Curs 9 - Logica de ordinul I (LP1) - Sintaxa

cuantificator

p : Orice om este muritor. — Muritor(x) — M(x)

q : Socrate este om — Om(x)

r : Socrate este muritor

Cuantificator : $\forall$ $\exists$

Predicate : afirmații (relații) a căror valoare de adevăr (A/F) depinde de 0 sau mai mulți parametri

$$\forall x. \left(Om(x) \to Muritor(x) \right)$$

$$Om(s)$$ s – Socrate

$$Muritor(s)$$

Structură $S = (D, Pred, Fun)$

domeniul (mulțime nevidă)

predicate

funcții

Între orice 2 elemente există un al treilea.

$$\varphi = \left(\forall x. \left(\forall y. \left(x < y \to \exists z. \left(x < z \land z < y \right) \right) \right) \right)$$

Predicat $<(x, y)$

Ex de struct : $S_1 = (\mathbb{R}, \{<, =\}, \{+, -, 0, 1\})$

diferite

$+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$

funcție cu 0 parametri (constante)

$S_2 = (\mathbb{Z}, \{<, =\}, \{+, -, 0, 1\})$

$+ : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}$

Orice număr natural are succesor.

$$\left(\forall x.\left(Nat(x) \rightarrow \exists y.\left(succ(x)=y\right)\right)\right)$$

dacă x este natural — atunci există y a.î. y = succesorul lui x

Signatură $\Sigma = (\mathcal{P}, \mathcal{F})$

nume + aritate — simboluri predicative (simbolurile folosite pt predicate)

simboluri funcționale (simboluri pt funcții) — nume + aritate

+ arități (nr. de parametri)

pt orice $s \in \mathcal{P} \cup \mathcal{F}$ avem $ar(s) \in \mathbb{N}$

Σ-Structură: Pentru $\Sigma = (\mathcal{P}, \mathcal{F})$, $S = (D, Pred, Fun)$ este o Σ-structură dacă pt fiecare simbol predicativ / funcțional există un singur predicat / funcție în S corespunzător.

$$\Sigma = (\{P, Q\}, \{f, i, a, b\})$$

$ar(P) = 2$, $ar(Q) = 2$, $ar(f) = 2$, $ar(i) = 1$, $ar(a) = ar(b) = 0$

$$\left(\forall x.\left(\forall y.\left(P(x,y) \rightarrow \left(\exists z.\left(P(x,z) \land P(z,y)\right)\right)\right)\right)\right)$$

S_1 este o Σ-struct

P^{S_1} este $<$ — corespondentul lui P în S_1

Q^{S_1} este $=$

f^{S_1} este $+$

i^{S_1} este $-$

a^{S_1} este 0

b^{S_1} este 1

$\mathcal{P}_n$ – mulțimea de simboluri pred cu n parametri

$\mathcal{F}_n$ – ——— " ——— funcționale ——— " ———

Alfabetul LP1

1) conectori logici : $\neg, \wedge, \vee, \rightarrow, \leftrightarrow, \perp$

   cuantificatori : $\forall, \exists$

2) variabile : $\mathcal{X} = \{x, y, z, x', y', x_1, \ldots\}$

3) simboluri auxiliare : $(\ )$. , $(\ )$,

   $+(x, y)$

4) simboluri specifice signaturii $\Sigma = (\mathcal{P}, \mathcal{F})$

Termeni

$\mathcal{T}$ – cea mai mică mulțime a.î.

cazuri de bază

1. $\mathcal{F}_0 \subseteq \mathcal{T}$ (orice simbol constant este termen) $a \in \mathcal{T}$ $b \in \mathcal{T}$

2. $\mathcal{X} \subseteq \mathcal{T}$ (orice variabilă este termen) $x \in \mathcal{T}$ $y \in \mathcal{T}$ ...

