Curs 6

2024-2025

Programare Logică şi Funcţională

Cuprins

Semantica programelor Prolog

Mulţimi parţial ordonate

- O mulţime parţial ordonată (mpo) este o pereche $(M, \leq)$ unde $\leq \subseteq M \times M$ este o relaţie de ordine.
- O mpo $(L, \leq)$ se numeşte lanţ dacă este total ordonată, adică $x \leq y$ sau $y \leq x$ pentru orice $x, y \in L$. Vom considera lanţuri numărabile:

$$x_1 \leq x_2 \leq x_3 \leq \dots$$

- O mpo $(C, \leq)$ este completă (CPO) dacă:
  - C are prim element $\perp$ ($\perp \leq x$ oricare $x \in C$),
  - $\bigvee_n x_n$ există pentru orice lanţ $x_1 \leq x_2 \leq x_3 \leq \dots$

Teorema de punct fix

- Fie $(A, \leq_A)$ şi $(B, \leq_B)$ mulţimi parţial ordonate complete. O funcţie $f : A \to B$ este continuă dacă $f(\bigvee_n a_n) = \bigvee_n f(a_n)$ pentru orice lanţ $\{a_n\}_n$ din A.
- Un element $a \in C$ este punct fix al unei funcţii $f : C \to C$ dacă $f(a) = a$.

Teorema Knaster-Tarski pentru CPO

Fie $(C, \leq)$ o mulţime parţial ordonată completă şi $\mathbf{F} : C \to C$ o funcţie continuă. Atunci

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

este cel mai mic punct fix al funcţiei **F**.

- Secvenţa $\mathbf{F}^0(\perp) = \perp \leq \mathbf{F}(\perp) \leq \mathbf{F}^2(\perp) \leq \cdots \leq \mathbf{F}^n(\perp) \leq \cdots$ este un lanţ, deci $\bigvee_n \mathbf{F}^n(\perp)$ există.

Programe Prolog propoziţionale

Fie KB o multime de clauze definite propoziţionale şi fie At mulţimea variabilelor propoziţionale (atomilor) $p_1, p_2, \ldots$ care apar în KB.

Fie $AtFact = \{p_i \mid p_i \in KB\}$ mulţimea faptelor din KB.

Exemplu

$$\begin{array}{rcl}
\texttt{oslo} & \rightarrow & \texttt{windy} \\
\texttt{oslo} & \rightarrow & \texttt{norway} \\
\texttt{norway} & \rightarrow & \texttt{cold} \\
\texttt{cold} \land \texttt{windy} & \rightarrow & \texttt{winterIsComing} \\
 & & \texttt{oslo}
\end{array}$$

$At = \{oslo, windy, norway, cold, winterIsComing\}$

$AtFact = \{oslo\}$

Programelor Prolog propoziţionale

Fie KB o multime de clauze definite propoziţionale şi fie At mulţimea atomilor $p_1, p_2, \ldots$ care apar în KB.

Fie $AtFact = \{p_i \mid p_i \in KB\}$ mulţimea atomilor care apar în faptele din KB.

Definim funcţia $f_{KB} : \mathcal{P}(At) \to \mathcal{P}(At)$ prin

$$\begin{aligned} f_{KB}(Y) = Y \cup AtFact \\ \cup \{a \in At \mid (s_1 \wedge \ldots \wedge s_n \to a) \text{ este în } KB, \\ s_1 \in Y, \ldots, s_n \in Y\} \end{aligned}$$

- Funcţia f_{KB} este continuă.

Semantica programelor Prolog propoziţionale

Pentru funcţia continuă $f_{KB} : \mathcal{P}(At) \to \mathcal{P}(At)$

$$\boxed{\begin{aligned} f_{KB}(Y) = {} & Y \cup AtFact \\ & \cup \{a \in At \mid (s_1 \wedge \ldots \wedge s_n \to a) \text{ este în } KB, \\ & \qquad s_1 \in Y, \ldots, s_n \in Y\} \end{aligned}}$$

aplicând Teorema Knaster-Tarski pentru CPO, obţinem că

$$\bigcup_n f_{KB}^n(\emptyset)$$

este cel mai mic punct fix al lui f_{KB}.

