (x * y)^{-1}, u \dot{=} x * y$		

- $\nu = \{z/(x * y)^{-1}, u/x * y\}$ este cgu.

8.

S	R	
\emptyset	$x * y \dot{=} u * u^{-1}$	DESCOMPUNE
\emptyset	$x \dot{=} u, y \dot{=} u^{-1}$	REZOLVĂ
$x \dot{=} u$	$y \dot{=} u^{-1}$	REZOLVĂ
$y \dot{=} u^{-1}, x \dot{=} u$		

- $\nu = \{y/u^{-1}, x/u\}$ este cgu.

9.

S	R	
\emptyset	$x * y \dot{=} x * (y * (u * v)^{-1})$	DESCOMPUNE
\emptyset	$x \dot{=} x, y \dot{=} y * (u * v)^{-1}$	SCOATE
\emptyset	$y \dot{=} y * (u * v)^{-1}$	EȘEC

- Nu există unificator.

10.

S	R	
\emptyset	$x * y \dot{=} y * (u * v)^{-1}$	DESCOMPUNE
\emptyset	$x \dot{=} y, y \dot{=} (u * v)^{-1}$	REZOLVĂ
$x \dot{=} y$	$y \dot{=} (u * v)^{-1}$	REZOLVĂ
$y \dot{=} (u * v)^{-1}, x \dot{=} (u * v)^{-1}$		REZOLVĂ

- $\nu = \{y/(u * v)^{-1}, y/(u * v)^{-1}\}$ este cgu.

□

2 Deductie naturala

Sistemul de reguli al deductiei naturale

$$\begin{array}{c}
 \frac{\varphi \quad \psi}{\varphi \wedge \psi} (\wedge i) \\
 \\
 \frac{\boxed{\begin{array}{c} \varphi \\ \vdots \\ \psi \end{array}}}{\varphi \rightarrow \psi} (\rightarrow i) \\
 \\
 \frac{\varphi}{\varphi \vee \psi} (v i_1) \quad \frac{\psi}{\varphi \vee \psi} (v i_2) \\
 \\
 \frac{\boxed{\begin{array}{c} \varphi \\ \vdots \\ \perp \end{array}}}{\neg \varphi} (\neg i) \\
 \\
 \frac{\varphi}{\neg \neg \varphi} (\neg \neg i) \\
 \\
 \frac{}{\varphi \vee \neg \varphi} \text{ TND}
 \end{array}
 \quad
 \begin{array}{c}
 \frac{\varphi \wedge \psi}{\varphi} (\wedge e_1) \quad \frac{\varphi \wedge \psi}{\psi} (\wedge e_2) \\
 \\
 \frac{\varphi \quad \varphi \rightarrow \psi}{\psi} (\rightarrow e) \\
 \\
 \frac{\varphi \vee \psi \quad \boxed{\begin{array}{c} \varphi \\ \vdots \\ \chi \end{array}} \quad \boxed{\begin{array}{c} \psi \\ \vdots \\ \chi \end{array}}}{\chi} (v e) \\
 \\
 \frac{\varphi \quad \neg \varphi}{\perp} (\neg e) \\
 \\
 \frac{\neg \neg \varphi}{\varphi} (\neg \neg e) \\
 \\
 \frac{\perp}{\varphi} (\perp e)
 \end{array}$$

TND (*tertium non datur*) este regulă derivată.

Atenție! La acest sistem se adaugă regula de copiere.

Regula de copiere:

- la un pas al unei demonstrații poate fi copiată orice formulă demonstrată anterior.
- la un pas al unei demonstrații nu pot fi copiate formule din interiorul cutiilor care sunt închise în acel moment.

Exemplu în curs sau Huth și Ryan, pg. 20, unde cartea Huth și Ryan: <http://staff.ustc.edu.cn/~huangwc/book/LogicInCS.pdf>

(Exercițiul 2) Demonstrați că următorii secvenți sunt valizi:

- (1) $(p \wedge q) \wedge r, s \wedge t \vdash q \wedge s$
- (2) $p, \neg\neg(q \wedge r) \vdash \neg\neg p \wedge r$
- (3) $p \wedge q \rightarrow r \vdash p \rightarrow (q \rightarrow r)$
- (4) $p \wedge (q \vee r) \vdash (p \wedge q) \vee (p \wedge r)$
- (5) $p \rightarrow q, p \rightarrow \neg q \vdash \neg p$

