2-AIN-266 Deklaratívne programovanie

Letný semester 2021/22

1. prednáška

Ján Komara

Obsah 1. prednášky

Distančná výučba

Pokyny

- Vypnite si kameru a stlmte si mikrofón.
- Komunikovať je možné počas celého stretnutia.
- Ak máte otázku alebo chcete niečo povedať, tak zdvihnite ruku.
- Po vyzvaní zapnite si mikrofón.
- Po skončení ruku zložte a stlmte si mikrofón.
- Prezentácia je nahrávaná.
- Nahrávku zo stretnutia nájdete v tejto sekcii.
- Slajdy zo stretnutia nájdete pod položkou Súbory/Files.

Cieľ a obsah predmetu

Cieľ predmetu

- Vybudovať matematické základy deklaratívnych programovacích jazykov.

Kde hľadáme riešenie?

- V matematickej logike – v časti, ktorá sa volá *Teória rekurzívnych funkcií*.

Prečo?

- Churchova téza: trieda intuitívne vypočítateľných funkcií je totožná s obecne rekurzívnymi funkciami (Herbrand-Gödel).
- Návrh prvého jazyka deklaratívnej paradigmy jazyka Lisp bol silne ovplyvnený formalizmom čiastočne rekurzívnych funkcií.

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:
  - 1-AIN-470 Špecifikácia a verifikácia programov,
  - 2-AIN-285 Symbolické programovanie a LISP.
- Iné deklaratívne progamovacie jazyky:
  - funkcionálne programovacie jazyky – Lisp, 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.

Deklaratívne programovanie

Metamatematika deklaratívneho programovania

- Primitívne rekurzívne funkcie:
  - aritmetizácia dátových štruktúr, rekurzia s mierou.
- Obecne rekurzívne funkcie.
- Čiastočne rekurzívne funkcie:
  - Churchova téza.
- Algoritmicky nerozhodnuteľné problémy (problém zastavenia).
- Ohraničenie formálnych metód (uniformný problém zastavenia).

Primitívne rekurzívne funkcie

Druhy rekurzie

Primitívna rekurzia

- Umocňovanie:

$$x^0 = 1$$
$$x^{y+1} = x \cdot x^y.$$

(Course of values) rekurzia

- Fibonacciho postupnosť:

$$f_0 = 0$$
$$f_1 = 1$$
$$f_{n+2} = f_{n+1} + f_n.$$

Primitívne rekurzívne funkcie

Druhy rekurzie

Rekurzia s mierou

- Euklidov algoritmus:

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

Obecná rekurzia

- Ackermannova funkcia (1928).
- Univerzálna funkcia pre primitívne rekurzívne funkcie.

Primitívne rekurzívne funkcie

Definícia triedy primitívne rekurzívnych funkcií

- Základné funkcie:
  - konštantná funkcia $Z(x) = 0$,
  - funkcia nasledovníka $S(x) = x + 1$,
  - identity (projekcie):

$$I_i^n(x_1, \ldots, x_n) = x_i$$

pre každé $1 \leq i \leq n$.

- Kompozícia (skladanie) funkcií:

$$f(x_1, \ldots, x_n) = h\big(g_1(x_1, \ldots, x_n), \ldots, g_m(x_1, \ldots, x_n)\big).$$

- Primitívna rekurzia:

$$f(0, y_1, \ldots, y_n) = g(y_1, \ldots, y_n)$$
$$f\big(S(x), y_1, \ldots, y_n\big) = h\big(x, f(x, y_1, \ldots, y_n), y_1, \ldots, y_n\big).$$

Primitívne rekurzívne funkcie

Definícia triedy primitívne rekurzívnych funkcií

- Základné funkcie:
  - konštantná funkcia $\mathsf{Z}(x) = 0$,
  - funkcia nasledovníka $\mathsf{S}(x) = x + 1$,
  - identity (projekcie):

$$\mathsf{I}_i^n(\vec{x}) = x_i$$

pre každé $1 \leq i \leq n$.