Semantica programelor Prolog propoziţionale

- Analizaţi ce se întamplă când considerăm succesiv

$$\emptyset, \quad f_{KB}(\emptyset), \quad f_{KB}(f_{KB}(\emptyset)), \quad f_{KB}(f_{KB}(f_{KB}(\emptyset))), \ldots$$

  La fiecare aplicare a lui f_{KB}, rezultatul fie se măreşte, fie rămâne neschimbat.

- Să presupunem că în $\mathcal{S}$ avem k atomi. Atunci după $k+1$ aplicări ale lui f_{KB}, trebuie să existe un punct în şirul de mulţimi obţinute de unde o nouă aplicare a lui f_{KB} nu mai schimbă rezultatul (punct fix):

$$f_{KB}(X) = X$$

- Dacă aplicăm f_{KB} succesiv până găsim un X cu proprietatea $f_{KB}(X) = X$, atunci găsim cel mai mic punct fix al lui f_{KB}.

Cel mai mic punct fix

Exemplu

$$\begin{array}{rcl} cold & \rightarrow & wet \\ wet \wedge cold & \rightarrow & scotland \end{array}$$

$$\boxed{\begin{array}{l} f_{KB}(Y) = Y \cup AtFact \\ \cup\{a \in At \mid (s_1 \wedge \ldots \wedge s_n \rightarrow a) \text{ este în } KB, \\ s_1 \in Y, \ldots, s_n \in Y\} \end{array}}$$

Se observă că $f_{KB}(\emptyset) = \emptyset$, deci $\emptyset$ este cel mai mic punct fix.

De aici deducem că niciun atom nu este consecinţă logică a formulelor de mai sus.

Cel mai mic punct fix

Exemplu

$$\begin{array}{rcl} cold & & \\ cold & \rightarrow & wet \\ windy & \rightarrow & dry \\ wet \wedge cold & \rightarrow & scotland \end{array}$$

$f_{KB}(Y) = Y \cup AtFact$
$\cup\{a \in At \mid (s_1 \wedge \ldots \wedge s_n \rightarrow a)$ este în S,
$s_1 \in Y, \ldots, s_n \in Y\}$

$$\begin{array}{rcl} f_{KB}(\emptyset) & = & \{\, cold \,\} \\ f_{KB}(\{\, cold \,\}) & = & \{\, cold, wet \,\} \\ f_{KB}(\{\, cold, wet \,\}) & = & \{\, cold, wet, scotland \,\} \\ f_{KB}(\{\, cold, wet, scotland \,\}) & = & \{\, cold, wet, scotland \,\} \end{array}$$

Deci cel mai mic punct fix este $\{\, cold, wet, scotland \,\}$.

Semantica programelor Prolog propoziţionale

Funcţia $f_{KB} : \mathcal{P}(At) \to \mathcal{P}(At)$ este definită prin

$$\boxed{\begin{aligned} f_{KB}(Y) = & \ Y \cup AtFact \\ & \cup\{a \in At \mid (s_1 \wedge \ldots \wedge s_n \to a) \text{ este în } KB, s_1 \in Y, \ldots, s_n \in Y\} \end{aligned}}$$

unde At este mulţimea atomilor din $\mathcal{S}$ şi $Baza = \{p_i \mid p_i \in KB\}$ este mulţimea atomilor care apar în faptele din $\mathcal{S}$.

Teoremă

Fie FP_{KB} este cel mai mic punct fix al funcţiei f_{KB}. Atunci

$$q \in FP_{KB} \quad \text{dacă şi numai dacă} \quad KB \models q.$$

- Semantica programului KB este FP_{KB}, cel mai mic punct fix al funcţiei f_{KB}.
- FP_{KB} conţine toate consecinţele logice ale bazei de cunoştinţe KB.

