

1-AIN-470 Špecifikácia a verifikácia programov

Letný semester 2024/25

1. prednáška

Ján Komara

Obsah 1. prednášky

Cieľ a obsah predmetu

Deklaratívne programovanie

Programovací jazyk

Základné informácie o predmete

Záver

Cieľ a obsah predmetu

Cieľ predmetu

- ▶ Predmet rozvíja schopnosti študentov uvažovať o správnosti programov, formálne špecifikovať požadované vlastnosti a dokazovať ich splnenie využitím rôznych metód, najmä štrukturálnej indukcie.
- ▶ Absolventi získajú znalosť konkrétnej formalizácie rekurzívnych programov, ich vlastností a dôkazov v jednoduchej logickej teórii Peanovej aritmetiky.
- ▶ Získajú tiež praktickú skúsenosť so špecifikáciou a verifikáciou väčšieho počtu programov.

Cieľ a obsah predmetu

Obsah predmetu

- ▶ *Algoritmy a dátové štruktúry.* Reťazce. Zoznamy. Operácie na zoznamoch. Triedenie zoznamov. Aplikácie zoznamov. Binárne stromy. Binárne vyhľadávacie stromy. Aplikácie stromov. Obecné stromy. Symbolické výrazy. Interpreter programovacieho jazyka. Univerzálna funkcia.
- ▶ *Deklaratívne programovanie.* Primitívna rekurzia. Rekurzia s mierou. Iteratívna rekurzia. Rekurzia na notácii. Párovacia funkcia a aritmetizácia. Štrukturálna rekurzia.
- ▶ *Špecifikačno-verifikačný systém.* Peanova aritmetika. Matematická indukcia. Rozšírenia aritmetiky. Odvodené indukčné princípy: úplná matematická indukcia, indukcia s mierou, štrukturálna indukcia.

Deklaratívne programovanie

Paradigma deklaratívneho programovania

- ▶ 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.

Deklaratívne programovanie

Výučba deklaratívneho programovania na fakulte

- ▶ Softvér používaný pri výučbe tohto predmetu:
 - ▶ programovací jazyk a špecifikačno-verifikačný systém CL (Clausal Language).
- ▶ Súvisiace predmety:
 - ▶ 2-AIN-266 Deklaratívne programovanie,
 - ▶ 2-AIN-285 Symbolické programovanie a LISP.
- ▶ Iné deklaratívne programovacie jazyky:
 - ▶ funkcionálne programovacie jazyky – Haskell;
 - ▶ logické programovacie jazyky – Prolog.

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

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 .

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Deklaratívny program

Idea algoritmu je založená na tejto vlastnosti

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

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).
```


Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Deklaratívny program

```
gcd(x, y) = if  $x \neq 0 \wedge y \neq 0$  then  
    case  
         $x > y \Rightarrow \text{gcd}(x \div y, y)$   
         $x = y \Rightarrow x$   
         $x < y \Rightarrow \text{gcd}(x, y \div x)$   
    end  
else  
     $\max(x, y)$ .  
end
```

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Výpočet

$$\text{gcd}(12, 9) = \text{gcd}(3, 9) = \text{gcd}(3, 6) = \text{gcd}(3, 3) = 3.$$

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Deklaratívny program

```
gcd(x, y) = if  $x \neq 0 \wedge y \neq 0$  then  
    case  
         $x > y \Rightarrow \text{gcd}(x \div y, y)$   
         $x = y \Rightarrow x$   
         $x < y \Rightarrow \text{gcd}(x, y \div x)$   
    end  
else  
     $\max(x, y)$ .  
end
```

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Výpočet

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

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

Terminácia programu

Podmienky regularity zaručujú, že výpočet vždy skončí:

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

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

Je to regulárny rekurzívny program s mierou $\max(x, y)$.

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Deklaratívny program

