Logično programiranje

- ukazno programiranje: "kako rešimo problem"
  program = ukazi, s katerimi upravljamo stanje (pomnilnik, I/O, ..)
- deklarativno programiranje: "kaj je problem"
  → program = "sistem enačb"

  let rec fib n = ... fib = ϕ(fib)

  let search x = ...

  ? " $f(n) + g(n)^3 = n^2 + 7$ "

  ? $f(g(n)) = \ldots$

  ? $g(f(2n)) = \ldots$

→ logično programiranje

program = sistem logičnih formul

računanje / izvajanje programa = iskanje dokaza

→ SQL Standard query language

algebra relacij

Hornove formule

Logika 1. reda:

- logični konstanti: $\bot$, $\top$
- vezniki: $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$, $\neg$
- kvantifikatorja
  - $\forall$ univerzalni "za vsak"
  - $\exists$ eksistenčni "obstaja"
- primitivne relacije: $=$, $<$, vzporednost, ...

Hornove formule:

$$\forall x_1, x_2, \dots, x_i : (\phi_1 \wedge \phi_2 \wedge \dots \wedge \phi_j \Rightarrow \psi)$$

Pri čemer so $\phi_1, \dots, \phi_j$ in ψ osnovne formule, oblike

$$p(t_1, \dots, t_m)$$

p: osnovna relacija (primitivna)

$t_1, \dots, t_m$: termi (izrazi):
- spremenljivk
- osnovni funkcijski simboli

Posebna primera:

($j=0$) $\forall x_1, \dots, x_i . \psi$ dejstvo

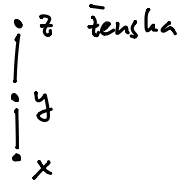
($i=0$) $\phi_1 \wedge \dots \wedge \phi_j \Rightarrow \psi$ ni spremenljivk

Primeri

$\forall\ a\ .\ (pes(a) \Rightarrow zival(a))$

$\forall\ x\ y\ z\ .\ (otrok(x, y) \wedge otrok(y, z) \wedge zenska(z) \Rightarrow babica(x, z))$

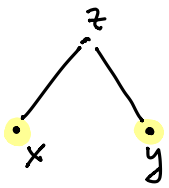
osnovna relacija



$\forall\ x\ y\ z\ .\ otrok(x, z) \wedge otrok(y, z) \wedge zenska(x) \wedge zenska(y) \Rightarrow sestra(x, y)$

Napaka:

- $x = y$?
- lahko sta polsestri



$\forall\ n\ .\ n + 0 = n$
$\forall\ k\ m\ .\ k + succ(m) = succ\ (k + m)$

R predstavlja funkcijo f, če velja $R(x,y) \iff f(x) = y$

Primer:

$R(x,y) := (y = 5)$ predstavlja $f(x) = 5$

$R(x,y) := (y \geq 0 \wedge y^2 = x)$ predstavlja $f(x) = \sqrt{x}$

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

$\forall\ n\ .\ vsota(n, zero, n)$ $\quad \forall n.\ n + zero = n$
$\forall\ k\ m\ n\ .\ vsota(k, m, n) \Rightarrow vsota(k, succ(m), succ(n))$

$\forall\ k\ m\ n\ .\ k+m = n \Rightarrow k + succ(m) = succ(n)$

$k + succ(m) = succ(k+m)$

$\phi_1 \vee \phi_2 \Rightarrow \psi$ ekvivalentno $\phi_1 \Rightarrow \psi$, $\phi_2 \Rightarrow \psi$

Program = seznam Hornovih formul

Zagon = poizvedba $\exists y_1, \ldots y_n . p(y_1, \ldots, y_n)$

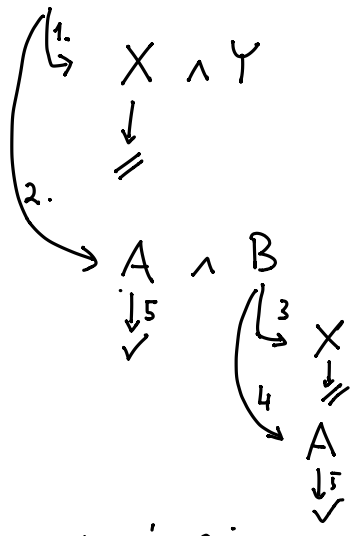
Kako iščemo dokaz?

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

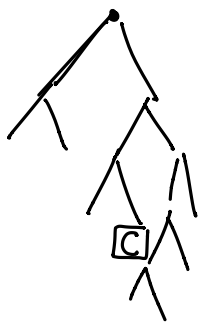
Poizvedba: C?

Backward-chaining:

- začnemo s ciljem in ga predelamo na pod-cilje

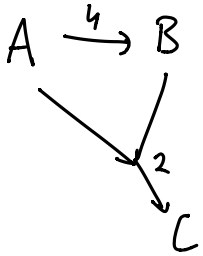


Iskanje v globino



Forward-chaining / reasoning:

- začnemo z dejstvi in s pravili generiramo nova dejstva



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

?

liho (succ (succ (succ (zero))))

(1) ↓ Katero vrednost x naj vstavimo?

t

REŠIMO ENAČBO:

liho(succ(x)) = liho (succ (succ (succ (zero))))

↓ Z ZDRUŽEVANJEM!

succ(x) = succ (succ (succ (zero)))

x = succ (succ (zero))

Ukazni jezik

$$\frac{P_1 \quad P_2 \quad \cdots \quad P_n}{Q}$$

$$\frac{\eta \mid e_1 \hookrightarrow \text{false} \quad \eta \mid e_2 \hookrightarrow v_2}{\eta \mid \text{if } e \text{ then } e_2 \text{ else } e_3 \hookrightarrow v_3}$$

$P_1 \wedge P_2 \wedge \ldots \wedge P_n \Rightarrow Q$ Hornova formul

$(\eta, C) \longmapsto (\eta', C') \longmapsto \eta'$

$[x \mapsto 3, y \mapsto 5, z \mapsto 10]$

$[(x,3), (y,5), (z,10)]$

$[x,3]$

$$\frac{\text{eval}(\eta_1, E, v) \quad \eta_1 \mid e \hookrightarrow v \qquad \text{put}(x, v, \eta_1, \eta_2) \quad \eta_2 = \eta_1[x \mapsto v]}{(\eta_1, x := e) \longmapsto \eta_2}$$

finish (let(x, E), η_1, η_2)