Logica cu predicate de ordinul I
Curs 3 - Rezoluție

Ștefan Ciobâcă

12 Decembrie 2016

Note organizatorice

1. Cursurile din partea a II-a:
   `http://profs.info.uaic.ro/~stefan.ciobaca/logica/`
   Verificați periodic această pagină.
   Raportare eventuale greșeli: `stefan.ciobaca@gmail.com`

Reminder - Sintaxa

O nouă logică, mai expresivă decât LP.
Mulțimea termenilor ($\mathcal{T}$):

$$t ::= x \mid c \mid f(\underbrace{t, \ldots, t}_{n}) \qquad x \in \mathcal{X}, c \in \mathcal{F}_0, f \in \mathcal{F}_n$$

Mulțimea formulelor atomice (At):

$$a ::= P \mid Q(\underbrace{t, \ldots, t}_{n}) \qquad P \in \mathcal{P}_0, Q \in \mathcal{P}_n$$

Mulțimea formulelor de ordinul I (LP1):

$$\begin{aligned} F ::= \quad & a \mid (\neg F) \mid (F) \mid (F \vee F) \mid (F \wedge F) \mid \\ & (F \rightarrow F) \mid (\forall x.F) \mid (\exists x.F) \qquad a \in \mathtt{At}, x \in \mathcal{X} \end{aligned}$$

Reminder - Sintaxa - Exemple

Exemplu

$$P \in \mathit{LP1}$$

$$Q(x) \in \mathit{LP1} \qquad R\Big(h(x), f(x, y)\Big) \in \mathit{LP1} \qquad (\neg Q(f(x, y))) \in \mathit{LP1}$$

$$(P \wedge Q(x)) \in \mathit{LP1} \qquad \Bigg(Q(x) \vee R\Big(h(x), f(x, y)\Big)\Bigg) \in \mathit{LP1}$$

$$(Q(x) \rightarrow R(x, y)) \in \mathit{LP1} \qquad ((Q(x) \wedge P) \vee Q(y)) \in \mathit{LP1}$$

$$(\forall x.(Q(x) \vee P)) \in \mathit{LP1} \qquad ((\exists x.Q(x)) \vee (\neg P)) \in \mathit{LP1}$$

Reminder - Semantica LP1 - Noțiunea de structură

Definiție (Structură)

O structură este o pereche $S = (U, I)$, formată din:

1. *o mulțime nevidă U, numită univers (sau domeniu) al structurii;*
2. *o funcție I, care asociază:*
   - 2.1 *fiecărui simbol constant $c \in \mathcal{F}_0$ din signatură un element al universului $I_c \in U$;*
   - 2.2 *fiecărui simbol funcțional $f \in \mathcal{F}_n$ de aritate n o funcție n-ară peste univers: $I_f : U \times \ldots \times U \to U$;*
   - 2.3 *fiecărui simbol predicativ $P \in \mathcal{P}_0$ de aritate 0 o valoare de adevăr $I_P \in \mathbb{B}$;*
   - 2.4 *fiecărui simbol predicativ $P \in \mathcal{F}_n$ de aritate n un predicat n-ar peste univers: $I_P : U \times \ldots \times U \to \mathbb{B}$;*
   - 2.5 *fiecărei variabile $x \in \mathcal{X}$ un element al universului: $I_x \in U$.*

Reminder - Semantica LP1 - Exemplu de structură

Fie $\mathcal{F}_0 = \{e\}$, $\mathcal{F}_1 = \{i\}$, $\mathcal{F}_2 = \{f\}$ și $\mathcal{P}_2 = \{equals\}$.
Vom considera structura $S = (U, I)$, unde $U = \mathbb{Z}$ și:

1. $I_e : \mathbb{Z}^0 \to \mathbb{Z}$, definită prin
$$I_e = 0;$$
2. $I_i : \mathbb{Z}^1 \to \mathbb{Z}$, definită prin
$$I_i(u) = -u,$$
pentru orice $u \in \mathbb{Z}$;
3. $I_f : \mathbb{Z}^2 \to \mathbb{Z}$, definită prin
$$I_f(u, v) = u + v,$$
pentru orice $u, v \in \mathbb{Z}$;
4. $I_{equals} : \mathbb{Z}^2 \to \mathbb{B}$, definit prin
$$I_{equals}(u, v) = \begin{cases} 1 & \text{dacă } u = v \\ 0 & \text{dacă } u \neq v; \end{cases}$$
5. $I_x \in \mathbb{Z}$, definită prin $I_x = 7$, pentru orice variabilă $x \in \mathcal{X}$.