```
gcd(x, y) = if  $x \neq 0 \wedge y \neq 0$  then
  case
     $x > y \Rightarrow \text{gcd}(x \div y, y)$ 
     $x = y \Rightarrow x$ 
     $x < y \Rightarrow \text{gcd}(x, y \div x)$ 
  end
else
   $\max(x, y)$ .
```

Definovateľnosť

- ▶ Podmienky regularity zaručujú, že tento program čítaný ako rovnosť je korektná definícia binárnej funkcie $\text{gcd}(x, y)$.
- ▶ Je to regulárna rekurzívna definícia s mierou $\max(x, y)$.

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Zhoda medzi definičnou a výpočtovou sémantikou

Pre každý program v tvare rovnosti, ktorý je zároveň regulárnou definíciou s mierou, platí:

to, čo je počítané týmto programom, je presne to, čo je touto rovnosťou definované.

Verifikácia programu

Na dôkaz špecifikačného tvrdenia

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

stačí tak elementárna matematika! Netreba formalizovať koncept výpočtu.

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Deklaratívny program

```
gcd(x, y) = if  $x \neq 0 \wedge y \neq 0$  then
    case
         $x > y \Rightarrow \text{gcd}(x \div y, y)$ 
         $x = y \Rightarrow x$ 
         $x < y \Rightarrow \text{gcd}(x, y \div x)$ 
    end
else
    max(x, y)
```

Výpočet

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

$$\text{gcd}(12, 9) = \text{gcd}(3, 9) = \text{gcd}(3, 6) = \text{gcd}(3, 3) = 3.$$

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Alternatívna implementácia