Limbajul PROLOG / Logica Horn

- Logica Horn: un fragment al logicii de ordinul I în care singurele formule admise sunt clauze Horn
  - formule atomice: $P(t_1, \ldots, t_n)$
  - $Q_1 \wedge \ldots \wedge Q_n \to P$
    unde toate Q_i, P sunt formule atomice, $\top$ sau $\bot$

- Problema programării logice: reprezentăm cunoştinţele ca o mulţime de clauze definite KB şi suntem interesaţi să aflăm răspunsul la o întrebare de forma $Q_1 \wedge \ldots \wedge Q_n$, unde toate Q_i sunt formule atomice

$$KB \models Q_1 \wedge \ldots \wedge Q_n$$

  - Variabilele din KB sunt cuantificate universal.
  - Variabilele din $Q_1, \ldots, Q_n$ sunt cuantificate existenţial.
- Semantica de punct fix a unui program PROLOG este definită folosind modele Herbrand (caz particular de modele ale logicii de ordinul I).

Modele în logica de ordinul I

Logica de ordinul I - sintaxa

Limbaj de ordinul I $\mathcal{L}$

- unic determinat de $\tau = (\mathbf{R}, \mathbf{F}, \mathbf{C}, ari)$

Termenii lui $\mathcal{L}$, notaţi $Trm_{\mathcal{L}}$, sunt definiţi inductiv astfel:

- orice variabilă este un termen;
- orice simbol de constantă este un termen;
- dacă $f \in \mathbf{F}$, $ar(f) = n$ şi $t_1, \ldots, t_n$ sunt termeni, atunci $f(t_1, \ldots, t_n)$ este termen.

Formulele atomice ale lui $\mathcal{L}$ sunt definite astfel:

- dacă $R \in \mathbf{R}$, $ar(R) = n$ şi $t_1, \ldots, t_n$ sunt termeni, atunci $R(t_1, \ldots, t_n)$ este formulă atomică.

Formulele lui $\mathcal{L}$ sunt definite astfel:

- orice formulă atomică este o formulă
- dacă φ este o formulă, atunci $\neg\varphi$ este o formulă
- dacă φ şi ψ sunt formule, atunci $\varphi \vee \psi$, $\varphi \wedge \psi$, $\varphi \rightarrow \psi$ sunt formule
- dacă φ este o formulă şi x este o variabilă, atunci $\forall x\, \varphi$, $\exists x\, \varphi$ sunt formule

Logica de ordinul I - semantică

O structură este de forma $\mathcal{A} = (A, \mathbf{F}^{\mathcal{A}}, \mathbf{R}^{\mathcal{A}}, \mathbf{C}^{\mathcal{A}})$, unde

- A este o mulţime nevidă
- $\mathbf{F}^{\mathcal{A}} = \{f^{\mathcal{A}} \mid f \in \mathbf{F}\}$ este o mulţime de operaţii pe A;
  dacă f are aritatea n, atunci $f^{\mathcal{A}} : A^n \to A$.
- $\mathbf{R}^{\mathcal{A}} = \{R^{\mathcal{A}} \mid R \in \mathbf{R}\}$ este o mulţime de relaţii pe A;
  dacă R are aritatea n, atunci $R^{\mathcal{A}} \subseteq A^n$.
- $\mathbf{C}^{\mathcal{A}} = \{c^{\mathcal{A}} \in A \mid c \in \mathbf{C}\}$.

O interpretare a variabilelor lui $\mathcal{L}$ în $\mathcal{A}$ ($\mathcal{A}$-interpretare) este o funcţie $I : V \to A$.

Inductiv, definim interpretarea termenului t în $\mathcal{A}$ sub I notat $t_I^{\mathcal{A}}$.

Inductiv, definim când o formulă este adevărată în $\mathcal{A}$ în interpretarea I notat $\mathcal{A}, I \models \varphi$. În acest caz spunem că $(\mathcal{A}, I)$ este model pentru φ.

