Logično programiranje

22. 4. 2020

- Ukazno programiranje: izvajanje = zaporedje ukazov, ki spreminjajo stanje
- Deklarativno programiranje: izvajanje = predelava izraza v vrednost (končna oblika)
- Logično programiranje: izvajanje = iskanje dokaza (prolog)
  λ-prolog

Hornove formule

Logika 1. reda:

- konstanti $\bot$, $\top$
- vezniki
  - $\wedge$ in (konjunkcija)
  - $\vee$ ali (disjunkcija)
  - $\Rightarrow$ implikacija
  - $\neg$ negacija
- kvantifikatorja
  - $\forall$ za vsak (univerzalni)
  - $\exists$ obstaja (eksistenčni)
- relacije & predikati:
  - $a = b$ enako
  - $a < b$ manjše
  - $p \parallel q$ vzporedno
  - $x \in A$ je element

Pišemo:

$$p \wedge q \Rightarrow (\neg p \Rightarrow q)$$

$$\forall x.\ x^2 + 1 < x^3 + 4x$$

"za vsak x velja . . . . . ."

$$\forall x. \forall y.\ x^2 + y < y^3 + x^4$$

$$\forall x, y.\ x^2 + y < y^3 + x^4$$

$$\exists x.\ x^2 + 1 = 0$$

$$\forall x. (x \geq 0 \Rightarrow \exists y.\ x = y^2)$$

"Vsako nenegativno število ima kvadratni koren."

Hornova formula

$$\forall x_1, \dots, x_m. (\phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_n \Rightarrow \psi)$$

kjer so $\phi_1, \dots, \phi_n, \psi$ osnovne formule, t.j. oblike

$$p(\underbrace{t_1, \dots, t_i}_{\text{termi}})$$

relacijski simbol

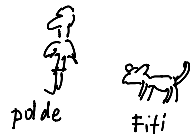
termi → izrazi sestavljeni iz spremenljivk, konstant in funkcijskih simbolov

Posebni primeri:

① $n=0$: $\forall x_1, \dots, x_m . \Psi$ dejstvo

② $m=0$: $\phi_1 \wedge \dots \wedge \phi_n \Rightarrow \Psi$

Primer: $\forall a. \,(\text{pes}(a) \Rightarrow \text{zival}(a))$



zival(polde) ?

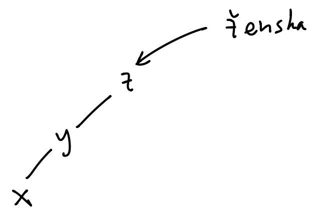
Vstavimo $a := \text{polde}$:

$\underset{\bot}{\text{pes}(\text{polde})} \Rightarrow \text{zival}(\text{polde})$

$\underset{\top}{\text{pes}(\text{Fifi})} \Rightarrow \underset{\top}{\text{zival}(\text{Fifi})}$

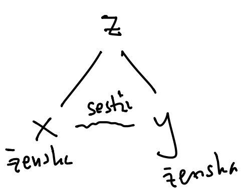
Primer:

∀ x y z . (otrok(x, y) ∧ otrok(y, z) ∧ zenska(z) ⇒ babica(x, z))



Primer:

∀ x y z . otrok(x, z) ∧ otrok(y, z) ∧ zenska(x) ∧ zenska(y) ⇒ sestra(x, y)



Problem: $x = y$?

$x=y$ ali
x in y sta sestri ali
x in y sta polsestri

Primer: števila

$\forall n. \; n + 0 = n$

$\forall k, m. \; k + \text{succ}(m) = \text{succ}(k+m)$

Def: Relacija R predstavlja funkcijo f, če velja

$$f(x) = y \iff R(x,y)$$

Logično programiranje: funkcije predstavimo z relacijami

Primer: $f : \mathbb{R} \to \mathbb{R}$ $\quad f(x) = x^2 - 2$

$R(x,y) \iff y = x^2 - 2 \iff 2 + y = x^2$

Primer: obratno? Kdaj relacija R predstavlja funkcijo?

Kadar je funkcijska: za vsak x obstaja natanko en y, da je $R(x,y)$.

Primer: $R(x,y) \iff x = y^2$ $\qquad -1 = y^2$? Ni y.

$\pm\sqrt{x}$ $\qquad 4 = y^2$? Dva $y = +2$, $y = -2$

Vsota kot relacija:

operacija $x + y$

relacija $plus(x,y,z)$ "z je vsota x in y"

$\forall n.\ n + 0 = n \rightsquigarrow \boxed{\forall n.\ plus(n,0,n)}$

$\forall k,m.\ k + succ(m) = succ(k+m) \rightsquigarrow$

$\forall k,m.\ plus(k, succ(m), succ(\underbrace{k+m}_{j}))$

$\rightsquigarrow \boxed{\forall k,m,j.\ plus(k,m,j) \Rightarrow plus(k, succ(m), succ(j))}$

4 $\qquad$ |||| $\qquad$ 100_2

Primitivni funkcijski simboli: $succ$ $\quad$ $\overset{0}{zero}$

$succ(succ(0))$

Kako iščemo dokaz?

1. $X \land Y \Rightarrow C$
2. $A \land B \Rightarrow C$
3. $X \Rightarrow B$
4. $A \Rightarrow B$
5. A

Ali iz teh $\overset{1-5}{\text{formul}}$ smemo sklepati C?

Naloga: C ✓

(1) X, Y

(2) A ✓, B ✓

(5)

(3) X

(4) A ✓

(5)

Primer:

1. ∀ x y . otrok(x, y) ⇒ mlajsi(x, y)
2. otrok(miha, mojca)

? mlajsi(miha, mojca)

↳ 1 ✓ x = miha, y = mojca ⟶ otrok(miha, mojca)
↳ 2 ✓

Primer

1. sodo(zero)
2. ∀ x . sodo(x) ⇒ liho(succ(x))
3. ∀ y . liho(y) ⇒ sodo(succ(y))

? sodo(succ(succ(zero)))

↳ 3 y = succ(zero) ⇝ liho(succ(zero))
↳ 2 x = zero ⇝ sodo(zero)
↳ 1 ✓

Primer s postopkom združevanja:

1. sodo(zero)
2. ∀ x . sodo(x) ⇒ liho(succ(x))
3. ∀ y . liho(y) ⇒ sodo(succ(y))

? ∃ z . liho(z) odgovor z = succ(zero)

↳ 1 liho(z) = sodo(zero) NI REŠITVE
↳ 2 liho(z) = liho(succ(x))
z = succ(x)
⇝ sodo(x)
↳ 1 sodo(x) = sodo(zero)
x = zero

1. ∀ x . sodo(x) ⇒ liho(succ(x))
2. ∀ y . liho(y) ⇒ sodo(succ(y))
3. sodo(zero)

SE ZACIKLA

? ∃ z . liho(z)

↳ 1 $z = succ(x_1)$ ⇝ $sodo(x_1)$
↳ 2 $sodo(succ(y_1)) = sodo(x_1)$ ⇝ $liho(y_1)$
$x_1 = succ(y_1)$
↳ 1 $y_1 = succ(x_2)$ ⇝ $sodo(x_2)$
↳ 2 $x_2 = succ(y_2)$ ....

join(cons(A, X), Y, cons(A, Z)) :- cons(X, Y, Z)