

# 2-AIN-285 Symbolické programovanie a LISP

Letný semester 2019/20

1. prednáška

Ján Komara

# Obsah 1. prednášky

Cieľ a obsah predmetu

Programovací jazyk Lisp

Euklidov algoritmus (názorný príklad)

Záver

# Cieľ a obsah predmetu

## Cieľ predmetu

- ▶ Po absolvovaní predmetu budú študenti vedieť programovať v programovacom jazyku Lisp dialekt Scheme.
- ▶ Naučia sa používať všeobecné programovacie techniky ako abstrakcia na procedúrach a dátach, streamové spracovávanie dát, a ďalšie.
- ▶ Získajú skúsenosť ako navrhnúť a vytvoriť interpreter programovacieho jazyka Lisp.

## Stručná osnova predmetu

- ▶ Abstrakcia procedúr a dát.
- ▶ Modularita, objekty a lokálny stav.
- ▶ Univerzálna funkcia pre čiastočne rekurzívne funkcie.
- ▶ Interpreter programovacieho jazyka LISP.

# Programovací jazyk Lisp

- ▶ Lisp je multiparadigmatický programovací jazyk (funkcionálne, deklaratívne, imperatívne programovanie).
- ▶ Autorom je John McCarthy z MIT (1958, 1960).
- ▶ Lisp je skratka *LIS*t Processor.
- ▶ Návrh jazyka bol silne ovplyvnený formalizmom čiastočne rekurzívnych funkcií a  $\lambda$ -kalkulom.
  - ▶ Churchova téza (1936): trieda intuitívne vypočítateľných funkcií je totožná s obecnými rekurzívnymi funkciami.
- ▶ Základný dátový typ: symbolické výrazy (numerické dáta, zoznamy, stromy, programy).
- ▶ Využitie: umelá inteligencia.
- ▶ Hlavné dialekty jazyka Lisp:
  - ▶ Scheme (1975) – používaný pri výučbe tohto predmetu.
  - ▶ Common Lisp (1984).

# Programovací jazyk Lisp

Lisp – prvý programovací jazyk deklaratívnej paradigmy.

- ▶ Deklaratívne programy sú definície matematických objektov (funkcie, relácie).
- ▶ Zhoda medzi definičnou a výpočtovou sémantikou umožňuje analyzovať programy elementárnymi prostriedkami.
- ▶ Všetky časti tvorby programu je možné realizovať v tom istom formalizme:
  - ▶ špecifikácia,
  - ▶ implementácia,
  - ▶ verifikácia,
  - ▶ výpočet.
- ▶ Jednoduchá sémantika sa kombinuje s expresívnymi programátorskými konštrukciami.

# Euklidov algoritmus (názorný príklad)

## Špecifikácia

- Špecifikačný predikát

$$x \mid y \leftrightarrow \exists z y = x \cdot z.$$

- Špecifikácia programu

$$x \neq 0 \vee y \neq 0 \rightarrow \text{gcd}(x, y) \mid x \wedge \text{gcd}(x, y) \mid y \wedge \\ \forall z (z \mid x \wedge z \mid y \rightarrow z \leq \text{gcd}(x, y)).$$

Tu  $\text{gcd}(x, y)$  označuje najväčšieho spoločného deliteľa prirodzených čísel  $x$  a  $y$ .

- Idea algoritmu

$$x > y \wedge z \mid y \rightarrow z \mid x \leftrightarrow z \mid x \div y.$$

# Euklidov algoritmus (názorný príklad)

## Implementácia

```
gcd(x, y) = if x ≠ 0 ∧ y ≠ 0 then
             case
               x > y ⇒ gcd(x ÷ y, y)
               x = y ⇒ x
               x < y ⇒ gcd(x, y ÷ x)
             end
           else
             max(x, y)
```

## Výpočet

Výpočet s argumentami klesajúcimi v miere  $\max(x, y)$ :

$$\begin{array}{ccccccccc} \text{gcd}(9, 12) & = & \text{gcd}(9, 3) & = & \text{gcd}(6, 3) & = & \text{gcd}(3, 3) & = & 3 \\ 12 & & 9 & & 6 & & 3 & & \\ & & > & & > & & > & \end{array}$$