Reminder - Interpretarea termenilor/formulelor într-o structură

Exemplu

1. $S(f(f(x,e),i(x))) = 0;$
2. $S\Big(equals(x,y)\Big) = 1;$
3. $S\Big(equals(x,e)\Big) = 0;$
4. $S[x \mapsto 0]\Big(equals(x,e)\Big) = 1;$
5. $S\Big(\exists x.(equals(x,e))\Big) = 1;$
6. $S\Big(\forall x.(equals(x,e))\Big) = 0;$

Reminder - Forme normale Skolem

O formulă F este în *formă normală Skolem* dacă există variabilele distincte $\{x_1, \ldots, x_n\}$ astfel încât:

$$F = \forall x_1.\forall x_2.\ldots.\forall x_n.G$$

pentru o formulă $G \in \mathtt{LP1}$ astfel încât $\mathit{free}(G) = \{x_1, \ldots, x_n\}$ și $\mathit{bound}(G) = \emptyset$.

Exemplu

Formule în FNS:

1. $\forall x.\forall y.(P(x, y))$
2. $\forall x.\forall y.(P(x, y) \land Q(x))$
3. $\forall x.\forall y.\forall z.((P(x, y) \land Q(x)) \lor R(x, y, z))$
4. $P(c) \land Q(f(a, b))$

Formula F este în FNSC dacă, în plus, G este în FNC (conjuncție de clauze).

Reminder - Aducerea unei formule în FNSC

Pentru orice formulă F, există o formulă F_{FNSC}, slab echivalentă cu F, aflată în FNSC.

F	orice formulă $\in$ LP1
$\equiv$	
F_p	o formulă în FN prenex
$\equiv_s$	
F_s	o formulă în FN Skolem
$\equiv$	
F_{FNSC}	o formulă în FN Skolem clauzală

Planul cursului

1. Domeniul *Hebrand*
2. Rezoluția de bază în `LP1`
3. Unificare
4. Rezoluția "pură"în `LP1`

Domeniul *Herbrand*

Definiție

Domeniul Herbrand asociat mulțimii $\mathcal{F}$ este mulțimea de termeni notată cu $\mathcal{D}(\mathcal{F})$ și definită inductiv astfel:

1. *dacă $\mathcal{F}_0 \neq \emptyset$, atunci $\mathcal{F}_0 \subseteq \mathcal{D}(\mathcal{F})$;*
2. *dacă $\mathcal{F}_0 = \emptyset$, atunci $c \in \mathcal{D}(\mathcal{F})$ (c este un simbol constant pe care îl adăugăm "forțat");*
3. *dacă $t_1, \ldots, t_n \in \mathcal{D}(\mathcal{F})$ și $f \in \mathcal{F}_n$, atunci $f(t_1, \ldots, t_n) \in \mathcal{D}(\mathcal{F})$.*

Definiție

Termen de bază (engl. ground term) = termen fără variabile.
Domeniu Hebrand = mulțimea termenilor de bază.

Structuri Herbrand

Definiție

O structură $S = (U, I)$ se numește structură Herbrand dacă:

1. $U = \mathcal{D}(\mathcal{F})$; *($\mathcal{D}(\mathcal{F})$ = domeniul Herbrand)*
2. $S(t) = t$ *pentru orice termen* $t \in \mathcal{T} \cap \mathcal{D}(\mathcal{F})$.

Motivație:

Teoremă (Herbrand)

O formulă F, aflată în FNS, este satisfiabilă dacă și numai dacă F admite model Herbrand.

