1-AIN-470 Špecifikácia a verifikácia programov

Letný semester 2022/23

1. prednáška

Ján Komara

Obsah 1. prednášky

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 induktívne 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 progamovacie 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

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

Tu $\gcd(x, y)$ 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

$$x > y \wedge z \mid y \rightarrow z \mid x \leftrightarrow z \mid x \dot{-} y.$$

Implementácia

$$\begin{aligned}
\gcd(x, y) = \text{if } & x \neq 0 \wedge y \neq 0 \text{ then} \\
& \text{case} \\
& \quad x > y \Rightarrow \gcd(x \dot{-} y, y) \\
& \quad x = y \Rightarrow x \\
& \quad x < y \Rightarrow \gcd(x, y \dot{-} x) \\
& \text{end} \\
& \text{else} \\
& \quad \max(x, y).
\end{aligned}$$

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Deklaratívny program

$$
\begin{aligned}
\gcd(x, y) = \text{if } & x \neq 0 \wedge y \neq 0 \text{ then} \\
& \text{case} \\
& \quad x > y \Rightarrow \gcd(x \dot{-} y, y) \\
& \quad x = y \Rightarrow x \\
& \quad x < y \Rightarrow \gcd(x, y \dot{-} x) \\
& \text{end} \\
& \text{else} \\
& \quad \max(x, y).
\end{aligned}
$$

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Výpočet

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

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Deklaratívny program

$$
\begin{aligned}
\gcd(x, y) = \text{if } & x \neq 0 \wedge y \neq 0 \text{ then} \\
& \text{case} \\
& \quad x > y \Rightarrow \gcd(x \dot{-} y, y) \\
& \quad x = y \Rightarrow x \\
& \quad x < y \Rightarrow \gcd(x, y \dot{-} x) \\
& \text{end} \\
& \text{else} \\
& \quad \max(x, y).
\end{aligned}
$$

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Výpočet

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

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

Terminácia programu

Podmienky regularity zaručujú, že výpočet vždy skončí:

$$x \neq 0 \wedge y \neq 0 \wedge x > y \rightarrow \max(x \dot{-} y, y) < \max(x, y)$$
$$x \neq 0 \wedge y \neq 0 \wedge x < y \rightarrow \max(x, y \dot{-} x) < \max(x, y).$$

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

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Deklaratívny program

$$
\begin{aligned}
\gcd(x, y) = \text{if } & x \neq 0 \wedge y \neq 0 \text{ then} \\
& \text{case} \\
& \quad x > y \Rightarrow \gcd(x \dot{-} y, y) \\
& \quad x = y \Rightarrow x \\
& \quad x < y \Rightarrow \gcd(x, y \dot{-} x) \\
& \text{end} \\
& \text{else} \\
& \quad \max(x, y).
\end{aligned}
$$

Definovateľnosť

- Podmienky regularity zaručujú, že tento program čítaný ako rovnosť je korektná definícia binárnej funkcie $\gcd(x, y)$.
- 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

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

stačí tak elementárna matematika! Netreba formalizovať koncept výpočtu.

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Deklaratívny program

$$
\begin{aligned}
\gcd(x, y) = \text{if } & x \neq 0 \wedge y \neq 0 \text{ then} \\
& \text{case} \\
& \quad x > y \Rightarrow \gcd(x \dot{-} y, y) \\
& \quad x = y \Rightarrow x \\
& \quad x < y \Rightarrow \gcd(x, y \dot{-} x) \\
& \text{end} \\
& \text{else} \\
& \quad \max(x, y)
\end{aligned}
$$

Výpočet

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

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

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Alternatívna implementácia

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

Výpočet

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

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

Deklaratívne programovanie

Euklidov algoritmus pre výpočet najväčšieho spoločného deliteľa

Alternatívna implementácia

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

Výpočet

Ten nemusí skončiť:

$$\gcd(1, 0) = \gcd(1 \dot{-} 0, 0) = \gcd(1, 0) = \cdots .$$

Program počíta čiastočnú funkciu!

