

Izpeljava tipov

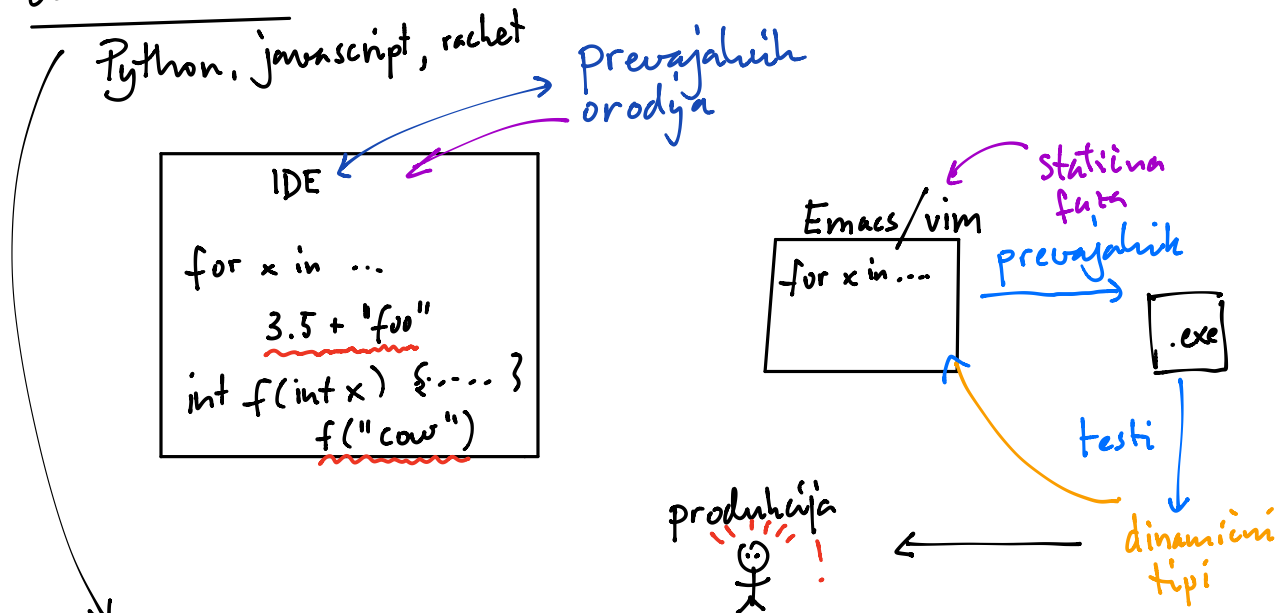
Tipi so lahko striktni:

- program je neveljaven, če je kaj narobe s tipi.
- kako striktno tipi opisujejo vrednosti & izraze

Statični & dinamični tipi

Kdaj se preverjajo tipi?

- statični → v statični fazi (preden potekemo)
Ocaml/SML, Haskell, Java, C/C++, ...
- dinamični → v dinamični fazi (ko program teče)



- izobraževanje - uvod v programiranje
- itd. (spletne strani)

Kombinacija dinamični + statični

- dinamično nalaganje nove kode
- gradual typing

Preverjamo ali izpeljujemo tipe

programer zapiše tipe,
prevajalnik preveri,
ali je vse ok.

(C/C++, Java, C#)

↓
podpirajo izpeljavo v manjši meri

Kako programer zapiše tip?

```
public int f(int x, List<int> y) {  
    int z = x + 3;  
    ...  
}
```

programer ne napiše tipa,
prevajalnik izračuna/izpelje
tip in s tem potrdi, da
je vse ok.

(Ocaml, Haskell, SML)

↑
dopuščajo tudi preverjanje
(programer sme napisati tip)

ListHashDictionary (Bod, truden) t=?

