Logica cu predicate de ordinul I
Curs 2 - Semantica LP1

Ștefan Ciobâcă

5 Decembrie 2016

Note organizatorice

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

Cuprins

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):

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

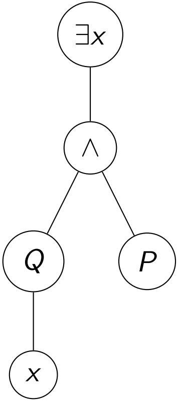
Reminder - Sintaxa - Exemple

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 - Arborele abstract - Exemplu

$arb((\exists x.(Q(x) \wedge P))) =$



Reminder - Despre paranteze

La fel cum scriem $-3 \times 4 + 5$ în loc de $(((-3) \times 4) + 5)$, vom renunța la paranteze după cum urmează:

1. vom scrie $\neg P(x) \wedge Q(y) \vee R(x, y)$ în loc de $\Big(\Big((\neg P(x)) \wedge Q(y)\Big) \vee R(x, y)\Big)$ (la fel ca la logica propozițională).

Ordinea de prioritate: $\neg, \forall, \exists, \wedge, \vee, \rightarrow$.

Exemplu

$$\forall x.(P(x) \wedge \neg Q(x) \rightarrow R(x)) \textit{ în loc de}$$

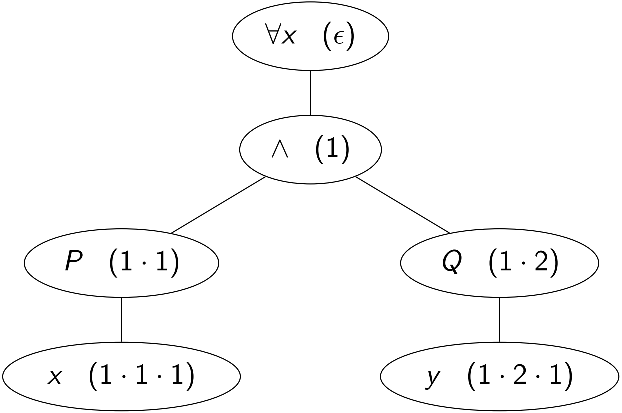
$$(\forall x.\Big(\big(P(x) \wedge (\neg Q(x))\big) \rightarrow R(x)\Big)) \textit{ în loc de}$$

When in doubt: use parantheses.

Reminder - Poziții

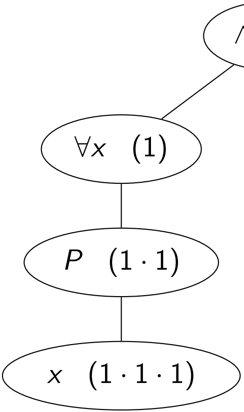
Exemplu

$\Big((\forall x)(P(x) \wedge Q(y))\Big)|_{1\cdot 2} = Q(y)$



Reminder - Apariții - Exemplu

$$F = (\forall x.P(x)) \wedge Q(x)$$



$$bound(F) = \{x\}, free(F) = \{x\}.$$

Reminder - Substituții

Fie substituția $\sigma : \mathcal{X} \to \mathcal{T}$, astfel încât:

1. $\sigma(x) = h(x)$;
2. $\sigma(y) = f(h(x), a)$;
3. $\sigma(z) = z$, pentru orice variabilă $z \in \mathcal{X} \setminus \{x, y\}$.

Avem că $(\forall x.(P(x) \land Q(h(y))))\sigma = \forall x.(P(x) \land Q(h(f(h(x), a))))$.

Cuprins

Funcții

O funcție n-ară peste o mulțime U este orice funcție

$$f : \underbrace{U \times \ldots \times U}_{n} \to U.$$

Câteodată notăm $\underbrace{U \times \ldots \times U}_{n} = U^n$ și scriem $f : U^n \to U$.

De exemplu, $+$ este o funcție binară peste mulțimea numerelor naturale: $+ : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$.
Convenție: dacă $c : U^0 \to U$ este o funcție 0-ară peste U, vom considera că $c \in U$. Funcțiile de 0 argumente (sau 0-are, sau null-are) se numesc și constante.