Terminácia programu

Rozšírené podmienky regularity

$$x \neq 0 \wedge y \neq 0 \wedge x > y \rightarrow$$
$$\max(x \dot{-} y, y) < \max(x, y) \wedge x \dot{-} y \neq 0 \wedge y \neq 0$$
$$x \neq 0 \wedge y \neq 0 \wedge x < y \rightarrow$$
$$\max(x, y \dot{-} x) < \max(x, y) \wedge x \neq 0 \wedge y \dot{-} x \neq 0$$

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 $\gcd(x, y)$?

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

Pridaním vstupnej podmienky dostaneme vlastnosť tejto funkcie:

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

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 $\gcd(x, y)$.

Deklaratívne programovanie

Zhrnutie

- *Univerzum* je množina prirodzených čísel

$$\mathrm{N} = \{0, 1, 2, 3, 4, 5, \ldots\}.$$

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

Programovací jazyk

Programy deklaratívnej paradigmy

Explicitné a regulárne rekurzívne definície:

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

Základné konštrukcie pri tvorbe programov

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

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

  - Obecné podmienkové výrazy:

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

Programovací jazyk

Charakteristická funkcia predikátu

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

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

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

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

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:

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

Program:

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

Preklad do definície s jednoduchými podmienkovými výrazmi:

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

Programovací jazyk

Diskriminácia na konštantách

Rekurzívna definícia Fibonacciho postupnosti:

$$
\begin{aligned}
f_n = \text{case} \\
\quad & n = 0 \Rightarrow 0 \\
\quad & n = 1 \Rightarrow 1 \\
\quad & \text{otherwise} \Rightarrow f_{n \dot{-} 1} + f_{n \dot{-} 2} \\
\text{end.}
\end{aligned}
$$

Program:

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

Preklad do definície s jednoduchými podmienkovými výrazmi:

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

Programovací jazyk

Podmienkové výrazy

Syntax:

case
$\varphi_1[\vec{x}] \Rightarrow \rho_1[\vec{x}]$
$\vdots$
$\varphi_m[\vec{x}] \Rightarrow \rho_m[\vec{x}]$
end

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

Podmienka úplnosti a jednoznačnosti (výlučnosti):

$$\bigvee_{i=1}^{m} \varphi_i[\vec{x}] \wedge \bigwedge_{i,j=1} \big(\varphi_i[\vec{x}] \wedge \varphi_j[\vec{x}] \to \rho_i[\vec{x}] = \rho_j[\vec{x}]\big)$$

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

Programovací jazyk

Podmienkové výrazy

Syntax:

case
$\varphi_1[\vec{x}] \Rightarrow \rho_1[\vec{x}]$
$\vdots$
$\varphi_m[\vec{x}] \Rightarrow \rho_m[\vec{x}]$
end

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

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

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

Sémantika:

$$\mathsf{D}\Big(\chi_1[\vec{x}], \rho_1[\vec{x}], \ldots, \mathsf{D}\big(\chi_m[\vec{x}], \rho_m[\vec{x}], 0\big) \ldots\Big).$$

Záver

Organizácia kurzu

- Prednášky: pondelok 8.10 2h, M-XI.
- Cvičenia: pondelok 11.30 2h, F2-T3.
- Konzultácie: po dohode.
- Web: http://ii.fmph.uniba.sk/cl/courses/

Hodnotenie

- Cvičenia, 1. test 60% – max. 60 bodov.
- Prémiové úlohy, 2. test 40% – max. 40 bodov.
- Známky: E 50%, D 60%, C 70%, B 80%, A 90%.

Záver

1. cvičenie

- Začína hneď po prednáške.

Čo ďalej nasleduje?

- 2. prednáška: párovacia funkcia a aritmetizácia karteziánskeho súčinu.
- 3. prednáška: zoznamy, triedenie zoznamov, kombinatorické funkcie na zoznamoch.
- 4. prednáška: binárne stromy, binárne vyhľadávacie stromy.

Koniec prednášky