O formulă φ este adevărată într-o structură $\mathcal{A}$, notat $\mathcal{A} \models \varphi$, dacă este adevărată în $\mathcal{A}$ sub orice interpretare. Spunem că $\mathcal{A}$ este model al lui φ.

O formulă φ este adevărată în logica de ordinul I, notat $\models \varphi$, dacă este adevărată în orice structură. O formulă φ este validă dacă $\models \varphi$.

O formulă φ este satisfiabilă dacă există o structură $\mathcal{A}$ şi o $\mathcal{A}$-interpretare I astfel încât $\mathcal{A}, I \models \varphi$.

Interpretare

Fie $\mathcal{L}$ un limbaj de ordinul I şi $\mathcal{A}$ o ($\mathcal{L}$-)structură.

Definiţie

O interpretare a variabilelor lui $\mathcal{L}$ în $\mathcal{A}$ este o funcţie

$$I : V \to A.$$

Definiţie

Inductiv, definim interpretarea termenului t în $\mathcal{A}$ sub I ($t_I^{\mathcal{A}}$) prin:

- dacă $t = x_i \in V$, atunci $t_I^{\mathcal{A}} := I(x_i)$
- dacă $t = c \in \mathbf{C}$, atunci $t_I^{\mathcal{A}} := c^{\mathcal{A}}$
- dacă $t = f(t_1, \ldots, t_n)$, atunci $t_I^{\mathcal{A}} := f^{\mathcal{A}}((t_1)_I^{\mathcal{A}}, \ldots, (t_n)_I^{\mathcal{A}})$

Interpretare

Definim inductiv faptul că o formulă este adevărată în $\mathcal{A}$ sub interpretarea I astfel:

- $\mathcal{A}, I \models P(t_1, \ldots, t_n)$ dacă $P^{\mathcal{A}}((t_1)_I^{\mathcal{A}}, \ldots, (t_n)_I^{\mathcal{A}})$
- $\mathcal{A}, I \models \neg\varphi$ dacă $\mathcal{A}, I \not\models \varphi$
- $\mathcal{A}, I \models \varphi \vee \psi$ dacă $\mathcal{A}, I \models \varphi$ sau $\mathcal{A}, I \models \psi$
- $\mathcal{A}, I \models \varphi \wedge \psi$ dacă $\mathcal{A}, I \models \varphi$ şi $\mathcal{A}, I \models \psi$
- $\mathcal{A}, I \models \varphi \rightarrow \psi$ dacă $\mathcal{A}, I \not\models \varphi$ sau $\mathcal{A}, I \models \psi$
- $\mathcal{A}, I \models \forall x\, \varphi$ dacă pentru orice $a \in A$ avem $\mathcal{A}, I_{x \leftarrow a} \models \varphi$
- $\mathcal{A}, I \models \exists x\, \varphi$ dacă există $a \in A$ astfel încât $\mathcal{A}, I_{x \leftarrow a} \models \varphi$

unde pentru orice $a \in A$, $I_{x \leftarrow a}(y) = \begin{cases} I(y) & \text{dacă } y \neq x \\ a & \text{dacă } y = x \end{cases}$

Model în logica de ordinul I

Exemplu

Fie limbajul $\mathcal{L}$ cu $\mathbf{F} = \{s\}$, $\mathbf{R} = \{P\}$, $\mathbf{C} = \{0\}$ cu $ari(s) = ari(P) = 1$.

Fie structura $\mathcal{N} = (\mathbb{N}, s^{\mathcal{N}}, P^{\mathcal{N}}, 0^{\mathcal{N}})$ unde $0^{\mathcal{N}} := 1$ şi

- $s^{\mathcal{N}} : \mathbb{N} \to \mathbb{N}$, $s^{\mathcal{N}}(n) := n^2$
- $P^{\mathcal{N}} \subset \mathbb{N}$, $P^{\mathcal{N}} = \{n \mid n \text{ este impar }\}$

Demonstraţi că $\mathcal{N} \models \forall x\,(P(x) \to P(s(x)))$.