Predicate

Un predicat este o funcție care întoarce o valoare de adevăr. În acest curs, mulțimea valorilor de adevăr am notat-o cu $\mathbb{B}$ și am convenit că $\mathbb{B} = \{0, 1\}$, unde 0 reprezintă valoarea de adevăr *fals* și 1 reprezintă valoarea de adevăr *adevărat*.
Un predicat n-ar peste o mulțime U este așadar orice funcție

$$P : U^N \to \mathbb{B}.$$

Un exemplu de predicat binar este predicatul $\leq$, pe care l-ați învățat încă din clasele primare (fără să știți că este un predicat):

$$\leq : \mathbb{Z} \times \mathbb{Z} \to \mathbb{B}.$$

Pentru a simplifica notația, ați folosit scrierea $x \leq y$ în loc de $\leq (x, y) = 1$ și $x \not\leq y$ în loc de $\leq (x, y) = 0$.

Funcții versus simboluri funcționale, predicate versus simboluri predicative

În cursul precedent, am discutat despre *simboluri funcționale* și despre *simboluri predicative*. Care este diferența dintre un simbol funcțional și o funcție? Care este diferența dintre un simbol predicativ și un predicat? Un simbol este doar un nume, similar cu un identificator într-un limbaj de programare. Simbolul funcțional este un nume care ține locul unei funcții, în timp ce simbolul predicativ ține locul unui predicat. Aritatea unui simbol funcțional/predicativ reprezintă numărul de argumente al funcției/predicatului căreia/căruia îi ține locul.

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: $I_x \in U$.*

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
$$I_x = 7,$$

Valoarea unui termen într-o structură

Dacă $S = (U, I)$ este o structură și $t \in T$ este un termen, atunci *valoarea termenului* t *în structura* S (sau *interpretarea termenului* t *în structura* S) se notează cu $S(t)$ și este definită inductiv astfel:

1. $S(c) = I_c$, pentru orice simbol funcțional $c \in \mathcal{F}_0$ de aritate 0;
2. $S(x) = I_x$, pentru orice variabilă $x \in \mathcal{X}$;
3. $S(f(t_1, \ldots, t_n)) = I_f(S(t_1), \ldots, S(t_n))$, pentru orice simbol funcțional $f \in \mathcal{F}_n$ de aritate $n \geq 1$ și pentru orice termeni $t_1, \ldots, t_n \in T$.

Valoarea unui termen într-o structură

Exemplu

Fie termenul $t = S(f(f(x,e),i(x)))$. *Avem că*

$$\begin{aligned}
S(t) &= S\Big(f\big(f(x,e),i(x)\big)\Big)\\
&= I_f\Big(S\big(f(x,e)\big),S\big(i(x)\big)\Big)\\
&= \Big(S\big(f(x,e)\big)+S\big(i(x)\big)\Big)\\
&= I_f\big(S(x),S(e)\big)\big)+I_i\big(S(x)\big)\\
&= \big(S(x)+S(e)\big)+\big(-S(x)\big)\\
&= (I_x+I_e)+(-I_x)\\
&= (7+0)+\big(-7\big)\\
&= 0.
\end{aligned}$$

Actualizarea valorii unei variabile într-o structură

Fie $S = (U, I)$ o structură, $x \in \mathcal{X}$ o variabilă și $u \in U$ un element al universului.
Cu $S[x \mapsto u]$ notăm o nouă structură $S[x \mapsto u] = (U', I')$, definită astfel:

1. $U' = U$;
2. $I'_x = u$;
3. $I'_o = I_o$, pentru orice $o \in \mathcal{F} \cup \mathcal{P} \cup \mathcal{X} \setminus \{x\}$.

Numim $S[x \mapsto u]$ structura S după actualizarea valorii variabilei x la u. Continuând exemplul precedent, calculați $S[y \mapsto 3](f(x, f(e, y)))$ (răspunsul trebuie să fie 10).