Demonstrație: (2) Huth și Ryan, pg 8; (3) Huth și Ryan, pg 15;

(4) Huth și Ryan Example 1.18, pg 19; (5) Huth și Ryan, exemple 1.21, pg 22 □

(Exercițiul 3) Demonstrați că următoarele reguli pot fi derivate din regulile deducției naturale:

$$\frac{\varphi \rightarrow \psi \quad \neg\psi}{\neg\varphi} \text{ MT} \qquad \frac{\begin{array}{c} \boxed{\neg\varphi} \\ \vdots \\ \perp \end{array}}{\varphi} \text{ RAA}$$

MT = *modus tollens* RAA = *reductio ad absurdum*

Demonstrație: Huth și Ryan, secțiunea 1.2.2 □

(Exercițiul 4) Fie $n \geq 1$ și $\varphi_1, \dots, \varphi_n, \varphi$ formule. Demonstrați că

dacă $\vdash \varphi_1 \rightarrow (\varphi_2 \rightarrow (\dots \rightarrow (\varphi_n \rightarrow \varphi) \dots))$ este valid, atunci $\varphi_1, \dots, \varphi_n \vdash \varphi$ este valid.

Demonstrație: Huth și Ryan, pagina 53, Step 3 □

(Exercițiul 5) Știm că *echivalența logică* este definită astfel: $\varphi \leftrightarrow \psi = (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$. Găsiți reguli de introducere și eliminare pentru \leftrightarrow .

Demonstrație: Observăm că \leftrightarrow este o combinație între \rightarrow și \wedge . Regulile pentru \leftrightarrow se obțin combinând regulile pentru \rightarrow și \wedge .

Introducerea ($\leftrightarrow i$): pentru a introduce $\varphi \leftrightarrow \psi$ trebuie să introducem $\varphi \rightarrow \psi$ și $\psi \rightarrow \varphi$, apoi să introducem \wedge ; în consecință regula va arăta așa

$$\frac{\begin{array}{|c|} \hline \varphi \\ \hline \vdots \\ \hline \psi \\ \hline \end{array} \quad \begin{array}{|c|} \hline \psi \\ \hline \vdots \\ \hline \varphi \\ \hline \end{array}}{\varphi \leftrightarrow \psi} (\leftrightarrow i)$$

Eliminarea ($\leftrightarrow e$): pentru a elimina $\varphi \leftrightarrow \psi$ trebuie să eliminăm \wedge apoi să eliminăm o \rightarrow ; vom avea două variante:

$$\frac{\varphi \leftrightarrow \psi \quad \psi}{\varphi} (\leftrightarrow e_1) \quad \frac{\varphi \leftrightarrow \psi \quad \varphi}{\psi} (\leftrightarrow e_2)$$

□

(Exercițiul 6) Formalizați și demonstrați folosind deducția naturală faptul că din ipotezele (i1)-(i5) deducem (c):

- (i1) *Toți scriitorii care înțeleg natura umană sunt înțelepți.*
- (i2) *Un scriitor care este poet adevărat poate trezi sentimente puternice.*
- (i3) *Eminescu este scriitorul care a scris "Luceafărul".*
- (i4) *Un poet care trezește sentimente puternice înțelege natura umană.*
- (i5) *Numai un poet adevărat putea scrie "Luceafărul".*

(c) *Eminescu este înțelept.*

Demonstrație: Formalizarea:

- $S \wedge NU \rightarrow I$
- $S \wedge P \rightarrow SP$
- $E \rightarrow S \wedge L$
- $P \wedge SP \rightarrow NU$
- $L \rightarrow P$

□

3 Puncte fixe

Teorie

O mulțime parțial ordonată (mpo) este o pereche (M, \leq) unde $\leq \subseteq M \times M$ este o relație de ordine (i.e., reflexivă, antisimetrică, tranzitivă). O mulțime parțial ordonată (C, \leq) este *completă* (cpo) dacă C are prim element \perp ($\perp \leq x$ oricare $x \in C$) și $\bigvee_n x_n$ există în C pentru orice lanț $x_1 \leq x_2 \leq x_3 \leq \dots$

Fie (C, \leq_C) o mulțime parțial ordonată. Un element $a \in C$ este *punct fix* al unei funcții $f : C \rightarrow C$ dacă $f(a) = a$. Un element $lfp \in C$ este *cel mai mic punct fix* al unei funcții $f : C \rightarrow C$ dacă este punct fix și pentru orice alt punct fix $a \in C$ al lui f avem $lfp \leq_C a$.