Ko programer zapiše tipe:

- poda dodatno informacijo prevajalniku
- in svojim sodelavcem
- in samemu sebi

Monomorfní in polimorfní tipi

Vsaki izraz ima natanko en tip

izraz ima lahko več tipov

ad hoc

Java, C++

overloading

parametrični polimorfizem

OCaml / Haskell

ENA IMPLEMENTACIJA

ima več tipov

$f(42)$
f ima dva tipa
IN DVE IMPLEMENTACIJI

```
public f(int x) { -I- }  
public f(bool b, int y) { -II- }  
f ima dva tipa
```

```
+ : int x int → int  
float x float → float
```

List<String> l = ...

List<Bool>

List<List<Integer>>>

```
public class List<T> {  
    ...  
}
```

Parametrični polimorfizem

fun x → x

λ x. x

int → int

bool → bool

int × β → int × β

α → α

GLAVNI TIP

parameter
α poljuben

Glavni tip : vse ostale tipe izraza dobimo kot posebne glavnega tipa (parametre v glavnem tipu nadomestimo z drugimi tipi)

fun $x \rightarrow x$

GLAVNI TIP

$\alpha \rightarrow \alpha$

$\text{int} \times \beta \rightarrow \text{int} \times \beta$

Ustanovimo $\alpha := \text{int} \times \beta$

$\text{bool} \rightarrow \text{bool}$

$\alpha := \text{bool}$

OCaml / SML

če ima izraz tip, ima glavni tip, ki ga lahko izpeljemo.

Kako izpeljemo glavni tip?

Imamo izraz e in izpeljemo njegov glavni tip.

① Izračunamo kandidata za tip e , ki vsebuje neznanke + enačbe, ki jim morajo zadoščati neznanke

② Rešimo enačbe in poiščemo vrednosti neznank (združevanje, unification)

$$f(X) = f(a)$$

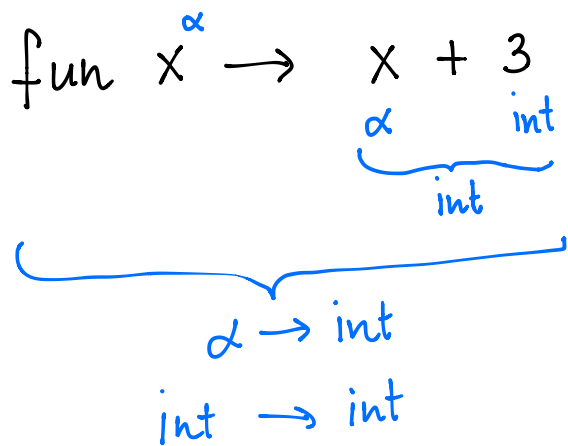
$$\Rightarrow X = a$$

$$x^2 + x - 3 = 0$$

$$\underline{x} = 3 - \underline{x^2}$$

$$\begin{aligned} \alpha &= \alpha \times \text{bool} & (\alpha = \text{vec } t, t \times \text{bool}) \\ &= (\alpha \times \text{bool}) \times \text{bool} \\ &= ((\alpha \times \text{bool}) \times \text{bool}) \times \text{bool} \\ &\dots \end{aligned}$$