- Kompozícia (skladanie) funkcií:

$$f(\vec{x}) = h\big(g_1(\vec{x}), \ldots, g_m(\vec{x})\big).$$

- Primitívna rekurzia:

$$f(0, \vec{y}) = g(\vec{y})$$
$$f\big(\mathsf{S}(x), \vec{y}\big) = h\big(x, f(x, \vec{y}), \vec{y}\big).$$

Primitívne rekurzívne funkcie

Definícia

- Základné funkcie: $\mathsf{Z}(x) = 0$, $\mathsf{S}(x) = x + 1$, $\mathsf{I}_i^n(\vec{x}) = x_i$.
- Kompozícia funkcií: $f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x}))$.
- Primitívna rekurzia:

$$f(0, \vec{y}) = g(\vec{y}) \qquad f(\mathsf{S}(x), \vec{y}) = h(x, f(x, \vec{y}), \vec{y}).$$

Sčítanie

$$0 + y = y$$
$$x + 1 + y = x + y + 1.$$

Primitívne rekurzívne funkcie

Definícia

- Základné funkcie: $\mathsf{Z}(x) = 0$, $\mathsf{S}(x) = x + 1$, $\mathsf{I}_i^n(\vec{x}) = x_i$.
- Kompozícia funkcií: $f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x}))$.
- Primitívna rekurzia:

$$f(0, \vec{y}) = g(\vec{y}) \qquad f(\mathsf{S}(x), \vec{y}) = h(x, f(x, \vec{y}), \vec{y}).$$

Sčítanie

$$h(x, z, y) = \mathsf{S}\,\mathsf{I}_2^3(x, z, y)$$

$$\begin{aligned} 0 + y &= \mathsf{I}(y) \\ \mathsf{S}(x) + y &= h(x, x + y, y). \end{aligned}$$

Primitívne rekurzívne funkcie

Definícia

- Základné funkcie: $\mathsf{Z}(x) = 0$, $\mathsf{S}(x) = x + 1$, $\mathsf{I}_i^n(\vec{x}) = x_i$.
- Kompozícia funkcií: $f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x}))$.
- Primitívna rekurzia:

$$f(0, \vec{y}) = g(\vec{y}) \qquad f(\mathsf{S}(x), \vec{y}) = h(x, f(x, \vec{y}), \vec{y}).$$

Násobenie

$$0 \cdot y = 0$$
$$(x + 1) \cdot y = x \cdot y + y.$$

Primitívne rekurzívne funkcie

Definícia

- Základné funkcie: $\mathsf{Z}(x) = 0$, $\mathsf{S}(x) = x + 1$, $\mathsf{I}_i^n(\vec{x}) = x_i$.
- Kompozícia funkcií: $f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x}))$.
- Primitívna rekurzia:

$$f(0, \vec{y}) = g(\vec{y}) \qquad f(\mathsf{S}(x), \vec{y}) = h(x, f(x, \vec{y}), \vec{y}).$$

Násobenie

$$h_2(x, z, y) = \mathsf{I}_2^3(x, z, y) + \mathsf{I}_3^3(x, z, y)$$

$$\begin{aligned} 0 \cdot y &= \mathsf{Z}(y) \\ \mathsf{S}(x) \cdot y &= h_2(x, x \cdot y, y). \end{aligned}$$

Primitívne rekurzívne funkcie

Definícia

- Základné funkcie: $\mathsf{Z}(x) = 0$, $\mathsf{S}(x) = x + 1$, $\mathsf{I}_i^n(\vec{x}) = x_i$.
- Kompozícia funkcií: $f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x}))$.
- Primitívna rekurzia:

$$f(0, \vec{y}) = g(\vec{y}) \qquad f(\mathsf{S}(x), \vec{y}) = h(x, f(x, \vec{y}), \vec{y}).$$

Umocňovanie

$$x^0 = 1$$
$$x^{y+1} = x \cdot x^y.$$

Primitívne rekurzívne funkcie

Definícia

- Základné funkcie: $\mathsf{Z}(x) = 0$, $\mathsf{S}(x) = x + 1$, $\mathsf{I}_i^n(\vec{x}) = x_i$.
- Kompozícia funkcií: $f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x}))$.
- Primitívna rekurzia:

$$f(0, \vec{y}) = g(\vec{y}) \qquad f(\mathsf{S}(x), \vec{y}) = h(x, f(x, \vec{y}), \vec{y}).$$