(Exercițiul 7) Care sunt punctele fixe ale următoarelor funcții? Indicați cel mai mic punct fix.

- 1) $f_1 : \mathcal{P}(\{1, 2, 3\}) \rightarrow \mathcal{P}(\{1, 2, 3\})$, $f_1(Y) = Y \cup \{1\}$.
- 2) $f_2 : \mathcal{P}(\{1, 2, 3\}) \rightarrow \mathcal{P}(\{1, 2, 3\})$, $f_2(Y) = \begin{cases} \{1\} & \text{dacă } 1 \in Y \\ \emptyset & \text{altfel} \end{cases}$.
- 3) $f_3 : \mathcal{P}(\{1, 2, 3\}) \rightarrow \mathcal{P}(\{1, 2, 3\})$, $f_3(Y) = \begin{cases} \emptyset & \text{dacă } 1 \in Y \\ \{1\} & \text{altfel} \end{cases}$.

Demonstrație:

- 1) Se observă că punctele fixe ale lui f_1 sunt submulțimile Y ale lui $\{1, 2, 3\}$ care îl conțin pe 1 (dacă $1 \notin Y$, atunci $f_1(Y) = Y \cup \{1\}$ și evident $Y \neq Y \cup \{1\}$). Deci punctele fixe ale lui f_1 sunt $\{1\}, \{1, 2\}, \{1, 3\}, \{1, 2, 3\}$. Evident, cel mai mic punct fix este $\{1\}$.
- 2) Se observă că singurele puncte fixe ale lui f_2 sunt \emptyset și $\{1\}$. Evident \emptyset este cel mai mic punct fix.
- 3) Se observă că f_3 nu are puncte fixe.

□

Teorie

Fie (A, \leq_A) și (B, \leq_B) mulțimi parțial ordonate. O funcție $f : A \rightarrow B$ este *monotonă* (crescătoare) dacă $a_1 \leq_A a_2$ implică $f(a_1) \leq_B f(a_2)$ oricare $a_1, a_2 \in A$.

O *clauză definită propozițională* este o formulă care poate avea una din formele:

- q (clauză unitate)
- $p_1 \wedge \dots \wedge p_k \rightarrow q$

unde q, p_1, \dots, p_n sunt variabile propoziționale.

Fie S o mulțime de clauze definite propoziționale. Fie A mulțimea variabilelor propoziționale p_1, p_2, \dots care apar în S și $Baza = \{p_i \mid p_i \in S\}$ mulțimea clauzelor unitate din S . Definim funcția $f_S : \mathcal{P}(A) \rightarrow \mathcal{P}(A)$ prin

$$f_S(Y) = Y \cup Baza \cup \{a \in A \mid (s_1 \wedge \dots \wedge s_n \rightarrow a) \text{ este în } S, s_1 \in Y, \dots, s_n \in Y\}$$

(Exercițiul 8) Arătați că funcția f_S este monotonă.

Demonstrație: Fie $Y_1, Y_2 \subseteq A$ astfel încât $Y_1 \subseteq Y_2$. Trebuie să arătăm că $f_S(Y_1) \subseteq f_S(Y_2)$. Fie următoarele mulțimi:

$$\begin{aligned} Z_1 &= \{a \in A \mid (s_1 \wedge \dots \wedge s_n \rightarrow a) \text{ este în } S, s_1 \in Y_1, \dots, s_n \in Y_1\}, \\ Z_2 &= \{a \in A \mid (s_1 \wedge \dots \wedge s_n \rightarrow a) \text{ este în } S, s_1 \in Y_2, \dots, s_n \in Y_2\}. \end{aligned}$$

Deci $f_S(Y_1) = Y_1 \cup Baza \cup Z_1$ și $f_S(Y_2) = Y_2 \cup Baza \cup Z_2$. Cum $Y_1 \subseteq Y_2$, rămâne să arătăm doar că $Z_1 \subseteq Z_2$. Fie $a \in Z_1$. Atunci există $s_1 \wedge \dots \wedge s_n \rightarrow a \in S$ și $s_1, \dots, s_n \in Y_1$. Deci $s_1, \dots, s_n \in Y_2$, de unde rezultă că $a \in Z_2$. □

Teorie

Fie (A, \leq_A) și (B, \leq_B) mulțimi parțial ordonate complete. O funcție $f : A \rightarrow B$ este *continuă* dacă $f(\bigvee_n a_n) = \bigvee_n f(a_n)$ pentru orice lanț $\{a_n\}_n$ din A . Observăm că orice funcție continuă este crescătoare.