Valoarea de adevăr a unei formule într-o structură

Dacă S este o structură și $F \in \texttt{LP1}$ este o formulă, atunci *valoarea de adevăr a formulei F în structura S* se notează cu $S(F)$ (vom avea $S(F) \in \mathbb{B}$) și este definită inductiv astfel:

1. $S(P(t_1, \ldots, t_n)) = I_P(S(t_1), \ldots, S(t_n))$, pentru orice predicat P de aritate $n \geq 0$;
2. $S(F_1 \wedge F_2) = S(F_1) \cdot S(F_2)$, pentru orice formule $F_1, F_2 \in \texttt{LP1}$ (funcția $\cdot : \mathbb{B} \times \mathbb{B} \to \mathbb{B}$ este aici funcția *și* logic);
3. $S(\neg F) = \overline{S(F)}$, pentru orice formulă $F \in \texttt{LP1}$ (funcția $\overline{\cdot} : \mathbb{B} \to \mathbb{B}$ este aici funcția *negație* logică);
4. $S(\exists x.F) =$
$$\begin{cases} 1 & \text{dacă există un element } u \in U \text{ a.î. } S[x \mapsto u](F) = 1, \\ 0 & \text{altfel.} \end{cases}$$

Valoarea de adevăr a unei formule într-o structură - exemplu

Continuând exemplul precedent, avem că:

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

Exercițiu

Scrieți formulele care exprimă că e este element neutru la dreapta pentru f și că i este funcția invers.

Cuprins

Formulă validă

Exemplu

Formula $equals(x, y) \vee \neg equals(x, y) \in \mathtt{LP1}$ *este adevărată în orice structură* S.

Definiție (Formulă validă)

Fie $F \in \mathtt{LP1}$ *o formulă. Spunem că* F *este o formulă validă (sau, în mod echivalent, o tautologie) dacă, pentru orice structură* $S = (U, I)$, *avem că* $S(F) = 1$.

Formulă satisfiabilă

Definiție (Formulă satisfiabilă)

Fie $F \in \mathtt{LP1}$ o formulă. Spunem că F este o formulă satisfiabilă dacă există o structură $S = (U, I)$ a.î. $S(F) = 1$.

Exercițiu

Dați exemplu de o formulă satisfiabilă.

Observația

O formulă care nu este satisfibilă se numește *formulă nesatisfiabilă*, sau, în mod echivalent, *contradicție*.

Formule echivalente

Două formule $F_1, F_2 \in \texttt{LP1}$ se numesc *(tare) echivalente* dacă au aceeași valoare de adevăr în orice structură: pentru orice structură $S = (U, I)$, $S(F_1) = S(F_2)$. Faptul că F_1 este echivalentă cu F_2 se notează cu $F_1 \equiv F_2$.

Exemplu

1. $P(x) \vee Q \equiv Q \vee P(x)$;
2. $\forall x.P(x) \equiv \forall y.P(y)$.

Teorema echivalențelor

Pentru orice $F, G, H \in \mathtt{LP1}$, pentru orice $x \in \mathcal{X}$,

$$F \wedge G \equiv G \wedge F, \qquad F \vee G \equiv G \vee F;$$

$$F \wedge (G \wedge H) \equiv (F \wedge G) \wedge H, \qquad F \vee (G \vee H) \equiv (F \vee G) \vee H;$$

$$F \wedge (G \vee H) \equiv (F \wedge G) \vee (F \wedge H), F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H),$$

$$\neg(F \wedge G) \equiv (\neg F) \vee (\neg G), \neg(F \vee G) \equiv (\neg F) \wedge (\neg G);$$

$$\neg(\forall x.F) \equiv \exists x.(\neg F), \qquad \neg(\exists x.F) \equiv \forall x.(\neg F);$$

dacă $x \notin \mathit{free}(G)$, atunci:

$$(\forall x.F) \vee G \equiv \forall x.(F \vee G), \qquad (\exists x.F) \vee G \equiv \exists x.(F \vee G),$$

$$(\forall x.F) \wedge G \equiv \forall x.(F \wedge G), \qquad (\exists x.F) \wedge G \equiv \exists x.(F \wedge G).$$

Consecință semantică

Fie $n+1$ formule $F_1, \ldots, F_n, F \in \mathtt{LP1}$ ($n \geq 0$). Formula F este *consecință semantică* a formulelor $F_1, \ldots, F_n$ dacă, pentru orice structură $S = (U, I)$, dacă formulele $F_1, \ldots F_n$ sunt adevărate în S, atunci și F este adevărată în S.
Faptul că F este consecință semantică din $F_1, \ldots, F_n$ se notează cu $F_1, \ldots, F_n \models F$.