Fie $I : V \to \mathbb{N}$ o interpretare. Observăm că
$\mathcal{N}, I \models P(x)$ dacă $P^{\mathcal{N}}(I(x))$, adică $\mathcal{N}, I \models P(x)$ dacă $I(x)$ este impar.

$\mathcal{N}, I \models \forall x\,(P(x) \to P(s(x)))$ dacă
$\mathcal{N}, I_{x\leftarrow n} \models P(x) \to P(s(x))$ oricare $n \in N$
$\mathcal{N}, I_{x\leftarrow n} \not\models P(x)$ sau $\mathcal{N}, I_{x\leftarrow n} \models P(s(x))$ oricare $n \in N$
$I_{x\leftarrow n}(x)$ nu este impar sau $I_{x\leftarrow n}(s(x))$ este impar oricare $n \in \mathbb{N}$
n este par sau n^2 este impar oricare $n \in \mathbb{N}$

ceea ce este întodeauna adevărat.

Modele Herbrand

Modele Herbrand

Fie $\mathcal{L}$ un limbaj de ordinul I.

- Presupunem că are cel puţin un simbol de constantă!
- Dacă nu are, adăugăm un simbol de constantă.

Universul Herbrand este mulţimea $T_{\mathcal{L}}$ a tututor termenilor lui $\mathcal{L}$ fără variabile.

Un model Herbrand este o structură $\mathcal{H} = (T_{\mathcal{L}}, \mathbf{F}^{\mathcal{H}}, \mathbf{P}^{\mathcal{H}}, \mathbf{C}^{\mathcal{H}})$, unde

- pentru orice simbol de constantă c, $c^{\mathcal{H}} = c$
- pentru orice simbol de funcţie f de aritate n,
  $f^{\mathcal{H}}(t_1, \ldots, t_n) = f(t_1, \ldots, t_n)$
- pentru orice simbol de relaţie R de aritate n, $R^{\mathcal{H}}(t_1, \ldots, t_n) \subseteq (T_{\mathcal{L}})^n$

Pentru a defini un model Herbrand concret trebuie sa definim interpretarea relaţiilor.

Model Herbrand

Exemplu

Fie $\mathcal{L}$ un limbaj de ordinul I cu un simbol de funcţie f de aritate 1, un simbol de constantă a şi un simbol de relaţie R de aritate 2.

O structură Herbrand $\mathcal{H} = (T_{\mathcal{L}}, \mathbf{F}^{\mathcal{H}}, \mathbf{R}^{\mathcal{H}}, \mathbf{C}^{\mathcal{H}})$ unde

- $T_{\mathcal{L}} = \{a, f(a), f(f(a)), \ldots\}$
- $a^{\mathcal{H}} = a \in T_{\mathcal{L}}$
- $f^{T}_{\mathcal{H}}(t) = f(t)$
- $R^{\mathcal{H}} = \{(t,t) \mid t \in T_{\mathcal{L}}\} = \{(a,a), (f(a), f(a)), (f(f(a)), f(f(a))), \ldots\}$

$\mathcal{H} \models \forall x\, R(x,x)$.

Model Herbrand

Exemplu

Fie $\mathcal{L}$ un limbaj de ordinul I cu un simbol de funcţie f de aritate 1, un simbol de constantă a şi un simbol de relaţie R de aritate 2.

O structură Herbrand $\mathcal{H} = (T_{\mathcal{L}}, \mathbf{F}^{\mathcal{H}}, \mathbf{R}^{\mathcal{H}}, \mathbf{C}^{\mathcal{H}})$ unde

- $T_{\mathcal{L}} = \{a, f(a), f(f(a)), \ldots\}$
- $a^{\mathcal{H}} = a \in T_{\mathcal{L}}$
- $f^{T}_{\mathcal{H}}(t) = f(t)$
- $R^{\mathcal{H}} = \{(t, f(t)) \mid t \in T_{\mathcal{L}}\} = \{(a, f(a)), (f(a), f(f(a))), (f(f(a)), f(f(f(a)))), \ldots\}$

$\mathcal{H} \not\models \forall x\, R(x, x)$.