```
gcd(x, y) = case
    x > y ⇒ gcd(x ÷ y, y)
    x = y ⇒ x
    x < y ⇒ gcd(x, y ÷ x)
end.
```

Výpočet

Podmienka $x \neq 0 \wedge y \neq 0$ sa už zbytočne opakovane nevyhodnocuje

$$\text{gcd}(12, 9) = \text{gcd}(3, 9) = \text{gcd}(3, 6) = \text{gcd}(3, 3) = 3.$$

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Alternatívna implementácia

$\text{gcd}(x, y) = \text{case}$

$x > y \Rightarrow \text{gcd}(x \div y, y)$

$x = y \Rightarrow x$

$x < y \Rightarrow \text{gcd}(x, y \div x)$

end.

Výpočet

Ten nemusí skončiť:

$$\text{gcd}(1, 0) = \text{gcd}(1 \div 0, 0) = \text{gcd}(1, 0) = \dots$$

Program počíta čiastočnú funkciu!

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Terminácia programu

Rozšírené podmienky regularity

$$x \neq 0 \wedge y \neq 0 \wedge x > y \rightarrow$$

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

$$x \neq 0 \wedge y \neq 0 \wedge x < y \rightarrow$$

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

zaručia, že výpočet programu skončí pre všetky vstupy spĺňajúce túto vstupnú podmienku

$$x \neq 0 \wedge y \neq 0.$$

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Zoslabenie podmienky definovateľnosti

Ako súvisí nasledujúca rekurzívna rovnosť s funkciou $\text{gcd}(x, y)$?

$$\begin{aligned} \text{gcd}(x, y) = \text{case} \\ & x > y \Rightarrow \text{gcd}(x \div y, y) \\ & x = y \Rightarrow x \\ & x < y \Rightarrow \text{gcd}(x, y \div x) \\ \text{end.} \end{aligned}$$

Pridaním vstupnej podmienky dostaneme vlastnosť tejto funkcie:

$$\begin{aligned} x \neq 0 \wedge y \neq 0 \rightarrow \text{gcd}(x, y) = \text{case} \\ & x > y \Rightarrow \text{gcd}(x \div y, y) \\ & x = y \Rightarrow x \\ & x < y \Rightarrow \text{gcd}(x, y \div x) \\ \text{end.} \end{aligned}$$

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Totálna korektnosť

Vyhodnocovanie programu skončí pre všetky vstupy spĺňajúce vstupnú podmienku

$$x \neq 0 \wedge y \neq 0$$

s vypočítanou korektnou hodnotou $\text{gcd}(x, y)$.

Deklaratívne programovanie

Zhrnutie

- ▶ *Univerzum* je množina prirodzených čísel

$$\mathbb{N} = \{0, 1, 2, 3, 4, 5, \dots\}.$$

- ▶ *Dátové štruktúry* kódujeme do \mathbb{N} v štýle jazyka Lisp s pomocou vhodnej párovacej funkcie.
- ▶ *Programy* počítajú (totálne) funkcie nad oborom \mathbb{N} pre vstupy spĺňajúce určité vstupné podmienky. Zároveň vyjadrujú ich vlastnosti pre argumenty spĺňajúce tie vstupné podmienky.
- ▶ *Formálny systém* je druhorádová formalizácia aritmetiky.

Deklaratívne programovanie

Literatúra

- ▶ Ján Komara. Specification and Verification of Programs. Downloadable lecture notes available through the web page of the course.
- ▶ Ján Kľuka. Prednášky z Úvodu do deklaratívneho programovania LS 2014/2015.
- ▶ Ján Komara. Recursive Functions. Lecture notes for the course 2-INF-264 Theory of Declarative Programming Winter 2015/16.

Programovací jazyk

Programy deklaratívnej paradigmy

Explicitné a regulárne rekurzívne definície:

$$f(x_1, \dots, x_n) = \tau[f; x_1, \dots, x_n].$$

Základné konštrukcie pri tvorbe programov

- ▶ Premenné a konštanty.
- ▶ Funkčné aplikácie.
- ▶ Podmienkové výrazy.
 - ▶ Jednoduché podmienkové výrazy:

$$D(\tau_1, \tau_2, \tau_3) \equiv \text{if } \tau_1 \neq 0 \text{ then } \tau_2 \text{ else } \tau_3.$$