Exercițiu

Arătați că $\forall x.P(x) \models P(c)$*, unde* P *este un simbol predicativ unar,* c *este un simbol funcțional de aritate* 0*, iar* x *este o variabilă.*

Cuprins

Problema validității/satisfiabilității

În continuare, cursul este dedicat rezolvării următoarei probleme:
`Input`: O formulă $F \in \texttt{LP1}$.
`Output`: Da, dacă formula este validă.
Din păcate, problema de mai sus este *nedecidabilă*: nu există niciun algoritm care să o rezolve.
Totuși, vom găsi un *semialgoritm* pentru problema de mai sus:

1. Semialgoritmul se oprește cu răspunsul "da" ddacă formula este validă;
2. Dacă răspunsul este "nu", atunci formula nu este validă;
3. Dacă formula nu este validă, există posibilitatea ca semialgoritmul să bucleze la infinit.

De fapt, din rațiuni istorice, ne vom concentra asupra problemei nesatisfiabilității. Dar F este validă ddacă $\neg F$ este nesatisfiabilă și deci dacă rezolvăm una din probleme am rezolvat-o și pe cealaltă.

Cuprins

Apariție a unei formule

Fie $F \in \texttt{LP1}$ o formulă și $p \in pos(F)$ o poziție a formulei F la care se află o formulă $G \in \texttt{LP1}$: $F|_p = G$. Atunci spunem că *p este o apariție a lui G în F*.

Fie $F, G \in \texttt{LP1}$ două formule și $p \in pos(F)$ o apariție a formulei G în F. Atunci formula obținută din F prin înlocuirea apariției p a lui G cu H, notată cu $F[H]_p$, este definită inductiv astfel:

1. $(F_1)[H]_\epsilon = H$;
2. $(F_1 \wedge F_2)[H]_{1 \cdot p} = (F_1[H]_p) \wedge F_2$;
3. $(\neg F_1)[H]_{1 \cdot p} = \neg((F_1)[H]_p)$;
4. $(\forall x.F_1)[H]_{1 \cdot p} = \forall x.((F_1)[H]_p)$;
5. $\ldots$.

Exemplu

$$\Big((P(x) \vee Q(y)) \rightarrow \forall z.R(x, y, z)\Big)[P(a)]_{2 \cdot 1} =$$

$$\Big((P(x) \vee Q(y)) \rightarrow \forall z.P(a)\Big)$$

Teorema substituției

Fie $F, G, H \in \texttt{LP1}$, $p \in pos(F)$ astfel încât:

1. $G \equiv H$ și
2. $G = F|_p$.

Atunci $F = F[G]_p \equiv F[H]_p$.

Exemplu

Fie $G = P(x) \vee Q(x)$ *și* $H = Q(x) \vee P(x)$*. Fie* $F = \forall x.(P(x) \vee Q(x))$ *și poziția* $p = 1$*. Avem că* $F|_p = G$ *și că* $G \equiv H$*. Prin teorema substituției, putem concluziona că* $\forall x.(P(x) \vee Q(x)) \equiv \forall x.(Q(x) \vee P(x))$*.*

Lema de redenumire a variabilelor legate

Dacă $x, y \in \mathcal{X}$ sunt două variabile, $F \in \texttt{LP1}$ este o formulă astfel încât $y \notin \mathit{free}(F)$, atunci

$$\forall x.F \equiv \forall y.(F[x \mapsto y]) \qquad \text{și} \qquad \exists x.F \equiv \exists y.(F[x \mapsto y]).$$

Exemplu

Exemple de aplicare ale lemei redenumirii de variabile:

1. $\forall x.P(x, z) \equiv \forall y.P(y, z)$.
2. $\forall x.P(x, z) \not\equiv \forall z.P(z, z)$.
3. $\forall x.(P(x, z) \wedge \exists y.Q(y)) \equiv \forall y.(P(y, z) \wedge \exists y.Q(y))$.