Reprezentarea ca mulțimi

Fie $F = \forall x_1. \ldots . \forall x_n.(C_1 \wedge \ldots \wedge C_m)$ o formulă în FNSC. Scriem și

$$F = \{C_1, \ldots, C_m\}.$$

Dacă $C_1 = L_1^1 \vee \ldots \vee L_{k_1}^1$, $C_2 = L_1^2 \vee \ldots \vee L_{k_2}^2$, ..., $C_n = L_1^m \vee \ldots \vee L_{k_m}^m$, scriem și

$$C_i = \{L_i^i, \ldots, L_{k_i}^i\}$$

și respectiv:

$$F = \{\{L_1^1, \ldots, L_{k_1}^1\}, \ldots, \{L_1^m, \ldots, L_{k_m}^m\}, \}.$$

Exemplu

Fie $F = \forall x.\forall y.(P(x) \wedge (\neg Q(x) \vee R(y)))$. *Mai notăm* $F = \Big\{\{P(x)\}, \{\neg Q(x), R(y)\}\Big\}$.

Extensia Herbrand

Definiție

Fie $F = \forall x_1. \ldots .\forall x_n.G$, *cu* $G = F^*$, *o formulă în FNS. Atunci extensia Hebrand a lui* F *este mulțimea:*

$$E(F) = \Big\{ G\sigma \mid dom(\sigma) = \{x_1, \ldots, x_n\}, \sigma(x_i) \in \mathcal{D}(\mathcal{F}) \Big\}.$$

Exemplu

Fie $\mathcal{F}_0 = \{c\}, \mathcal{F}_1 = \{h\}, \mathcal{F}_2 = \{f\}$. *Atunci:*

$$\mathcal{D}(\mathcal{F}) = \{c, h(c), h(h(c)), f(c, c), f(h(c), c), \ldots\},$$

$$E(\forall x.P(x)) = \{P(c), P(h(c)), P(h(h(c))), P(f(c, c)), P(f(h(c), c)), \ldots\}$$

Extensia Herbrand generalizată

Definiție

Fie $F = \forall x_1. \ldots . \forall x_n . G$, *cu* $G = F^* = C_1 \wedge \ldots \wedge C_m$, *o formulă în FNSC. Atunci extensia Hebrand a lui* F *este mulțimea:*

$$E'(F) = \bigcup_{i \in \{1 \ldots m\}} E(\forall x_1. \ldots . \forall x_n . C_i).$$

Exemplu

Fie $\mathcal{F}_0 = \{c\}, \mathcal{F}_1 = \{h\}, \mathcal{F}_2 = \{f\}$. *Atunci:*

$$E'(\forall x.(P(x) \wedge Q(x)) = \Big\{ P(c), P(h(c)), P(h(h(c))), P(f(c,c)), \\ Q(c), Q(h(c)), Q(h(h(c))), Q(f(c,c)), Q(f(h(c),c)), \ldots \Big\}.$$

Rezoluția de bază

$$\frac{P(t_1, \ldots, t_n) \vee C \qquad \neg P(t_1, \ldots, t_n) \vee D}{C \vee D} \text{ Rezolutie de baza}$$

Exemplu

$$F = \forall x. \forall y. \forall z. (P(x) \wedge (\neg P(y) \vee Q(y)) \wedge \neg Q(z)).$$

Avem că

$$E'(F) = \{P(c), \neg P(c) \vee Q(c), \neg Q(c), \ldots\}.$$

1. $P(c)$
2. $\neg P(c) \vee Q(c)$
3. $\neg Q(c)$
4. $Q(c)$ *rezoluție între* 1 *și* 2
5. $\square$ *rezoluție între* 3 *și* 4

Rezoluția de bază

Teoremă (Herbrand)

F nu este satisfiabilă dacă și numai dacă există o submulțime finită M a lui $E'(F)$ astfel încât se poate obține $\square$ din M aplicând regula rezoluției de bază.