Cel mai mic model Herbrand

- Definim o ordine între modelele Herbrand:

  $\mathcal{H}_1 \leq \mathcal{H}_2$ dacă şi numai dacă

  pentru orice $R \in \mathbf{R}$ cu $ari(R) = n$ şi pentru orice termeni $t_1, \ldots, t_n$
  dacă $\mathcal{H}_1 \models R(t_1, \ldots, t_n)$, atunci $\mathcal{H}_2 \models R(t_1, \ldots, t_n)$

- Pentru KB, un program Prolog (o mulţime de clauze definite), definim intersecţia modelelor Herbrand ale programului

$$\mathcal{LH}_{KB} := \bigcap \{\mathcal{H} \mid \mathcal{H} \models KB\}$$

  Se observă că $\mathcal{LH}_{KB} \models KB$.

Semantica unui program logic definit KB este dată de cel mai mic model Herbrand al lui KB. Vom caracteriza cel mai mic model Herbrand $\mathcal{LH}_{KB}$ printr-o construcţie de punct fix.

Baza Herbrand

- O formulă fără variabile se numeşte închisă.
- Baza Herbrand $B_{\mathcal{L}}$ este mulţimea formulelor atomice închise.
- O instanţă închisă a unei clauze $Q_1 \wedge \ldots \wedge Q_n \to P$ este rezultatul obţinut prin înlocuirea variabilelor cu termeni fără variabile.

Exemplu

- Fie $\mathcal{L}$ un limbaj cu un simbol de constantă 0, un simbol de funcţie unară s şi un simbol de relaţie unară par. Notăm $s(x)$ cu sx.
- $T_{\mathcal{L}} = \{0, s0, ss0, \ldots\}$
- Fie KB mulţimea clauzelor:

$$par(0) \qquad\qquad par(x) \to par(ssx)$$

- Instanţe de bază:
  - $par(0) \to par(ss0)$
  - $par(s0) \to par(sss0)$
- $B_{\mathcal{L}} = \{par(0), par(s0), par(ss0), par(sss0) \ldots\}$

Baza Herbrand

- O formulă fără variabile se numeşte închisă.
- Baza Herbrand $B_{\mathcal{L}}$ este mulţimea formulelor atomice închise.
- O instanţă închisă a unei clauze $Q_1 \wedge \ldots \wedge Q_n \to P$ este rezultatul obţinut prin înlocuirea variabilelor cu termeni fără variabile.
- Pentru o mulţime de clauze definite KB, dacă $P \in B_{\mathcal{L}}$ şi $X \subseteq B_{\mathcal{L}}$ spunem că

  $$oneStep_{KB}(P, X) \text{ este adevărat}$$

  dacă există $Q_1, \ldots, Q_n \in X$ astfel încât $Q_1 \wedge \ldots \wedge Q_n \to P$ este o instanţă de închisă a unei clauze din KB.
- Pentru o mulţime de clauze definite KB, definim

  $$f_{KB} : \mathcal{P}(B_{\mathcal{L}}) \to \mathcal{P}(B_{\mathcal{L}})$$

  $$f_{KB}(X) = \{P \in B_{\mathcal{L}} \mid oneStep_{KB}(P, X)\}$$
- f_{KB} este continuă (exerciţiu).

