

Druhy rekurze

Svarny Petr

Katedra logiky FF UK

21. března 2022

Overview

Rekurze

DeRekurze

Rekurze - úloha

- ▶ Násobení vs sčítání

Není rekurze jako rekurze

- ▶ **Lineární** - jen disjunktní, ne současná, rekurzivní volání (faktoriál).
- ▶ **Koncová (tail)** - rekurze je poslední příkaz (např. Euclidův algoritmus největšího společného dělitele).
- ▶ **Kaskádová** - v těle jsou alespoň dvě rekurzivní volání (Fibonacci).
- ▶ **Vnořená** - argumenty funkce rekurzivně dané! (Ackermannova funkce)

Ackermannova funkce

$$A(0, n) = n + 1$$

$$A(m + 1, 0) = A(m, 1)$$

$$A(m + 1, n + 1) = A(m, A(m + 1, n))$$

Derekurze pomocí koncové rekurze

=MoertelBlog1=

=MoertelBlog2=

- ▶ Z plně rekurzivního volání uděláme koncové volání
- ▶ Převédeme rekurzi na jednu smyčku
- ▶ Upravíme pomocí continue nebo podobného

Factorial

```
def factorial(n):  
    if n < 2:  
        return 1  
    return n * factorial(n - 1)
```

Tail call

```
def factorial1a(n, acc=1):  
    if n < 2:  
        return 1 * acc  
    return factorial1a(n - 1, acc * n)
```


One call

```
def factorial1b(n, acc=1):  
    while True:  
        if n < 2:  
            return 1 * acc  
        return factorial1b(n - 1, acc * n)  
    break
```

Continue

```
def factorial1c(n, acc=1):  
    while True:  
        if n < 2:  
            return 1 * acc  
        (n, acc) = (n - 1, acc * n)  
        continue  
    break
```

Cleaning

```
def factorial1d(n, acc=1):  
    while n > 1:  
        (n, acc) = (n - 1, acc * n)  
    return acc
```