

Lambda kalkul : definice pojmů , operací , reprezentace čísel

Lambda calculus - ekvivalentní s Turingovým strojem

- nejmenší univerzální programovací jazyk na světě

$\langle \text{expr} \rangle ::= \langle \text{name} \rangle \mid \langle \text{function} \rangle \mid \langle \text{application} \rangle$

$\langle \text{expr} \rangle ::= (\langle \text{expr} \rangle)$ - libovolný výraz můžeme uzavřít do závorek

$\langle \text{name} \rangle ::= \text{constant} \mid \text{variable}$

$\langle \text{function} \rangle ::= \lambda \langle \text{name} \rangle . \langle \text{expr} \rangle$ - jistina libovolného slova

$\langle \text{application} \rangle ::= \langle \text{expr} \rangle \langle \text{expr} \rangle$

příklady : $\lambda x. x$ - identity function

$(\lambda x. x) y$ - aplikace funkce na proměnnou y

\rightarrow všechna x budou nahrazena hodnotou y (bela redukce)

$(\lambda x. + x 1) 4 \rightarrow (+ 4 1) \rightarrow 5$

$(\lambda x. + x x) 3 \rightarrow (+ 3 3) \rightarrow 6$ (argumenty se nemusí vyskytovat vůbec)

- argumentem mohou být jiné funkce

$(\lambda f. f 4) (\lambda x. + x 1) \rightarrow (\lambda x. + x 1) 4 \rightarrow 5$

$(\lambda x. + x y)$
 $\swarrow \searrow$
 free variable
 bound variable

- při substituci nahrazení pouze volné x v E

\rightarrow v případě stejného pojmenování mohou mít problém

$\rightarrow \alpha$ -redukce = přejmenování volných proměnných ve funkci

$(\lambda x. E) F \rightarrow E'$

- E' vznikne nahrazením všech volných x v E hodnotou F

representable čísel: - pomocí funkce

$$0 \equiv \lambda s z. z \quad \text{— dva stejné argumenty}$$

$$1 \equiv \lambda s z. s(z) \approx \text{succ}(\text{zero})$$

$$2 \equiv \lambda s z. s(s(z)) \approx \text{succ}(\text{succ}(\text{zero}))$$

$$3 \equiv \lambda s z. s(s(s(z)))$$

→ successor function: $S \equiv \lambda x y x. y (w y x)$

$$\lambda y x. y ((\lambda s z. z) y x) \rightarrow \lambda y x. y ((\lambda z. z) x) \rightarrow \lambda y x. y(x) \equiv 1$$

The Y combinator: funkce pro rekursi

$$\boxed{\lambda f. (\lambda x. f(x, x)) (\lambda x. f(x, x))}$$

TRUE: $\lambda x y. x$ FALSE: $\lambda x y. y$ NOT: $\lambda a. a (\lambda b c. c) (\lambda d e. d)$

AND: $\lambda a b. a b (\lambda x y. y) \rightarrow \lambda a b. a b$ FALSE

OR: $\lambda a b. a (\lambda x y. x) b \rightarrow \lambda a b. b$ TRUE $\neq b$

IS-ZERO n: $\lambda n. n \text{ FALSE NOT FALSE}$

GREATER-OR-EQUAL n m: $\lambda n m. \text{IS-ZERO } (n P m)$

← predecessor function

sčítání: rekursivní successor funkce a kód na b

$$3 + 5 \equiv 3 \text{ succ } 5$$

množení: $\lambda a b c. a(b c)$