Pentru orice mulțime de clauze definite propoziționale S , funcția f_S este continuă.

Teorema 1 (Knaster-Tarski). Fie (C, \leq) o mulțime parțial ordonată completă și $\mathbf{F} : C \rightarrow C$ o funcție continuă. Atunci

$$a = \bigvee_n \mathbf{F}^n(\perp)$$

este cel mai mic punct fix al funcției \mathbf{F} .

(Exercițiul 9) Calculați cel mai mic punct fix pentru funcția f_{S_i} , $i \in \{1, 2, 3\}$, pentru următoarele mulțimi de clauze definite propoziționale:

- 1) $S_1 = \{x_1 \wedge x_2 \rightarrow x_3, x_4 \wedge x_2 \rightarrow x_5, x_2, x_6, x_6 \rightarrow x_1\}$.
- 2) $S_2 = \{x_1 \wedge x_2 \rightarrow x_3, x_4 \rightarrow x_1, x_5 \rightarrow x_2, x_2 \rightarrow x_5, x_4\}$.
- 3) $S_3 = \{x_1 \rightarrow x_2, x_1 \wedge x_3 \rightarrow x_1, x_3\}$.

Demonstrație:

- 1) Observăm că $A = \{x_1, x_2, \dots, x_6\}$ și $Baza = \{x_2, x_6\}$. Cum f_S este continuă, aplicăm Teorema Knaster-Tarski pentru a calcula cel mai mic punct fix:

$$\begin{aligned} f_{S_1}(\emptyset) &= Baza = \{x_2, x_6\} \\ f_{S_1}(\{x_2, x_6\}) &= \{x_2, x_6, x_1\} \\ f_{S_1}(\{x_2, x_6, x_1\}) &= \{x_2, x_6, x_1, x_3\} \\ f_{S_1}(\{x_2, x_6, x_1, x_3\}) &= \{x_2, x_6, x_1, x_3\} \end{aligned}$$

În concluzie, cel mai mic punct fix căutam este $\{x_2, x_6, x_1, x_3\}$.

- 2) Observăm că $A = \{x_1, x_2, \dots, x_5\}$ și $Baza = \{x_4\}$. Cum f_S este continuă, aplicăm Teorema Knaster-Tarski pentru a calcula cel mai mic punct fix:

$$\begin{aligned} f_{S_2}(\emptyset) &= Baza = \{x_4\} \\ f_{S_2}(\{x_4\}) &= \{x_4, x_1\} \\ f_{S_2}(\{x_4, x_1\}) &= \{x_4, x_1\} \end{aligned}$$

În concluzie, cel mai mic punct fix căutam este $\{x_4, x_1\}$.

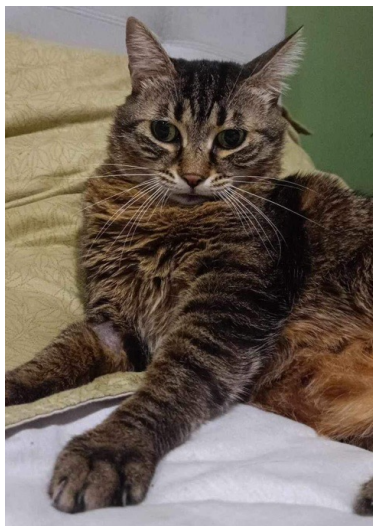
- 3) Observăm că $A = \{x_1, x_2, x_3\}$ și $Baza = \{x_3\}$. Cum f_S este continuă, aplicăm Teorema Knaster-Tarski pentru a calcula cel mai mic punct fix:

$$\begin{aligned} f_{S_3}(\emptyset) &= Baza = \{x_3\} \\ f_{S_3}(\{x_3\}) &= \{x_3\} \end{aligned}$$

În concluzie, cel mai mic punct fix căutam este $\{x_3\}$.

□

4 Rezoluție SLD



Arata mai rau decat e de fapt, dar pana si Grasa e putin sperziata.