- ▶ Obecné podmienkové výrazy:

$$\text{case } \varphi_1 \Rightarrow \rho_1 \dots \varphi_m \Rightarrow \rho_m \text{ end.}$$

Programovací jazyk

Charakteristická funkcia predikátu

Charakteristická funkcia n -árneho predikátu P je n -árna funkcia P_* definovaná predpisom

$$P_*(x_1, \dots, x_n) = \begin{cases} 1 & \text{ak platí } P(x_1, \dots, x_n), \\ 0 & \text{ak neplatí } P(x_1, \dots, x_n). \end{cases}$$

Notačná konvencia $x P_* y$ pre binárne predikáty s infixovou notáciou. Napr. $x =_* y$, $x \leq_* y$.

Charakteristický term formuly

Výraz φ_* je charakteristický term formuly φ , ak

$$(\varphi \rightarrow \varphi_* = 1) \wedge (\neg\varphi \rightarrow \varphi_* = 0).$$

Programovací jazyk

Dichotomická diskriminácia

Explicitná definícia funkcie $\max(x, y)$:

$$\begin{aligned} \max(x, y) = & \text{ case} \\ & x \geq y \Rightarrow x \\ & x < y \Rightarrow y \\ & \text{end.} \end{aligned}$$

Program:

$$\begin{aligned} \max(x, y) = & \text{ case} \\ & (x \geq_* y) = 1 \Rightarrow x \\ & (x <_* y) = 1 \Rightarrow y \\ & \text{end.} \end{aligned}$$

Preklad do definície s jednoduchým podmienkovým výrazom:

$$\max(x, y) = \text{if } (x \geq_* y) \neq 0 \text{ then } x \text{ else } y.$$

Programovací jazyk

Dichotomická diskriminácia

Explicitná definícia funkcie $\max(x, y)$:

$$\begin{aligned} \max(x, y) = & \text{ case} \\ & x \geq y \Rightarrow x \\ & x \leq y \Rightarrow y \\ & \text{end.} \end{aligned}$$

Program:

$$\begin{aligned} \max(x, y) = & \text{ case} \\ & (x \geq_* y) = 1 \Rightarrow x \\ & (x \leq_* y) = 1 \Rightarrow y \\ & \text{end.} \end{aligned}$$

Preklad do definície s jednoduchým podmienkovým výrazom:

$$\max(x, y) = \text{if } (x \geq_* y) \neq 0 \text{ then } x \text{ else } y.$$

Programovací jazyk

Diskriminácia na konštantách

Rekurzívna definícia Fibonacciho postupnosti:

$$f_n = \text{case}$$
$$n = 0 \Rightarrow 0$$
$$n = 1 \Rightarrow 1$$
$$n \neq 0 \wedge n \neq 1 \Rightarrow f_{n-1} + f_{n-2}$$
$$\text{end.}$$

Program:

$$f_n = \text{case}$$
$$(n =_* 0) = 1 \Rightarrow 0$$
$$(n =_* 1) = 1 \Rightarrow 1$$
$$(n \neq_* 0 \wedge_* n \neq_* 1) = 1 \Rightarrow f_{n-1} + f_{n-2}$$
$$\text{end.}$$

Preklad do definície s jednoduchými podmienkovými výrazmi:

$$f_n = \text{if } (n =_* 0) \neq 0 \text{ then } 0 \text{ else if } (n =_* 1) \neq 0 \text{ then } 1 \text{ else } f_{n-1} + f_{n-2}.$$

Programovací jazyk

Diskriminácia na konštantách

Rekurzívna definícia Fibonacciho postupnosti:

$$f_n = \text{case}$$
$$n = 0 \Rightarrow 0$$
$$n = 1 \Rightarrow 1$$
$$\text{otherwise} \Rightarrow f_{n-1} + f_{n-2}$$
$$\text{end.}$$

Program:

$$f_n = \text{case}$$
$$(n =_* 0) = 1 \Rightarrow 0$$
$$(n =_* 1) = 1 \Rightarrow 1$$
$$(n \geq_* 2) = 1 \Rightarrow f_{n-1} + f_{n-2}$$
$$\text{end.}$$

Preklad do definície s jednoduchými podmienkovými výrazmi:

$$f_n = \text{if } (n =_* 0) \neq 0 \text{ then } 0 \text{ else if } (n =_* 1) \neq 0 \text{ then } 1 \text{ else } f_{n-1} + f_{n-2}.$$

Programovací jazyk

Podmienkové výrazy

Syntax:

case	case
$\varphi_1[\vec{x}] \Rightarrow \rho_1[\vec{x}]$	$\chi_1[\vec{x}] = 1 \Rightarrow \rho_1[\vec{x}]$
\vdots	\vdots
$\varphi_m[\vec{x}] \Rightarrow \rho_m[\vec{x}]$	$\chi_m[\vec{x}] = 1 \Rightarrow \rho_m[\vec{x}]$
