Izpeljava tipov (še enkrat)

Združevanje (unification)

Imamo enačbe in neznanke $\alpha, \beta, \gamma, \delta \ldots$.
(prolog: neznanke se začnejo z veliko začetnico)

Primer

$$f(\alpha) = f(g(x)) \Rightarrow \alpha = g(x)$$

Primer

① $f(\alpha, g(\beta)) = f(a, \gamma)$

② $\gamma = g(g(\alpha))$

② → Rešitev: $\gamma = g(g(\alpha))$

Vstavimo γ v ①: $f(\alpha, g(\beta)) = f(a, g(g(\alpha)))$

$\alpha = a$

$\gamma = g(g(a))$

$g(\beta) = g(g(\alpha))$

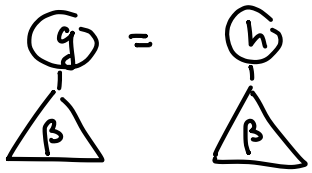
$g(\beta) = g(g(a))$

$\beta = g(a)$

Rešitev: $\alpha = a$
$\beta = g(a)$
$\gamma = g(g(a))$

Primer: $f(\alpha, g(\beta)) = f(a, h(\beta))$

$\Downarrow$



- $\alpha = a$ ✓
- $g(\beta) = h(\beta)$ NI REŠITVE

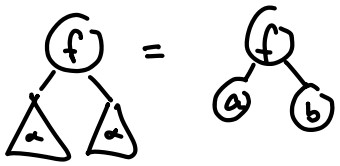
Primer: $f(\alpha, \alpha) = f(a, b)$

$\Downarrow$

Rešitev: $\alpha = a$

1. $\alpha = a$
2. ~~$\alpha = b$~~

$a = b$



$x^2 + x - 1 = 0$

$x = 1 - x^2$

Primer: $f(\alpha) = f(g(\alpha))$

$\Downarrow$

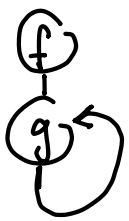
$\alpha = g(\alpha)$ α se pojavi na obeh straneh

$\alpha = g(g(\alpha))$

$\alpha = g(g(g(\alpha)))$

$\vdots$

Ni rešitve



neskončna rešitev

Postopek združevanja

Primer — $f(\alpha, \beta) = f(\alpha, \alpha)$

$\Downarrow$

~~$\alpha = \alpha$~~ $\beta = \alpha$

Rešitev: α poljuben, $\beta = \alpha$

Vprašanje: $\alpha = \alpha \vee \neg\alpha$ (rešitev: α = true)

SIMBOLA $\vee$ IN $\neg$ IMATA POMEN!

Upoštevali smo neke dodatne enačbe:

true $\vee\ \beta$ = true

$\neg$ true = false

$\neg$ false = true

$\gamma \vee \delta = \delta \vee \gamma$

⋮

Združevanje: Simboli, ki nastopajo, so neinterpretirani (o njih ne vemo nič)

α = or(α, not(α))

Primer v prog. jeziku: Izpeljava tipa

e_1 e_2

↓ ↓

τ_1 τ_2

E_1 E_2

odgovor: ψ

enačbe: $E_1, E_2, \tau_1 = \vartheta \to \psi, \tau_2 = \vartheta$

Dodamo enačbo (naj bosta ϑ in ψ novi neznanki)

$\tau_1 = \vartheta \to \psi$

$\tau_2 = \vartheta$

ϑ theta

ψ psi

Izboljšava:

e_1 e_2

↓ ↓

τ_1 τ_2

E_1 E_2

Naj bo ψ nova neznanka:

odgovor: ψ (tip aplikacije e_1 e_2)