Teorie

- O *clauză definită* este o formulă de forma:
 - $P(t_1, \dots, t_n)$ (formulă atomică), unde P este un simbol de predicat, iar t_1, \dots, t_n termeni
 - $P_1 \wedge \dots \wedge P_n \rightarrow Q$, unde toate P_i, Q sunt formule atomice.
- O regulă din Prolog $Q : - P_1, \dots, P_n$ este o clauză $P_1 \wedge \dots \wedge P_n \rightarrow Q$, iar un fapt din Prolog $P(t_1, \dots, t_n)$ este o formulă atomică $P(t_1, \dots, t_n)$.
- O clauză definită $P_1 \wedge \dots \wedge P_n \rightarrow Q$ poate fi gândită ca formula $Q \vee \neg P_1 \vee \dots \vee \neg P_n$.
- Pentru o mulțime de clauze definite T , *regula rezoluției SLD* este

$$\text{SLD} \quad \frac{\neg P_1 \vee \dots \vee \neg P_i \vee \dots \vee \neg P_n}{(\neg P_1 \vee \dots \vee \neg Q_1 \vee \dots \vee \neg Q_m \vee \dots \vee \neg P_n)\theta}$$

unde $Q \vee \neg Q_1 \vee \dots \vee \neg Q_m$ este o clauză definită din T (în care toate variabilele au fost redenumite) și θ este c.g.u pentru P_i și Q .

- Fie T o mulțime de clauze definite și $P_1 \wedge \dots \wedge P_m$ o țintă, unde P_i sunt formule atomice. O derivare din T prin rezoluție SLD este o secvență $G_0 := \neg P_1 \vee \dots \vee \neg P_m, G_1, \dots, G_k, \dots$ în care G_{i+1} se obține din G_i prin regula SLD. Dacă există un k cu $G_k = \square$ (clauza vidă), atunci derivarea se numește *SLD-respingere*.

Teorema 2 (Completitudinea SLD-rezoluției). *Sunt echivalente:*

- există o SLD-respingere a lui $P_1 \wedge \dots \wedge P_m$ din T ,*
- $T \models P_1 \wedge \dots \wedge P_m$.

(Exercițiul 10) Găsiți o SLD-respingere pentru următoarele programe Prolog și ținte:

- $r :- p, q.$
 - $s :- p, q.$
 - $v :- t, u.$
 - $w :- v, s.$
 - $t.$
 - $q.$
 - $u.$
 - $p.$
 - $q(X, Y) :- q(Y, X), q(Y, f(f(Y))).$
 - $q(a, f(f(X))).$
 - $p(X) :- q(X, f(Y)), r(a).$
 - $p(X) :- r(X).$
 - $q(X, Y) :- p(Y).$
 - $r(X) :- q(X, Y).$
 - $r(f(b)).$
- ?- w.
?- q(f(Z), a).
?- p(X), q(Y, Z).

Demonstrație:

(a)

$$\begin{aligned}
G_0 &= \neg w && \\
G_1 &= \neg v \vee \neg s && (4) \\
G_2 &= \neg t \vee \neg u \vee \neg s && (3) \\
G_3 &= \neg u \vee \neg s && (5) \\
G_4 &= \neg s && (7) \\
G_5 &= \neg p \vee \neg q && (2) \\
G_6 &= \neg q && (8) \\
G_7 &= \square && (6)
\end{aligned}$$

(b)

$$\begin{aligned}
G_0 &= \neg q(f(Z), a) \\
G_1 &= \neg q(a, f(Z)) \vee \neg q(a, f(f(a))) && (1 \text{ cu } \theta(X) = f(Z) \text{ și } \theta(Y) = a) \\
G_2 &= \neg q(a, f(Z)) && (2 \text{ cu } \theta(X) = a) \\
G_3 &= \square && (2 \text{ cu } \theta(Z) = f(X))
\end{aligned}$$

(c)

$$\begin{aligned}
G_0 &= \neg p(X) \vee \neg q(Y, Z) \\
G_1 &= \neg r(X_1) \vee \neg q(Y, Z) && (2 \text{ cu } \theta(X) = X_1) \\
G_2 &= \neg q(Y, Z) && (5 \text{ cu } \theta(X_1) = f(b)) \\
G_3 &= \neg p(Z_1) && (3 \text{ cu } \theta(X) = Y_1 \text{ și } \theta(Y) = Z_1) \\
G_4 &= \neg r(X) && (2 \text{ cu } \theta(Z_1) = X) \\
G_5 &= \square && (5 \text{ cu } \theta(X) = f(b))
\end{aligned}$$

□

Teorie

Fie T o mulțime de clauze definite și o țintă $G_0 = \neg P_1 \vee \dots \vee \neg P_m$. Un *arbore SLD* este definit astfel:

- Fiecare nod al arborelui este o țintă (posibil vidă)
- Rădăcina este G_0
- Dacă arborele are un nod G_i , iar G_{i+1} se obține din G_i folosind regula SLD folosind o clauză $C_i \in T$, atunci nodul G_i are copilul G_{i+1} . Muchia dintre G_i și G_{i+1} este etichetată cu C_i .