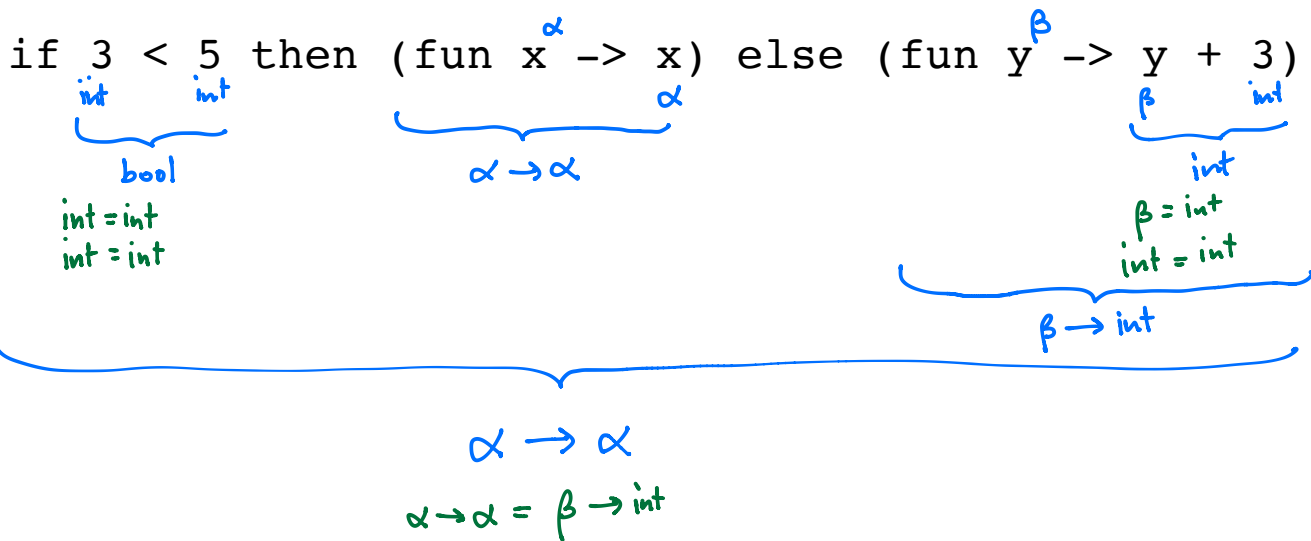
Primer



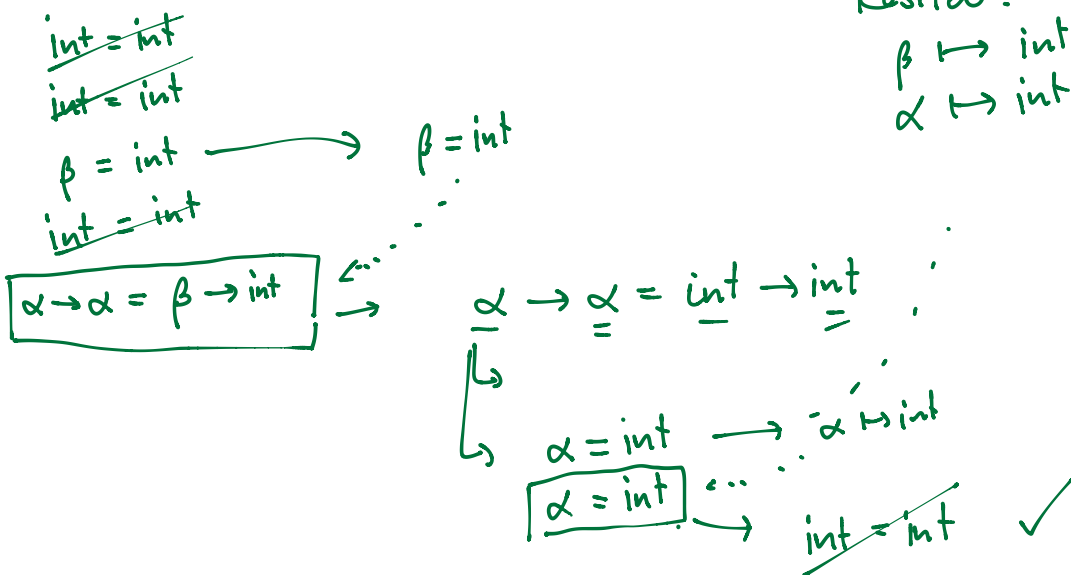
Resitev:
 $\alpha \mapsto \text{int}$

~~$\alpha = \text{int}$~~
 ~~$\text{int} = \text{int}$~~

Primer:



② Resnjemo:



Resitev:

$\beta \mapsto \text{int}$
 $\alpha \mapsto \text{int}$

ODgovor: $\text{int} \rightarrow \text{int}$

$$\text{fun } f^\alpha \rightarrow f \underbrace{4}_{\substack{\alpha \text{ int} \\ \beta}}$$

$$\alpha = \text{int} \rightarrow \beta$$

Resiter: $\alpha \mapsto (\text{int} \rightarrow \beta)$

$$\underbrace{\hspace{10em}}_{\alpha \rightarrow \beta}$$

$$(\text{int} \rightarrow \beta) \rightarrow \beta$$

$$\text{fun } f^\alpha \rightarrow (f \underbrace{3}_{\substack{\alpha \text{ int} \\ \beta}}) f \alpha$$

$$\alpha = \text{int} \rightarrow \beta$$

$$\beta = \alpha \rightarrow \gamma \rightsquigarrow$$

Resiter:

$$\alpha \mapsto \text{int} \rightarrow \beta$$

$$\boxed{\beta = (\text{int} \rightarrow \beta) \rightarrow \gamma}$$

NI RESITVE

$$\underbrace{\hspace{10em}}_{\alpha \rightarrow \gamma}$$

TA IZRAZ NIMA TIPA

Resiter:

$$\beta \mapsto \alpha \rightarrow \gamma$$

$$\boxed{\alpha = \text{int} \rightarrow (\alpha \rightarrow \gamma)}$$

NI RESITVE

$$\lambda f x^\beta . f x^\alpha$$

$$\alpha = \beta \rightarrow \gamma$$

$$\underbrace{\hspace{10em}}$$

$$\alpha \rightarrow \beta \rightarrow \gamma$$

$$(\hat{\alpha} \rightarrow \hat{\beta}) \rightarrow \hat{\beta} \rightarrow \hat{\gamma}$$

Ekvivalentno: $(\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \beta$

$(\xi \rightarrow \delta) \rightarrow \xi \rightarrow \delta$

$$\lambda f x^{\alpha \rightarrow \beta} . f (f x^\alpha)$$

$$\underbrace{\alpha \rightarrow \beta \quad \alpha \rightarrow \beta \quad \alpha}_{\beta}$$

$$\underbrace{\hspace{10em}}_{\gamma}$$

$$\underbrace{\hspace{10em}}_{(\alpha \rightarrow \beta) \rightarrow \alpha \rightarrow \gamma}$$

$$\alpha \rightarrow \beta = \beta \rightarrow \gamma$$

$$\rightarrow \alpha = \beta$$

$$\rightarrow \beta = \gamma$$

$$\alpha = \beta = \gamma$$

$$(\alpha \rightarrow \alpha) \rightarrow \alpha \rightarrow \alpha$$

```

let rec f =
  fun n ->
    if n = 0 then
      1
    else
      n * f (n - 1)

```

Annotations and Diagrams:

- α above f
- $\beta \rightarrow \text{int}$ above n
- $\beta = \text{int}$ above $=$ in $n = 0$
- $\text{int} \checkmark$ above 0
- int above 1
- int above n in $n * f$
- α above $*$
- int above n in $(n - 1)$
- int above 1 in $(n - 1)$
- Underline under $(n - 1)$ with int below it
- Brace under $n * f (n - 1)$ with γ and $\textcircled{2}$ below it
- Brace under $\text{if } n = 0 \text{ then } 1 \text{ else } n * f (n - 1)$ with $\text{int} \rightarrow \text{int}$ below it
- Large brace under the entire function definition with $f : \alpha$ below it
- Below $f : \alpha$:
 - $\text{int} \rightarrow \gamma$
 - $\text{int} \rightarrow \text{int}$

$\beta = \text{int} \checkmark$
 $\alpha = \text{int} \rightarrow \gamma$
 $\textcircled{1} = \textcircled{2}$ $\text{int} = \gamma$
 rekurzija $\alpha = \text{int} \rightarrow \text{int}$
 $\text{int} \rightarrow \gamma = \text{int} \rightarrow \text{int}$
 $\text{int} \rightarrow \text{int} = \text{int} \rightarrow \text{int}$
 Rezultat:
 $\beta \mapsto \text{int}$
 $\alpha \mapsto \text{int} \rightarrow \gamma$
 $\gamma \mapsto \text{int}$