# Euklidov algoritmus (názorný príklad)

## Implementácia

```
gcd(x, y) = if x ≠ 0 ∧ y ≠ 0 then
  case
    x > y ⇒ gcd(x ÷ y, y)
    x = y ⇒ x
    x < y ⇒ gcd(x, y ÷ x)
  end
else
  max(x, y)
```

## Výpočet

Podmienky regularity zaručujú, že výpočet vždy skončí:

$$x \neq 0 \wedge y \neq 0 \wedge x > y \rightarrow \max(x \div y, y) < \max(x, y)$$

$$x \neq 0 \wedge y \neq 0 \wedge x < y \rightarrow \max(x, y \div x) < \max(x, y).$$



# Euklidov algoritmus (názorný príklad)

## Implementácia

```
gcd(x, y) = if x ≠ 0 ∧ y ≠ 0 then
             case
               x > y ⇒ gcd(x ÷ y, y)
               x = y ⇒ x
               x < y ⇒ gcd(x, y ÷ x)
             end
           else
             max(x, y)
```

## Výpočet

Podmienka  $x \neq 0 \wedge y \neq 0$  sa zbytočne opakovane vyhodnocuje

$$\gcd(9, 12) = \gcd(9, 3) = \gcd(6, 3) = \gcd(3, 3) = 3$$

# Euklidov algoritmus (názorný príklad)

Alternatívna implementácia so vstupnou podmienkou

```
x ≠ 0 ∧ y ≠ 0 → gcd(x, y) = case  
    x > y ⇒ gcd(x ÷ y, y)  
    x = y ⇒ x  
    x < y ⇒ gcd(x, y ÷ x)  
end
```

Výpočet

Ten nemusí skončiť:

$$\gcd(1, 0) = \gcd(1 \div 0, 0) = \gcd(1, 0) = \dots$$

Program počíta čiastočnú funkciu!

# Euklidov algoritmus (názorný príklad)

Alternatívna implementácia so vstupnou podmienkou

```
x ≠ 0 ∧ y ≠ 0 → gcd(x, y) = case
    x > y ⇒ gcd(x ÷ y, y)
    x = y ⇒ x
    x < y ⇒ gcd(x, y ÷ x)
end
```

## Výpočet

Rozšírené podmienky regularity pre mieru  $\max(x, y)$ :

$$x \neq 0 \wedge y \neq 0 \wedge x > y \rightarrow \max(x \div y, y) < \max(x, y) \wedge x \div y \neq 0 \wedge y \neq 0$$

$$x \neq 0 \wedge y \neq 0 \wedge x < y \rightarrow \max(x, y \div x) < \max(x, y) \wedge x \neq 0 \wedge y \div x \neq 0.$$

Výpočet skončí pre vstupy spĺňajúce vstupnú podmienku.

# Záver

## Organizácia kurzu

- ▶ Prednášky/cvičenia: pondelok 11:30-13:50, m. I-H3.
- ▶ Konzultácie: po dohode.
- ▶ Web: <http://ii.fmph.uniba.sk/cl/courses/>

## Hodnotenie

- ▶ Cvičenia 40 % – max. 40 bodov.
- ▶ Testy, prémiové úlohy 30 % – max. 30 bodov.
- ▶ Projekty, projektové úlohy 30 % – max. 30 bodov.
- ▶ Znamky: E 50 %, D 60 %, C 70 %, B 80 %, A 90 %.

# Záver

## 1. cvičenie

- ▶ Začína hneď po prednáške v tej istej miestnosti.
- ▶ Téma:
  - ▶ Vyhodnocovanie výrazov.
  - ▶ Explicitné definície.
  - ▶ Rekurzívne definície.
- ▶ Úlohy odovzdať najneskôr do 18:00 v nedeľu tento týždeň.

## 2. prednáška/cvičenie

- ▶ Definície predikátov.
- ▶ Iteratívna rekurzia.
- ▶ Funkcionálne programovanie.