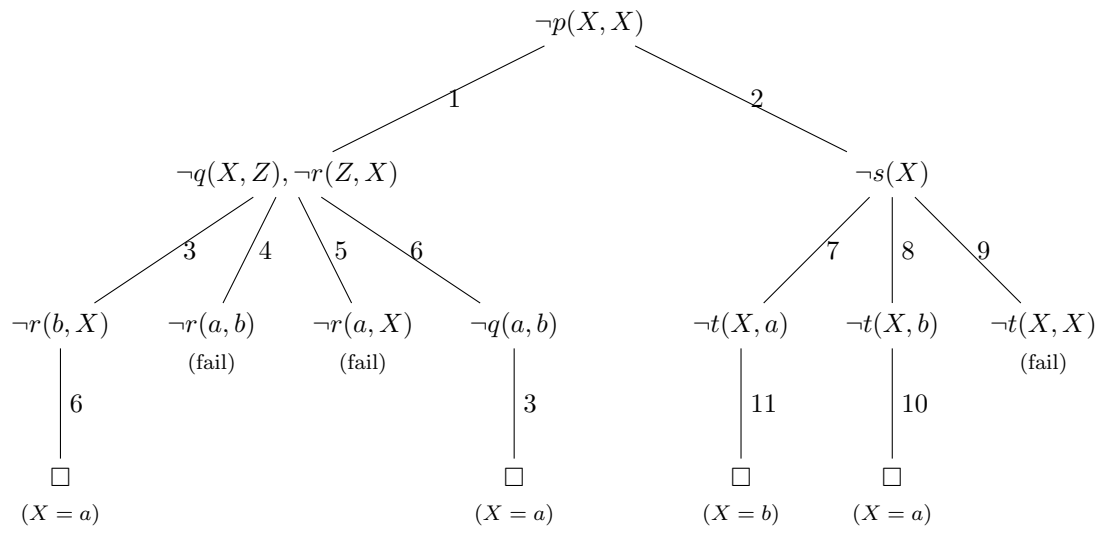
Dacă un arbore SLD cu rădăcina G_0 are o frunză \square (clauza vidă), atunci există o SLD-respingere a lui G_0 din T .

(Exercițiul 11) Desenați arborele SLD pentru programul Prolog de mai jos și ținta $?- p(X, X)$.

- | | |
|-----------------------------------|-----------------------|
| 1. $p(X, Y) :- q(X, Z), r(Z, Y).$ | 7. $s(X) :- t(X, a).$ |
| 2. $p(X, X) :- s(X).$ | 8. $s(X) :- t(X, b).$ |
| 3. $q(X, b).$ | 9. $s(X) :- t(X, X).$ |
| 4. $q(b, a).$ | 10. $t(a, b).$ |
| 5. $q(X, a) :- r(a, X).$ | 11. $t(b, a).$ |
| 6. $r(b, a).$ | |

Demonstrație:

□



5 Lambda-calcul

Teorie

Lambda-calculul (sau λ -calculul) a fost dezvoltat între 1929-1932 de Alonzo Church și a fost propus ca sistem formal pentru logica matematică. În 1935, a demonstrat că orice funcție calculabilă peste numerele naturale, poate fi calculată în λ -calcul. Tot în 1935, independent de Church, Alan Turing a dezvoltat mecanismul numit astăzi mașina Turing, iar în 1936 a argumentat și el că orice funcție calculabilă peste numerele naturale poate fi calculată de o mașină Turing și, în plus, a arătat echivalența celor două modele de calcul (λ -calcul și mașini Turing), ducând la ceea ce numim astăzi *Teza Church-Turing*.

Sintaxa λ -calculului:

$$t = x \mid \lambda x.t \mid t t$$

λ -termeni. Fie $Var = \{x, y, z, \dots\}$ o mulțime infinită de variabile. Mulțimea λ -termenilor ΛT este definită inductiv, astfel:

[Variabilă] $Var \subseteq \Lambda T$

[Aplicare] dacă $t_1, t_2 \in \Lambda T$ atunci $(t_1 t_2) \in \Lambda T$

[Abstractizare] dacă $x \in Var$ și $t \in \Lambda T$, atunci $(\lambda x.t) \in \Lambda T$

Convenții de scriere.