Rezoluția de bază - semialgoritmul lui Gilmore

- Input: o formulă F în FNSC
- Output: da, dacă formula nu este satisfiabilă (altfel rulează la infinit)

1. $M \leftarrow \emptyset$ o mulțime de clauze
2. $i \leftarrow 0$
3. cât timp $\square \notin Res^i(M)$:
   3.1 alege $C \in E'(F) \setminus M$
   3.2 $M \leftarrow M \cup \{C\}$
   3.3 $i \leftarrow i + 1$
4. întoarce DA

A step back

1. Vrem să determinăm dacă o formulă F este validă
2. Construim $\neg F$ (F este validă dacă și numai dacă $\neg F$ nu este satisfiabilă)
3. Calculăm F', o formă normală Skolem clauzală pentru $\neg F$
4. Aplicăm semialgoritmul lui Gilmore
   - Dacă algoritmul se termină, atunci $\neg F$ nu este satisfiabilă și deci F este validă (suntem bucuroși)
   - Altfel...

Exercițiu

Arătați că $\forall x.(P(x) \to Q(x)) \wedge P(s) \to Q(s)$ *este validă.*

Rezoluție în LP1

1. semialgoritmul lui Gilmore este doar de interes teoretic
2. avem nevoie de o metodă mai... practică

Reminder - Substituții

Definiție

O substituție este o funcție $\sigma : \mathcal{X} \to \mathcal{T}$, *cu proprietatea că* $\sigma(x) \neq x$ *pentru un număr finit de variabile* $x \in \mathcal{X}$.

Definiție

Dacă $\sigma : \mathcal{X} \to \mathcal{T}$ *este o substituție, atunci mulțimea* $dom(\sigma) = \{x \mid \sigma(x) \neq x\}$ *se numește domeniul substituției* σ.

Fie funcția $\sigma : \mathcal{X} \to \mathcal{T}$, astfel încât: $\sigma(x) = h(x)$, $\sigma(y) = f(h(x), a)$, $\sigma(z) = z$, pentru orice variabilă $z \in \mathcal{X} \setminus \{x, y\}$.
Conform definiției de mai sus, σ este o substituție de domeniu $dom(\sigma) = \{x, y\}$.
Noutate: vom nota $\sigma = \{x \mapsto h(x), y \mapsto f(h(x), a)\}$.

Reminder - Substituții - aplicare

Fie $\sigma = \{x \mapsto h(x), y \mapsto f(h(x), a)\}$. Avem:

1. $g(x)\sigma = g(h(x))$.
2. $\Big(\forall x.(P(x) \wedge Q(h(y)))\Big)\sigma = \forall x.(P(x) \wedge Q(h(f(h(x), a))))$.

Unificare

Definiție

O substituție σ este unificator pentru t_1 și t_2 dacă $t_1\sigma = t_2\sigma$.

Exemplu

Fie termenii $t_1 = f(x, h(y))$ și $t_2 = f(h(z), z')$. Un unificator pentru t_1 și t_2 ar fi:

$$\sigma = \{x \mapsto h(z), z' \mapsto h(y)\} \qquad (t_1\sigma = f(h(z), h(y)) = t_2\sigma)$$

Un altul:

$$\sigma' = \{z \mapsto a, x \mapsto h(a), z' \mapsto h(y)\} \qquad (t_1\sigma' = f(h(a), h(y)) = t_2\sigma').$$

Termenii $t_1 = f(x, y)$ și $t_2 = h(z)$ nu au unificator. (De ce? Pentru orice substituție σ avem $t_1\sigma = f(x, y)\sigma = f(x\sigma, y\sigma) \neq h(z\sigma) = h(z)\sigma = t_2\sigma$.)
Termenii $t_1 = x$ și $t_2 = h(x)$ nu au unificator (De ce? Pentru orice substituție σ, avem că $len(x\sigma) < len(h(x\sigma)) = len(h(x)\sigma)$).

Problemă de unificare

Definiție

O problemă de unificare P este:

- *sau o mulțime*

$$P = \{t_1 \doteq t'_1, \dots, t_n \doteq t'_n\}$$

  formată din n perechi de termeni

- *sau simbolul special*

$$P = \bot.$$

Definiție

O problemă de unificare are soluție (sau are unificator) dacă este de forma

$$P = \{t_1 \doteq t'_1, \dots, t_n \doteq t'_n\}$$

și există o substituție σ care să fie unificator pentru t_i și t'_i pentru orice $i \in \{1, \dots, n\}$, adică $t_1\sigma = t'_1\sigma, \dots, t_n\sigma = t'_n\sigma$.