Cuprins

FNP

O formulă $F \in \texttt{LP1}$ este în *formă normală prenex* dacă există variabilele distincte $x_1, \ldots, x_n$, cuantificatorii $Q_1, \ldots, Q_n \in \{\forall, \exists\}$ și o formulă fără cuantificatori $G \in \texttt{LP1}$, astfel încât:

$$F = Q_1 x_1. \ldots . Q_n x_n . G.$$

Exemplu

$\forall x . \exists y . \forall z . (P(x) \vee \neg(Q(x, y, z) \wedge R))$

Teoremă (Teorema de aducere în FNP)

Pentru orice formulă $F_1 \in \texttt{LP1}$, există o formulă $F_2 \in \texttt{LP1}$ aflată în FNP astfel încât $F_1 \equiv F_2$.

Teorema de aducere în FNP

Pentru orice formulă $F_1 \in \texttt{LP1}$, există o formulă $F_2 \in \texttt{LP1}$ aflată în FNP astfel încât $F_1 \equiv F_2$.
Schiță de demonstrație. Aplicăm următoarele transformări:

1. $\neg(\forall x.F) \equiv \exists x.(\neg F)$;
2. $\neg(\exists x.F) \equiv \forall x.(\neg F)$;
3. $(\forall x.F) \vee G \equiv \forall y.(F[x \mapsto y] \vee G)$ (unde $y \in \mathcal{X} \setminus free(G)$);
4. $(\exists x.F) \vee G \equiv \exists y.(F[x \mapsto y] \vee G)$ (unde $y \in \mathcal{X} \setminus free(G)$);
5. $(\forall x.F) \wedge G \equiv \forall y.(F[x \mapsto y] \wedge G)$ (unde $y \in \mathcal{X} \setminus free(G)$);
6. $(\exists x.F) \wedge G \equiv \exists y.(F[x \mapsto y] \wedge G)$ (unde $y \in \mathcal{X} \setminus free(G)$);

Example

$P(x) \vee \exists x.(Q(x) \wedge \forall x.P(x))$

Cuprins

FNS

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 $free(G) = \{x_1, \ldots, x_n\}$ și $bound(G) = \emptyset$.

Exemplu

Formule în FNS:

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

Echivalența slabă

Fie $F_1, F_2 \in \mathtt{LP1}$. Formula F_1 este *slab echivalentă* cu F_2 (notat $F_1 \equiv_s F_2$) dacă exact una din următoarele condiții este adevărată:

1. F_1 și F_2 sunt satisfiabile
2. F_1 și F_2 sunt nesatisfiabile

Cu alte cuvinte, două formule sunt slab echivalente dacă sunt *echisatisfiabile* (dacă una dintre ele este satisfiabilă atunci și cealaltă este satisfiabilă și invers).

Lema de Skolemizare

Fie $F = \forall x_1.\forall x_2.\ldots.\forall x_n.\exists y.G$, unde $n \geq 0$, $G \in \texttt{LP1}$ (G poate conține alți cuantificatori).
Fie $f \in \mathcal{F}_n$ un simbol funcțional de aritate n care nu apare în G.
Atunci:

$$F \equiv_s \forall x_1.\forall x_2.\ldots.\forall x_n.G[y \mapsto f(x_1, \ldots, x_n)].$$

Schiță.

Presupunem că exists o structură S astfel încât $S(F) = 1$. Găsim o structură S' astfel încât
$S'(\forall x_1.\forall x_2.\ldots.\forall x_n.G[y \mapsto f(x_1, \ldots, x_n)]) = 1$.
Presupunem că exists o structură S' astfel încât
$S'(\forall x_1.\forall x_2.\ldots.\forall x_n.G[y \mapsto f(x_1, \ldots, x_n)]) = 1$. Găsim o structură S astfel încât $S(F) = 1$.

$\square$

Cuprins

Literal

O formulă $F \in \texttt{LP1}$ se numește *literal* dacă există un predicat $P \in \mathcal{P}_n$ de aritate $n \geq 0$ și n termeni $t_1, \ldots, t_n$ astfel încât

$$F = P(t_1, \ldots, t_n)$$

sau

$$F = \neg P(t_1, \ldots, t_n).$$

Informal, spunem că un literal este o formulă atomică sau negația unei formule atomice.