- în scrierea λ -termenilor vom elimina parantezele exterioare și vom scrie, de exemplu, xy în loc de (xy) , sau $\lambda x.xy$ în loc de $(\lambda x.(xy))$;
- aplicarea este asociativă la stânga: $t_1 t_2 t_3$ este $(t_1 t_2) t_3$;
- corpul abstractizării este extins la dreapta: $\lambda x.t_1 t_2$ este $\lambda x.(t_1 t_2)$;
- scriem $\lambda xyz.t$, în loc de $\lambda x.\lambda y.\lambda z.t$.

5.1 Variabile libere și legate

Variabile libere și legate. Pentru un termen $\lambda x.t$ spunem că:

- aparițiile variabilei x în t sunt legate (*bound*);
- λx este legătura (*binder*), iar t este domeniul (*scope*) legării;
- o apariție a unei variabile este liberă (*free*) dacă apare într-o poziție în care nu este legată.

Un termen fără variabile libere se numește **închis** (*closed*).

Mulțimea variabilelor libere $FV(t)$. Pentru un λ -termen t , mulțimea variabilelor libere este definită astfel:

[Variabilă] $FV(x) = \{x\}$

[Aplicare] $FV(t_1 t_2) = FV(t_1) \cup FV(t_2)$

[Abstractizare] $FV(\lambda x.t) = FV(t) - \{x\}$

Exercițiul 1. Calculați mulțimea variabilelor libere pentru următorii λ -termeni:

a. $\lambda x.xy$

c. $x(\lambda xy.xyz)(\lambda v.yv)$

b. $x\lambda x.xy$

d. $\lambda t.((\lambda xyz.yzx)t)$

5.2 Substituții

Fie t un λ -termen și $x \in Var$. Pentru un λ -termen u vom nota prin $[u/x]t$ rezultatul înlocuirii tuturor aparițiilor libere ale lui x cu u în t .

$$[\text{Variabilă}] [u/x]x = u$$

$$[\text{Variabilă}] [u/x]y = y \text{ dacă } x \neq y$$

$$[\text{Aplicare}] [u/x](t_1 t_2) = [u/x]t_1 [u/x]t_2$$

$$[\text{Abstractizare}] [u/x]\lambda y.t = \lambda y.[u/x]t \text{ unde } x \neq y \text{ și } y \notin FV(u)$$

Variabilele legate pot fi redenumite.

Exercițiul 2. *Aplicați substituțiile indicate în următorii λ -termeni:*

$$a. [y/x]\lambda z.x$$

$$c. [\lambda z.z/x](\lambda x.yx)$$

$$b. [y/x]\lambda y.x$$

$$d. [\lambda z.z/x](\lambda y.yx)$$

5.3 α -conversie (α -echivalență)

α -conversia $=_\alpha$ este relația binară care satisface următoarele proprietăți:

$$[\text{Reflexivitate}] t =_\alpha t$$

$$[\text{Simetrie}] t_1 =_\alpha t_2 \text{ implică } t_2 =_\alpha t_1$$

$$[\text{Tranzitivitate}] t_1 =_\alpha t_2 \text{ și } t_2 =_\alpha t_3 \text{ implică } t_1 =_\alpha t_3$$

$$[\text{Redenumire}] \lambda x.t =_\alpha \lambda y.[y/x]t \text{ dacă } y \notin FV(t)$$

$$[\text{Compatibilitate}] t_1 =_\alpha t_2 \text{ implică } tt_1 =_\alpha tt_2, t_1 t =_\alpha t_2 t \text{ și } \lambda x.t_1 =_\alpha \lambda x.t_2$$

Compatibilitatea cu substituția.

$$t_1 =_\alpha t_2 \text{ și } u_1 =_\alpha u_2 \text{ implică } [u_1/x]t_1 =_\alpha [u_2/x]t_2$$