Unificatori mai generali

Exemplu

Dacă σ *și* σ' *sunt doi unificatori pentru termenii* t_1 *și* t_2, σ *este un unificator mai general decât* σ' *dacă există o substituție* σ'' *astfel încât* $t_1\sigma' = t_1\sigma\sigma'' = t_2\sigma\sigma'' = t_2\sigma'$.

Exemplu

Fie $t_1 = f(x, a)$ *și* $t_2 = f(y, a)$. *Unificatorul* $\{y \mapsto x\}$ *este mai general decât* $\{x \mapsto a, y \mapsto a\}$.

Cel mai general unificator (most general unifier)

Definition

Substituția σ este cel mai general unificator pentru o problemă de unificare $P = \{t_1 \doteq t_1', \ldots, t_n \doteq t_n'\}$ dacă:

1. σ este unificator pentru P: $t_i\sigma = t_i'\sigma$, pentru orice $1 \leq i \leq n$;
2. σ este mai general decât orice alt unificator pentru P.

Definiție

Cu $mgu(P)$ notăm mulțimea unificatorilor cei mai generali pentru P. Pentru $P = \bot$, $mgu(P) = \emptyset$.

Example

Fie $P = \{f(x, a) \doteq f(y, a)\}$. Avem că
$mgu(P) = \{\{x \mapsto z, y \mapsto z\}, \{x \mapsto y\}, \ldots\}$.

Problemă de unificare - formă rezolvată

Definiție

O problemă de unificare P *este în formă rezolvată dacă* $P = \bot$ *sau* $P = \{x_1 \doteq t'_1, \ldots, x_n \doteq t'_n\}$ *și* $x_i \notin vars(t_j)$ *pentru orice* $i, j \in \{1, \ldots, n\}$.

Lemă

Dacă $P = \{x_1 \doteq t'_1, \ldots, x_n \doteq t'_n\}$ *este în formă rezolvată, atunci* $\{x_1 \mapsto t'_1, \ldots, x_n \mapsto t'_n\} \in mgu(P)$.

Aducerea unei probleme de unificare în formă rezolvată

Stergere	$P \cup \{t \doteq t\} \Rightarrow P$
Descompunere	$P \cup \{f(t_1, \ldots, t_n) \doteq f(t'_1, \ldots, t'_n)\} \Rightarrow$
	$P \cup \{t_1 \doteq t'_1, \ldots, t_n \doteq t'_n\}$
Orientare	$P \cup \{f(t_1, \ldots, t_n) \doteq x\} \Rightarrow P \cup \{x \doteq f(t_1, \ldots, t_n)\}$
Eliminare	$P \cup \{x \doteq t\} \Rightarrow P\{x \mapsto t\} \cup \{x \doteq t\}$
	daca $x \notin vars(t), x \in vars(P)$
Conflict	$P \cup \{f(t_1, \ldots, t_n) \doteq g(t'_1, \ldots, t'_m)\} \Rightarrow \bot$
Occurs check	$P \cup \{x \doteq f(t_1, \ldots, t_n)\} \Rightarrow \bot$
	daca $x \in vars(f(t_1, \ldots, t_n))$

Proprietățile regulilor de rescriere de mai sus

Lemă

Dacă P nu este în formă rezolvată, atunci există P' astfel încât $P \Rightarrow P'$.

Lemă

Dacă $P \Rightarrow P'$, atunci $mgu(P) = mgu(P')$.

Lemă

Nu există o secvență infinită $P \Rightarrow P_1 \Rightarrow P_2 \Rightarrow \ldots \Rightarrow P_i \Rightarrow \ldots$.

Corolar

Regulile precedente constituie un algoritm de calcul al celui mai general unificator pentru o problemă de unificare, dacă acesta există.