end	end.

Podmienka úplnosti a jednoznačnosti (výlučnosti):

$$\bigvee_{i=1}^m \varphi_i[\vec{x}] \wedge \bigwedge_{i,j=1} (\varphi_i[\vec{x}] \wedge \varphi_j[\vec{x}] \rightarrow \rho_i[\vec{x}] = \rho_j[\vec{x}])$$
$$\bigvee_{i=1}^m \varphi_i[\vec{x}] \wedge \bigwedge_{\substack{i,j=1 \\ i \neq j}}^m \neg(\varphi_i[\vec{x}] \wedge \varphi_j[\vec{x}]).$$

Programovací jazyk

Podmienkové výrazy

Syntax:

case	case
$\varphi_1[\vec{x}] \Rightarrow \rho_1[\vec{x}]$	$\chi_1[\vec{x}] = 1 \Rightarrow \rho_1[\vec{x}]$
\vdots	\vdots
$\varphi_m[\vec{x}] \Rightarrow \rho_m[\vec{x}]$	$\chi_m[\vec{x}] = 1 \Rightarrow \rho_m[\vec{x}]$
end	end.

Tu $\chi_i[\vec{x}]$ je charakteristický term pre $\varphi_i[\vec{x}]$:

$$\chi_i[\vec{x}] = 1 \vee \chi_i[\vec{x}] = 0 \quad \varphi_i[\vec{x}] \leftrightarrow \chi_i[\vec{x}] = 1.$$

Sémantika:

$$D\left(\chi_1[\vec{x}], \rho_1[\vec{x}], \dots, D(\chi_m[\vec{x}], \rho_m[\vec{x}], 0) \dots\right).$$

Základné informácie o predmete

Všeobecné informácie o kurze

<https://dai.fmph.uniba.sk/cl/view/courses/1-AIN-470-svp/?lang=sk>

- ▶ Informačný list predmetu:
 - ▶ sylabus predmetu,
 - ▶ literatúra.
- ▶ Výučba v predošlých rokoch.

Výučba tento semester

<https://dai.fmph.uniba.sk/cl/view/courses/1-AIN-470-svp/2425ls/?lang=sk>

- ▶ Novinky.
- ▶ Organizácia predmetu.
- ▶ Zadania cvičení.
- ▶ Termíny semestrálnych testov.
- ▶ Podmienky na absolvovanie predmetu.

Základné informácie o predmete

Prednášky

- ▶ Učiteľ: Ján Komara, m. I-16, jan.komara@fmph.uniba.sk.
- ▶ Prednášky: štvrtok 13.10 2h, M-XI.
- ▶ Cvičenia: štvrtok 14 50 2h, M-XI.
- ▶ Konzultácie:
 - ▶ pondelok pondelok o 14.30, m. I-16;
 - ▶ po dohode, online.

Základné informácie o predmete

Hodnotenie počas semestra

- ▶ Počas semestra môžete dohromady získať max. 60 bodov.
 - ▶ 1. a 2. test: 2 x 30 bodov.
 - ▶ Domáce neprémiové úlohy z cvičení.
- ▶ Požiadavka priebežného semestrálneho hodnotenia: aspoň 30 bodov.

Skúškové obdobie

- ▶ Zo skúšky môžete dohromady získať max. 40 bodov.
 - ▶ 3. test: 40 bodov.
 - ▶ Domáce prémiové úlohy z cvičení.

Celkové hodnotenie

- ▶ Dohromady možno získať 100 bodov.
- ▶ Znamky: E 50% (50 bodov), D 60%, C 70%, B 80%, A 90%.

Základné informácie o predmete

Distančná výučba

- ▶ Základné informácie
 - ▶ link: <https://uniba.sk/elearning>
- ▶ Moodle
 - ▶ domáce úlohy
 - ▶ link: <https://moodle.uniba.sk>
 - ▶ kurz: 1-AIN-470 Špecifikácia a verifikácia programov LS 2024/25
 - ▶ heslo: 1-AIN-470
- ▶ MS Teams
 - ▶ hodnotenie, konzultácie
 - ▶ link: <https://teams.microsoft.com>
 - ▶ kurz: 1-AIN-470 Špecifikácia a verifikácia programov LS 2024/25
 - ▶ kód: hmry3sm

Záver

Čo ďalej nasleduje?

- ▶ 1. cvičenie: začína hneď po prednáške.
- ▶ 2. prednáška: párovacia funkcia a aritmetizácia karteziánskeho súčinu.
- ▶ 3. prednáška: zoznamy, triedenie zoznamov, kombinatorické funkcie na zoznamoch.
- ▶ 4. prednáška: binárne stromy, binárne vyhľadávacie stromy.
- ▶ 5. prednáška: symbolické výrazy a numerické termy.
- ▶ 6. a 7. prednáška: logika.
- ▶ Zvyšné prednášky: verifikácia programov.
- ▶ Na záver ešte malý projekt: implementovať interpreter jednoduchého deklaratívneho programovacieho jazyka.

Záver

Otázky

- ▶ Ak máte otázku, tak zdvihnite ruku.

Koniec prednášky