Σ-termeni

3. dacă $f \in \mathcal{F}_n$ $(n > 0)$ și $t_1, \ldots, t_n \in \mathcal{T}$
   atunci $f(t_1, \ldots, t_n) \in \mathcal{T}$

$\Sigma = (\{P, Q\}, \{f, i, a, b\})$ $\quad ar(f) = 2 \quad ar(a) = 0$
$\quad ar(i) = 1 \quad ar(b) = 0$

Ex $\quad a \quad b \quad x \quad y \quad z \ldots$

$f(a, x) \qquad i(a) \qquad i(f(a, x))$

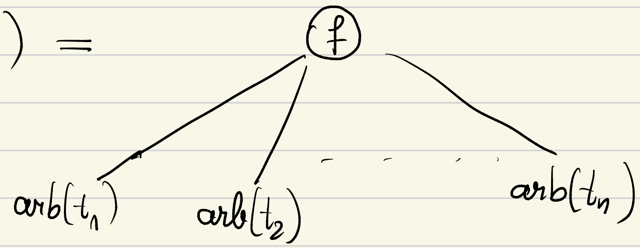
$i(i(a))$

$\cancel{P(a, b)}$ nu este termen.

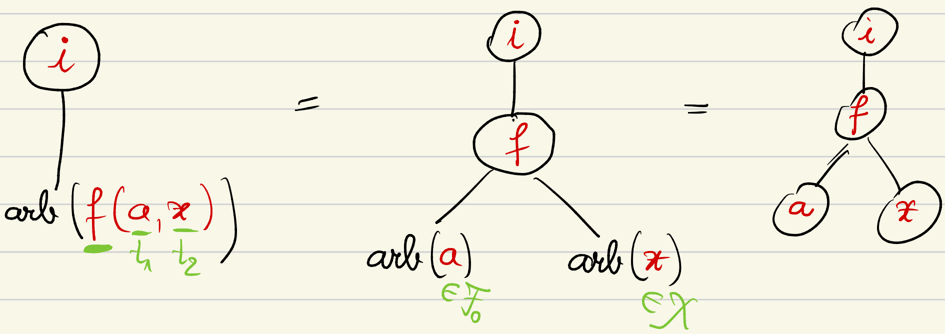
Arborele de sintaxă abstract pt termeni $t \in \mathcal{T}$

1. dacă $t = c \in \mathcal{F}_0$ atunci $arb(t) =$ (c)
2. dacă $t = x \in \mathcal{X}$ atunci $arb(t) =$ (x)
3. dacă $t = f(t_1, t_2, \ldots, t_n)$ și $f \in \mathcal{F}_n$ $(n>0)$, $t_1, \ldots, t_n \in \mathcal{T}$

atunci $arb(t) =$



$arb(\underbrace{i}_{f}(\underbrace{f(a,x)}_{t_1})) =$



Formulă atomică $P(t_1, \ldots t_n)$ unde $P \in \mathcal{P}_n$

$t_1, \ldots t_n \in \mathcal{T}$

dacă $n = 0$ $P(\ )$ înlocuit P

Ex $\Sigma = (\{P, Q\}, \{f, i, a, b\})$

$ar(f) = 2$ $ar(a) = 0$
$ar(i) = 1$ $ar(b) = 0$
$ar(P) = 2$ $ar(Q) = 2$

$Q(b, i(a))$

$P(f(a,x), i(b))$

Sintaxa LP1

$\mathbb{LP}1$ – cea mai mică mulțime a.î.

1. orice formulă atomică este din $\mathbb{LP}1$. $\big(P(t_1, \dots t_n) \in \mathbb{LP}1\big)$ pt orice $P \in \mathcal{P}_n$, $t_1 \dots t_n \in \mathcal{T}$
2. pt orice $\varphi_1, \varphi_2 \in \mathbb{LP}1$ avem
   a) $\neg \varphi_1 \in \mathbb{LP}1$
   b) $(\varphi_1 \wedge \varphi_2) \in \mathbb{LP}1$
   c) $(\varphi_1 \vee \varphi_2) \in \mathbb{LP}1$
   d) $(\varphi_1 \rightarrow \varphi_2) \in \mathbb{LP}1$
   e) $(\varphi_1 \leftrightarrow \varphi_2) \in \mathbb{LP}1$
   f) $(\forall x. \varphi_1) \in \mathbb{LP}1$ $\quad x \in \mathcal{X}$
   g) $(\exists x. \varphi_1) \in \mathbb{LP}1$