Baza Herbrand

Exemplu

- Fie $\mathcal{L}$ un limbaj cu un simbol de constantă 0, un simbol de funcţie unară s şi un simbol de relaţie unară par. Notăm $s(x)$ cu sx.
- $T_{\mathcal{L}} = \{0, s0, ss0, \ldots\}$
- Fie KB mulţimea clauzelor:

$$par(0) \qquad\qquad par(x) \to par(ssx)$$

- Instanţe de bază:
  - $par(0) \to par(ss0)$
  - $par(s0) \to par(sss0)$
- $B_{\mathcal{L}} = \{par(0), par(s0), par(ss0), par(sss0) \ldots\}$
- $oneStep_{KB}(P, \{\})$ selecteaza faptele din KB
  $f_{KB}(\{\}) = \{par(0)\}$
- $f_{KB}(\{par(0)\}) = \{par(0), par(ss0)\}$
- $f_{KB}(\{par(s(0))\}) = \{par(0), par(sss0)\}$
- $f_{KB}(\{par(ss0)\}) = \{par(0), par(ssss0)\}$

Cel mai mic model Herbrand

Fie KB un program logic definit.

- Din teorema Knaster-Tarski, f_{KB} are un cel mai mic punct fix FP_{KB}.
- FP_{KB} este reuniunea tuturor mulţimilor

$$f_{KB}(\{\}),\ f_{KB}(f_{KB}(\{\})),\ f_{KB}(f_{KB}(f_{KB}(\{\}))),\ldots$$

Caracterizarea $\mathcal{LH}_{KB}$ ca punct fix.

Pentru orice $R \in \mathbf{R}$ cu $ari(R) = n$ şi pentru orice $t_1, \ldots, t_n$ termeni, avem

$$(t_1, \ldots, t_n) \in R^{\mathcal{LH}_{KB}} \text{ ddacă } R(t_1, \ldots, t_n) \in FP_{KB}$$

Relaţiile care definesc cel mai mic model Herbrand al unui program Prolog sunt caracterizate folosind teorema de punct fix Knaster-Tarski.

Cel mai mic model Herbrand

Exemplu

- Fie $\mathcal{L}$ un limbaj cu un simbol de constantă 0, un simbol de funcţie unară s şi un simbol de relaţie unară par. Notăm $s(x)$ cu sx.
- $T_{\mathcal{L}} = \{0, s0, ss0, \ldots\}$
- Fie KB mulţimea clauzelor:

$$par(0) \qquad\qquad par(x) \to par(ssx)$$

- Instanţe de bază:
  - $par(0) \to par(ss0)$
  - $par(s0) \to par(sss0)$
- $B_{\mathcal{L}} = \{par(0), par(s0), par(ss0), par(sss0) \ldots\}$
- $FP_{KB} = \bigcup_n f^n_{KB}(\emptyset) = f_{KB}(\{\}) \cup \ f_{KB}(f_{KB}(\{\})) \cup \ f_{KB}(f_{KB}(f_{KB}(\{\}))) \cup \ldots$
  $= \{par(0), par(ss0), par(ssss0), \ldots\}$

Observăm ca relaţia $par^{\mathcal{LH}_{KB}}$ conţine numai reprezentările numerelor pare.

Cel mai mic model Herbrand

Fie KB un program logic definit.

Teoremă

Pentru orice formulă atomică Q,

$$KB \models Q \quad \text{dacă şi numai dacă} \quad \mathcal{LH}_{KB} \models Q.$$

- Semantica programului KB este $\mathcal{LH}_{KB}$, cel mai mic model Herbrand.
- Relaţiile modelului $\mathcal{LH}_{KB}$ sunt definite folosind teorema de punct fix Knaster-Tarski.

Cel mai mic model Herbrand

Fie KB un program logic definit.

Teoremă

Pentru orice formulă atomică Q,

$$KB \models Q \quad \text{dacă şi numai dacă} \quad \mathcal{LH}_{KB} \models Q.$$

Demonstraţie

Q este adevărată în $\mathcal{LH}_{KB}$
ddacă Q este adevărată în toate modelele Herbrand ale lui KB
ddacă $\neg Q$ este falsă în toate modelele Herbrand ale lui KB
ddacă $KB \cup \{\neg Q\}$ nu are niciun model Herbrand (1)
ddacă $KB \cup \{\neg Q\}$ nesatisfiabilă (2)
ddacă $KB \models Q$

(1) $\Leftrightarrow$ (2) rezultă din Teorema lui Herbrand