enačbe: $E_1, E_2, \tau_1 = \tau_2 \to \psi$

if e_1 then e_2 else e_3

e_1 ↓ τ_1, E_1 — e_2 ↓ τ_2, E_2 — e_3 ↘ τ_3, E_3

tip: τ_2

enacbe:
$\tau_2 = \tau_3$
E_1, E_2, E_3
$\tau_1 = bool$

Dodamo: $\tau_1 = bool$

C/C++/Java: $e_1 ? e_2 : e_3$

Python: e_2 if e_1 else e_3

$7 + (x < 5 ? x : 10)$ ✓

if e_1 then $e_2^{\tau_2}$ else $e_3^{\tau_3}$ $\quad \tau_2 \neq \tau_3$

$\Rightarrow$ if e_1 then e_2^{σ} else e_3^{σ} $\quad \sigma = \tau_1 + \tau_2$

type σ = Foo of τ_1
| Bar of τ_2

Java

```
class BlaBla {

    public Baz f() {
        if (e1) {
            return e2;
        } else {
            return e3;
        }
    }
}
```

Class Baz

Class Foo extends Baz

Class Bar extends Baz

e_2 → class Foo

e_3 → class Bar

Rekurzija

Imamo rekurzivno definicijo

↙ x se pojavi v e

let rec x = e ;

e je funkcija

Običajno: let rec f = fun y → .... ;

let rec f y = ...

Rekurzivno: $\overset{\alpha}{x} = e$ ↳ τ, enačbe E } x ima tip α, enačbe: $E, \alpha = \tau$

① Uvedemo novo neznanko α in zabeležimo $x : \alpha$

Primer: $x = 1 :: 2 :: x$

x: α, int list

1: int

2: int

x: α

$1 :: 2 :: x$: int list

$2 :: x$: int list

$\alpha = \text{int list}$ ✓ $\alpha \mapsto \text{int list}$

int list = int list ✓

~~α~~ = int list

int list = int list ✓

Seznami:

- [] : α list (α nova neznanka)
- $e_1 :: e_2$ } tip: τ_1 list; enačbe: $E_1, E_2, \tau_2 = \tau_1$ list

(e_1: τ_1, E_1; e_2: τ_2, E_2)

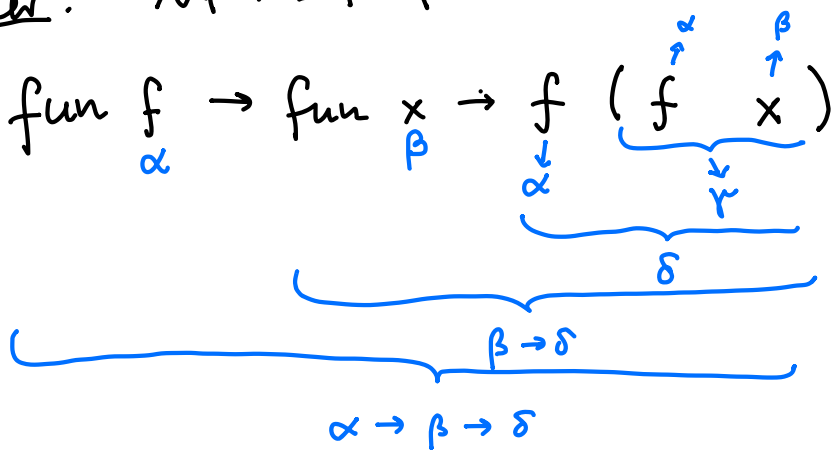
2 :: [3;4]

Primer: $x = (3, x)$ $\quad$ odgovor: α

α $\quad$ (int, α) $\quad$ $\alpha = \text{int} \times \alpha$

$\text{int} \times \alpha$ $\quad$ NI REŠITVE

Primer: $\lambda f\, x.\, f\,(f\, x)$



1) $\alpha = \beta \to \gamma$ ✓

$\alpha = \gamma \to \delta$ ✓

$\alpha := (\beta \to \gamma)$

$\beta \to \gamma = \gamma \to \delta$ ✓

↳ $\beta = \gamma$ ✓ $\quad \beta := \gamma$

↳ $\gamma = \delta$ $\quad \gamma := \delta$

Odgovor: $(\delta \to \delta) \to (\delta \to \delta)$

Rešitev:

$\alpha := (\beta \to \gamma)$

$\beta := \gamma$

$\gamma := \delta$

$\alpha := (\delta \to \delta)$

$\beta := \delta$

$\gamma := \delta$

IZREK O VARNOSTI

Program, ki ima tip, je varen.

?