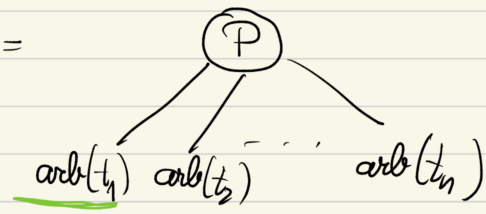
Ex $\quad P(a, b) \qquad Q(b, i(a))$

$\neg P(a, b) \qquad (P(a,b) \wedge Q(b, i(a)))$

$(\forall x. (P(a, y) \wedge Q(b, i(a))))$

Arborele de sintaxă abstractă pt $\varphi \in \mathbb{LP}1$.

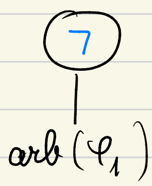
1. $\varphi = P(t_1, \dots t_n)$ atunci $\underline{arb}(\varphi) =$



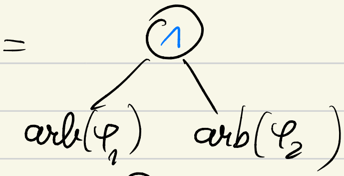
$arb : \mathbb{LP}1 \rightarrow Trees$

$arb : \mathcal{T} \rightarrow Trees$

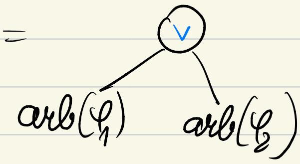
2. $\varphi = \neg \varphi_1$ atunci $arb(\varphi) =$



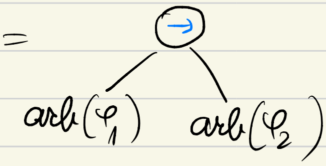
3. $\varphi = (\varphi_1 \wedge \varphi_2)$, atunci $arb(\varphi) =$



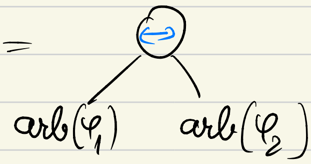
4. $\varphi = (\varphi_1 \vee \varphi_2)$ atunci $arb(\varphi) =$



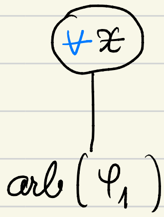
5. $\varphi = (\varphi_1 \rightarrow \varphi_2)$, $arb(\varphi) =$



6. $\varphi = (\varphi_1 \leftrightarrow \varphi_2)$, $arb(\varphi) =$



7. $\varphi = (\forall x . \varphi_1)$, $arb(\varphi) =$



8. $\varphi = (\exists x . \varphi_1)$, $arb(\varphi) =$

Prioritățile operatorilor : $\bot \quad \neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow \quad \forall \quad \exists$

Ex

$$\forall x . P(x, a) \rightarrow \exists y . \neg Q(x, i(y)) \wedge P(a, b)$$

$$(\forall x . (P(x, a) \rightarrow (\exists y . (\neg Q(x, i(y)) \wedge P(a, b)))))$$

Modelarea în LP1 din limbaj natural.

Orice om este muritor.

$\forall$

Muritor(x) : x este muritor

Om(x) : x este om

$$\forall x.\ Om(x) \rightarrow Muritor(x)$$

x poate fi orice element din domeniu (nu doar om)

Orice număr natural este egal cu 0 sau strict mai mare ca 0.

Nat(t) : t este nr. nat.

$t_1 = t_2$: t_1 este egal cu t_2

$t_1 > t_2$: t_1 este strict mai mare ca t_2

$$\Sigma = (\{Nat, =, >\}, \{0\})$$

$$\left(\forall x.\left(Nat(x) \rightarrow \left(\underbrace{=(x, 0)}_{x = 0} \vee \underbrace{>(x, 0)}_{x > 0}\right)\right)\right)$$