Exemplu

Exemple de literali:

$$P \qquad \neg Q(x, f(y)) \qquad R(a, f(x), b)$$

$$\neg P \qquad Q(x, f(y)) \qquad R(a, f(x), b)$$

Exemple de formule care nu sunt literali:

$$P \wedge Q(x, y) \qquad \neg\neg P(x) \qquad \forall x. Q(x, x)$$

Clauză

O formulă $F \in \texttt{LP1}$ se numește *clauză* dacă există n literali $L_1, \ldots, L_n \in \texttt{LP1}$ astfel încât

$$F = L_1 \vee L_2 \vee \ldots \vee L_n.$$

Exemplu

$$F_1 = P(x) \vee Q(x, a) \vee \neg R(x, f(a, y)). \quad F_2 = P(x) \quad F_3 = \square.$$

FNC

O formulă F este *în formă normală clauzală* (sau, echivalent, *în formă normală conjunctivă*) dacă există n clauze $C_1, \ldots, C_n$ astfel încât

$$F = C_1 \wedge C_2 \wedge \ldots \wedge C_n.$$

Exemplu

$$F_1 = (P(x) \vee Q(x)) \wedge (\neg P(x) \vee R(x, y))$$

$$F_2 = (P(x) \vee Q(x) \vee Q(a)) \wedge (\neg P(x)) \wedge (\neg Q(x) \vee R(x, z)).$$

Teorema de aducere în FNC

Fie F o formulă fără cuantificatori. Există o formulă G, echivalentă cu F, aflată în FNC.

Schiță de demonstrație.

Se aplică următoarele transformări (plus, eventual, comutativitatea și asociativitatea lui $\vee, \wedge$):

$$F \vee (G \wedge H) \equiv (F \vee G) \wedge (F \vee H),$$

$$\neg(F \wedge G) \equiv (\neg F) \vee (\neg G),$$

$$\neg(F \vee G) \equiv (\neg F) \wedge (\neg G);$$

Exemplu

$P(x) \vee \neg(P(x) \vee Q(x, y))$

Matricea unei formule

Fie $F \in \texttt{LP1}$ o formulă. *Matricea* lui F este formula notată cu F^*, obținută din F prin ștergerea tuturor cuantificatorilor. Formal, avem că:

1. $(\forall x.G)^* = G^*$ pentru orice $G \in \texttt{LP1}$;
2. $(\exists x.G)^* = G^*$ pentru orice $G \in \texttt{LP1}$;
3. $(\neg G)^* = (\neg(G^*))$ pentru orice $G \in \texttt{LP1}$;
4. $(G_1 \wedge G_2)^* = (G_1^* \wedge G_2^*)$ pentru orice $G_1, G_2 \in \texttt{LP1}$;
5. $(G_1 \vee G_2)^* = (G_1^* \vee G_2^*)$ pentru orice $G_1, G_2 \in \texttt{LP1}$;
6. $P(t_1, \ldots, t_n)^* = P(t_1, \ldots, t_n)$ pentru orice simbol predicativ $P \in \mathcal{F}_n$ și orice termeni $t_1, \ldots t_n \in \mathcal{T}$.

Exemplu

$$\Big(P(a) \wedge \forall x.(P(x) \rightarrow \exists y.R(x,y))\Big)^* = P(a) \wedge (P(x) \rightarrow R(x,y))$$

FNSC

O formulă F este în formă normală Skolem clauzală (FNSC) dacă

1. F este în formă normală Skolem și
2. F^* este în formă normală clauzală.

Exemplu

$$F_1 = \forall x.\forall y.\Big((P(x) \vee \neg Q(x, y)) \wedge (Q(a, y) \vee \neg P(a))\Big)$$

$$F_2 = \forall x.\forall y.\Big(P(a) \wedge (\neg R(x) \vee Q(y))\Big)$$

Exemple de formule care nu sunt în FNSC:

$$F_1 = \exists x.P(x) \qquad F_2 = \forall x.\Big(P(a) \wedge (\neg R(x) \vee Q(y))\Big)$$

$$F_3 = \forall x.\forall y.\Big(P(a) \wedge \neg(R(x) \vee Q(y))\Big)$$

Aducerea unei formule în FNSC - rezumat

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

Cursul următor: metoda rezoluției pentru a testa satisfiabilitatea unei formule aflată în FNSC.