Umocňovanie

$$f(0, x) = 1$$
$$f(y + 1, x) = x \cdot f(y, x)$$

$$x^y = f(y, x).$$

Primitívne rekurzívne funkcie

Definícia

- Základné funkcie: $\mathsf{Z}(x) = 0$, $\mathsf{S}(x) = x + 1$, $\mathsf{I}_i^n(\vec{x}) = x_i$.
- Kompozícia funkcií: $f(\vec{x}) = h(g_1(\vec{x}), \ldots, g_m(\vec{x}))$.
- Primitívna rekurzia:

$$f(0, \vec{y}) = g(\vec{y}) \qquad f(\mathsf{S}(x), \vec{y}) = h(x, f(x, \vec{y}), \vec{y}).$$

Umocňovanie

$$\mathsf{C}_1(x) = \mathsf{S}\,\mathsf{Z}(x)$$

$$h_3(y, z, x) = \mathsf{I}_3^3(y, z, x) \cdot \mathsf{I}_2^3(y, z, x)$$

$$f(0, x) = \mathsf{C}_1(x)$$
$$f(\mathsf{S}(y), x) = h_3(y, f(y, x), x)$$

$$x^y = f(\mathsf{I}_2^2(x, y), \mathsf{I}_1^2(x, y)).$$

Primitívne rekurzívne funkcie

Primitívne rekurzívne funkcie a deklaratívne programovanie

- Základný vývoj primitívne rekurzívnych funkcií.
- Primitívne rekurzívne predikáty a ohraničená minimalizácia.
- Párovacia funkcia a aritmetizácia dátových štruktúr.
- Vnorená jednoduchá rekurzia:

$$f(0, y) = g(y)$$
$$f(x+1, y) = h\Big(x, f\big(x, s_1(x, y)\big), f\big(x, s_2(x, y, f(x, s_1(x, y))), y\Big).$$

- Regulárne rekurzívne definície s mierou:

$$f(\vec{x}) = \tau[f; \vec{x}].$$

Podmienka regularity $\Gamma_{f(\vec{\rho})} \to \mu[\vec{\rho}] < \mu[\vec{x}]$ pre rekurzívne volanie $f(\vec{\rho})$ funkcie f v terme τ.

Primitívne rekurzívne funkcie

Primitívne rekurzívne funkcie a deklaratívne programovanie

- Základný vývoj primitívne rekurzívnych funkcií.
- Primitívne rekurzívne predikáty a ohraničená minimalizácia.
- Párovacia funkcia a aritmetizácia dátových štruktúr.
- Vnorená jednoduchá rekurzia:

$$f(0, y) = g(y)$$
$$f(x+1, y) = h\Big(x, f\big(x, s_1(x, y)\big), f\big(x, s_2(x, y, f(x, s_1(x, y))), y\Big).$$

- Regulárne rekurzívne definície s mierou:

$$f(\vec{x}) = \tau[f; \vec{x}].$$

Podmienka regularity $\Gamma_{f(\vec{\rho})} \to \mu[\vec{\rho}] < \mu[\vec{x}]$ pre rekurzívne volanie $f(\vec{\rho})$ funkcie f v terme τ.

Záver

Organizácia kurzu

- Prednášky: pondelok 10.40 2h, M-II alebo online.
- Cvičenia: pondelok 12.10 2h, I-H6 alebo online.
- Konzultácie: po dohode.
- Web: http://ii.fmph.uniba.sk/cl/courses/

Hodnotenie

- Cvičenia 50% – max. 50 bodov.
- Testy, prémiové úlohy 50% – max. 50 bodov.
- Známky: E 50%, D 60%, C 70%, B 80%, A 90%.

Záver

1. cvičenie

- Začne hneď po prednáške.
- Úlohy odovzdať do MOODLE najneskôr do 18:00 v sobotu tento týždeň.

2. prednáška

- Základný vývoj primitívne rekurzívnych funkcií:
  - explicitné definície,
  - primitívne rekurzívne definície.

Koniec prednášky