Logično programiranje

program = logična pravila $P_1, \dots, P_n$

zagon = poizvedba / vprašanje Q

izvajanje = iskanje dokaza $P_1 \wedge \dots \wedge P_n \Rightarrow Q$

Logične formule:

- izjavni račun:
  - osnovne/primitivne formule $(p_1, p_2, \dots)$
  - logični konstanti: $\bot$ neresnica, $\top$ resnica
  - vezniki: $\wedge$, $\vee$, $\Rightarrow$, $\neg$, $\Leftrightarrow$
- predikatni račun
  - osnovni predikati in relacije ($<$, $=$, "deli", "- je mati od -")
  - termi, ki označujejo neke objekte/individuume/entite (števila, točke, premice, stanja stroja/pomnilnika, ljudje, ...) - podatki
  - kvantifikatorja:

    $\forall$ univerzalni ("za vse")

    $\exists$ eksistenčni ("obstaja")

Vse skupaj: logika prvega reda

~~Hoareova~~ Horn logika

$\Rightarrow \forall x_1, \dots, x_n . (P_1 \wedge P_2 \wedge \dots \wedge P_m \Rightarrow Q)$

Za vse $x_1, x_2, \dots x_n$, če velja P_1 in P_2 in ... in P_m, potem Q.

$\Rightarrow$ $P_1, P_2, \dots, P_n, Q$ osnovni predikati in relacije

Primeri:

Termi / podatki:

- spremenljivke x, y, oseba$_1$, stanje
- konstante $0, \pi$, princHarry, začetnoStanje
- funkcijski simboli $+, -$, succ, višina(...), naslednjeStanje(...)

Posebni primeri: $\forall x_1, \dots, x_n . (P_1 \wedge P_2 \wedge \dots \wedge P_m \Rightarrow Q)$

① $m = 0$: $\forall x_1, \dots, x_n . Q$ aksiom / dejstvo

Primer: $\forall x . \underbrace{x = x}_{Q}$

② $n = 0$: $P_1 \wedge \dots \wedge P_m \Rightarrow Q$ ni spremenljivk

Primer: $less(4,3) \wedge less(2,3) \Rightarrow less(1,5$

Kako s Hornovimi formulami podamo funkcije?

Seštevanje: $\forall x.\ 0 + x = x$

$\forall x, y.\ succ(x) + y = succ(\underbrace{x+y}_{z})$ } aksiomi za $+$

Tehnika: seštevanje $+$ nadomestimo z relacijo

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

"Peanova Aksioma" kot Hornovi formuli:

$\forall x.\ plus(0, x, x)$

$\forall x, y, z.\ plus(x, y, z) \Rightarrow plus(succ(x), y, succ(z))$

v on-line zapiskih: vsota

Funkcija $f : X \times Y \to Z$

Relacijo $R(x, y, z) \Leftrightarrow f(x, y) = z$

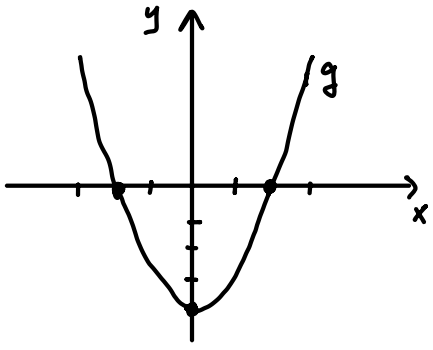
↳ Graf funkcije f

Zakaj "graf"? $g : \mathbb{R} \to \mathbb{R}$

$g(t) := t^2 - 4$

$R(x, y) \Leftrightarrow g(x) = y$

$x^2 - 4 = y$



Formule, ki jih lahko pretvorimo v Hornovo obliko:

$P_1 \vee P_2 \Rightarrow Q$ je ekvivalentno $P_1 \Rightarrow Q$ in $P_2 \Rightarrow Q$

$(\exists x. P) \Rightarrow Q$ je ekvivalentno $\forall x. (P \Rightarrow Q)$

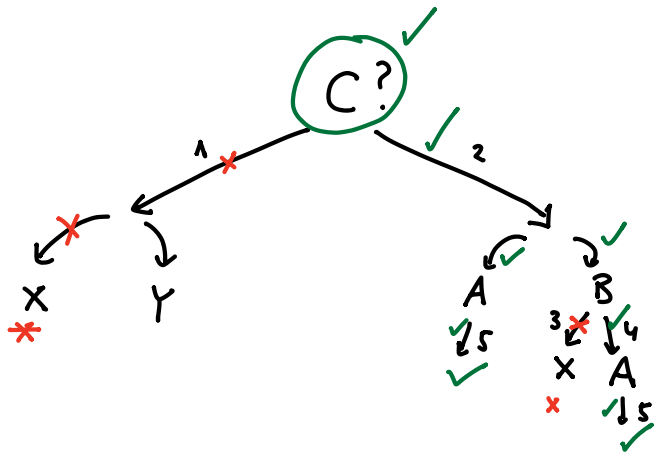
Formule, ki niso ekvivalentne Hornovim:

$\exists x. Q$

$\neg Q$

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

sledi C?



iskanje dokaza v globino

1. $\forall x\, y.\ \text{otrok}(x, y) \Rightarrow \text{mlajsi}(x, y)$
2. $\text{otrok}(\text{miha}, \text{mojca})$

mlajsi(miha, mojca)?

1 $x = \text{miha}$, $y = \text{mojca}$

otrok(miha, mojca)

2

1. $\forall x.\, \text{sodo}(x) \Rightarrow \text{liho}(\text{succ}(x))$
2. $\forall y.\, \text{liho}(y) \Rightarrow \text{sodo}(\text{succ}(y))$
3. $\text{sodo}(\text{zero})$

sodo(succ(succ(succ(zero))) ✗

$\overset{2}{\swarrow}$ $y = \text{succ}(\text{succ}(\text{zero}))$

liho(succ(succ(zero)))

$\overset{1}{\swarrow}$ $x = \text{succ}(\text{zero})$

sodo(succ(zero))

$\overset{2}{\swarrow}$ $y = \text{zero}$

liho(zero)

✗

Programiranje s Hornovimi formulami:

- program $H_1, \ldots, H_n$ Hornove formule
- poizvedba $\exists x_1, \ldots, x_k.\, G$?

$$\frac{(\eta, c_1) \mapsto (\eta', c_1')}{(\eta, (c_1 ; c_2)) \mapsto (\eta', (c_1' ; c_2))}$$

$$\forall \eta, c_1, c_2, \eta', c_1'.\ (\eta, c_1) \mapsto (\eta', c_1') \Rightarrow (\eta, (c_1; c_2)) \mapsto (\eta', (c_1'; c_2))$$

Hornova formula

OCaml	x :: y	x :: y :: z
Prolog	[x | y]	[x, y | z]