Exemplu 1

Exemplu

$$P = \{f(g(x_1, a), x_2) \doteq x_3, f(x_2, x_2) \doteq f(a, x_1)\} \overset{\text{DESCOMPUNERE}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, x_2 \doteq a, x_2 \doteq x_1\} \overset{\text{ELIMINARE}}{\Rightarrow}$$

$$\{f(g(x_1, a), a) \doteq x_3, x_2 \doteq a, a \doteq x_1\} \overset{\text{ORIENTARE}}{\Rightarrow}$$

$$\{f(g(x_1, a), a) \doteq x_3, x_2 \doteq a, x_1 \doteq a\} \overset{\text{ELIMINARE}}{\Rightarrow}$$

$$\{f(g(a, a), a) \doteq x_3, x_2 \doteq a, x_1 \doteq a\} \overset{\text{ORIENTARE}}{\Rightarrow}$$

$$\{x_3 \doteq f(g(a, a), a), x_2 \doteq a, x_1 \doteq a\}$$

Exemplu 2

$$P = \{f(g(x_1, a), x_2) \doteq x_3, f(x_2) \doteq f(x_3)\} \overset{\textsc{Descompunere}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, x_2 \doteq x_3\} \overset{\textsc{Orientare}}{\Rightarrow}$$

$$\{x_3 \doteq f(g(x_1, a), x_2), x_2 \doteq x_3\} \overset{\textsc{Eliminare}}{\Rightarrow}$$

Explicati de ce nu se mai poate aplica orientare

$$\{x_3 \doteq f(g(x_1, a), x_3), x_2 \doteq x_3\} \overset{\textsc{Occurs check}}{\Rightarrow}$$

$$\perp$$

Concluzie: $mgu(P) = \emptyset$.

Exemplu 3

$$P = \{f(g(x_1, a), x_2) \doteq x_3, f(g(x_4, x_5)) \doteq f(x_3)\} \overset{\text{DESCOMPUNERE}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, g(x_4, x_5) \doteq x_3\} \overset{\text{ORIENTARE}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq x_3, x_3 \doteq g(x_4, x_5)\} \overset{\text{ELIMINARE}}{\Rightarrow}$$

$$\{f(g(x_1, a), x_2) \doteq g(x_4, x_5), x_3 \doteq g(x_4, x_5)\} \overset{\text{CONFLICT}}{\Rightarrow}$$

$$\bot$$

Concluzie: $mgu(P) = \emptyset$.

Back to resolution

(Binary) Resolution

$$\frac{P(t_1,\ldots,t_n)\vee C \quad \neg P(t'_1,\ldots,t'_n)\vee D \quad \sigma\in mgu\{t_1\doteq t'_1,\ldots t_n\doteq t'_n\} \quad var(P(t_1,\ldots,t_n)\vee C)\cap var(\neg P(t'_1,\ldots,t'_n)\vee D)=\emptyset}{(C\vee D)\sigma}$$

(Positive) Factoring

$$\frac{P(t_1,\ldots,t_n)\vee P(t'_1,\ldots,t'_n)\vee C \qquad \sigma\in mgu\{t_1\doteq t'_1,\ldots t_n\doteq t'_n\}}{(P(t_1,\ldots,t_n)\vee C)\sigma}$$

1. $P(x)$
2. $\neg P(h(x))\vee Q(f(x))$
3. $\neg Q(f(g(a)))$
4. $Q(f(x))$ rezoluție între 1 și 2:
$$\frac{P(x') \qquad \neg P(h(x))\vee Q(f(x)) \qquad \{x'\mapsto h(x)\}\in mgu\{x'\doteq h(x)\}}{Q(f(x))\{x'\mapsto h(x)\}}$$
5. $\square$ rezoluție între 3 și 4:
$$\frac{Q(f(g(a))) \qquad Q(f(x)) \qquad \{x\mapsto g(a)\}\in mgu\{f(g(a))\doteq f(x)\}}{\square\{x\mapsto g(a)\}}$$

A step back

1. Fie F în FNSC reprezentată ca mulțime de clauze: $F = \{C_1, \ldots, C_n\}$.
2. F este nesatisfiabilă dacă și numai dacă $\Box$ se poate obține din $\{C_1, \ldots, C_n\}$ aplicând regulile $\textsc{Resolution}$ și $\textsc{Factoring}$.

Exemplu

Arătați că $\forall x.(P(x) \rightarrow Q(x)) \wedge P(s) \rightarrow Q(s)$ *este validă folosind rezoluția.*