Exercițiul 3. *Verificați care dintre α -conversiile următoare sunt adevărate:*

$$a. \lambda x.x =_\alpha \lambda y.y$$

$$d. \lambda x.xy =_\alpha \lambda y.yy$$

$$b. \lambda x.y =_\alpha \lambda y.x$$

$$e. x\lambda x.xy =_\alpha x\lambda z.zy$$

$$c. \lambda x.xy =_\alpha \lambda x.xz$$

$$f. x\lambda x.xy =_\alpha y\lambda x.xy$$

5.4 β -reducția

β -reducția este o relație definită pe mulțimea α -termenilor, $\beta \subseteq \Lambda T \times \Lambda T$, unde

$$[\text{Aplicarea}] (\lambda x.y)u \rightarrow_\beta [u/x]t$$

$$[\text{Compatibilitatea}] t_1 \rightarrow_\beta t_2 \text{ implică } tt_1 \rightarrow_\beta tt_2, t_1 t \rightarrow_\beta t_2 t \text{ și } \lambda x.t_1 \rightarrow_\beta \lambda x.t_2$$

Închiderea reflexivă și tranzitivă a acestei relații se notează $\rightarrow_\beta^* \subseteq \Lambda T \times \Lambda T$, iar $t_1 \rightarrow_\beta^* t_2$ dacă există $n \geq 0$ și u_0, \dots, u_n astfel încât $t_1 =_\alpha u_0 \rightarrow_\beta u_1 \rightarrow_\beta \dots \rightarrow_\beta u_n =_\alpha t_2$.

Exercițiul 4. *Să se aplice două β -reducții succesive asupra λ -termenului $(\lambda x.(\lambda y.yx)z)v$. Este o singură variantă de aplicare?*

Un termen poate fi β -redus în mai multe moduri, iar proprietatea de confluență ne asigură că vom ajunge întotdeauna la același rezultat. (forma normală este unică, modulo α -echivalență)

5.5 Numeralii Church

Fie numerele naturale definite prin:

- $0 := \lambda f x. x$
- $1 := \lambda f x. f x$
- $2 := \lambda f x. f(f x)$
- ...
- $m := \lambda f x. f^m x$
- ...

Fie functiile

- **succ** $:= \lambda n f x. f(n f x)$
- **add** $:= \lambda m n f x. m f(n f x)$

Sa se arate ca

- **succ** $2 =_{\alpha} 3$

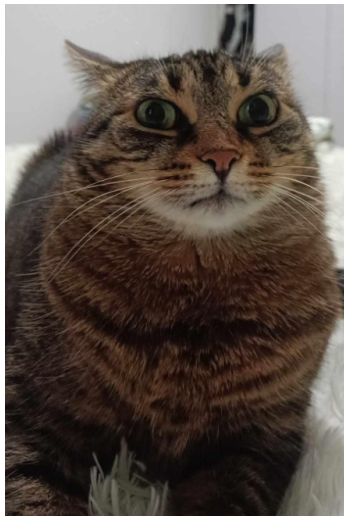
Demonstratie.

$$\begin{aligned}
 \mathbf{succ\ 2} &:= (\lambda n f x. f(n f x))(\lambda f' x'. f'(f' x')) \rightarrow_{\beta} \\
 &\rightarrow_{\beta} [(\lambda f' x'. f'(f' x'))/n] \lambda f x. f(n f x) =_{\alpha} \\
 &=_{\alpha} \lambda f x. f((\lambda f' x'. f'(f' x')) f x) \rightarrow_{\beta}^* \\
 &\rightarrow_{\beta}^* \lambda f x. f(f(f x)) =: 3
 \end{aligned}$$

- **add** $3\ 1 =_{\alpha} 4$

Demonstratie.

$$\begin{aligned}
 \mathbf{add\ 3\ 1} &:= (\lambda m n f x. m f(n f x))(\lambda f' x'. f'(f'(f' x')))(\lambda f'' x''. f'' x'') \rightarrow_{\beta}^* \\
 &\rightarrow_{\beta}^* [(\lambda f' x'. f'(f'(f' x')))/m][\lambda f'' x''. f'' x'']/n] (\lambda f x. m f(n f x)) =_{\alpha} \\
 &=_{\alpha} \lambda f x. (\lambda f' x'. f'(f'(f' x')) f)((\lambda f'' x''. f'' x'') f x) \rightarrow_{\beta}^* \\
 &\rightarrow_{\beta}^* \lambda f x. (\lambda f' x'. f'(f'(f' x')) f(f x)) =_{\alpha} \\
 &=_{\alpha} \lambda f x. f(f(f(f x))) =: 4
 \end{aligned}$$



Si Grasa e putin confuza, nu sunteti singurii!