1-AIN-625 Úvod do matematickej logiky pre programátorov
1-INF-450 Logika pre informatikov

Zimný semester 2011/12

1. prednáška

Ján Komara

Obsah 1. prednášky

Základné informácie o kurze

Organizácia kurzu

- Web: http://ii.fmph.uniba.sk/cl/courses/
- Prednášky: utorok 14:50-16:20, m. F1-108
- Cvičenia: utorok 16:30-18:00, m. I-H3
- Konzultácie: streda 14:00-14:30, m. I-16
- E-mail: komara@fmph.uniba.sk

Hodnotenie

- Cvičenia 20% – max. 20 bodov
- Test, prémiové úlohy 20% – max. 20 bodov
- Požiadavka priebežného hodnotenia: semester aspoň 20 bodov
- Ústna skúška 60%
- Známky: E 50%, D 60%, C 70%, B 80%, A 90%

Základné informácie o kurze

Podmieňujúce predmety (doporučené)

- 1-AIN-505 Úvod do deklaratívneho programovania
- 1-INF-465 Deklaratívne programovanie

Nadväzujúce predmety

- 1-AIN-470 Špecifikácia a verifikácia programov
- 2-AIN-130 Teória vypočítateľnosti pre programátorov

Softvér

- Pri výučbe používame prostredie programovacieho jazyka a špecifikačno-verifikačného systému CL

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

Cieľ a obsah predmetu

Deklaratívne programovanie

- Jednoduchá sémantika sa kombinuje s expresívnymi programátorskými konštrukciami
- 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
- Deklaratívne progamovacie jazyky
  - funkcionálne – LISP, HASKELL
  - logické – PROLOG

Cieľ a obsah predmetu

Euklidov algoritmus deklaratívne

Špecifikácia

- Špecifikačný predikát

$$x \mid y \leftrightarrow \exists z\, y = x{\cdot}z$$

- Špecifikácia programu, ktorý počíta funkciu gcd

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

- Idea algoritmu

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

Cieľ a obsah predmetu

Euklidov algoritmus deklaratívne

Implementácia

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

Výpočet

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

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

Cieľ a obsah predmetu

Euklidov algoritmus deklaratívne

Implementácia

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

Výpočet

Podmienky regularity zaručujú, že výpočet vždy skončí

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

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

Cieľ a obsah predmetu

Euklidov algoritmus deklaratívne

Implementácia

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

Výpočet

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

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

Alternatívna implementácia so vstupnou podmienkou

$$x \neq 0 \wedge y \neq 0 \rightarrow \gcd(x, y) = \textbf{case}$$
$$x > y \Rightarrow \gcd(x \dot{-} y, y)$$
$$x = y \Rightarrow x$$
$$x < y \Rightarrow \gcd(x, y \dot{-} x)$$
$$\textbf{end}$$

Výpočet

Ten nemusí skončiť:

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

Program počíta čiastočnú funkciu!

Cieľ a obsah predmetu

Euklidov algoritmus deklaratívne

Alternatívna implementácia so vstupnou podmienkou

$$x \neq 0 \wedge y \neq 0 \rightarrow \gcd(x, y) = \textbf{case}$$
$$x > y \Rightarrow \gcd(x \dot{-} y, y)$$
$$x = y \Rightarrow x$$
$$x < y \Rightarrow \gcd(x, y \dot{-} x)$$
$$\textbf{end}$$

Výpočet

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

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

Výpočet skončí pre vstupy spĺňajúce vstupnú podmienku

Cieľ a obsah predmetu

Zhrnutie

Pri našom prístupe k deklaratívnemu programovaniu:

- *Univerzum* je množina prirodzených čísel $\mathrm{N} = \{0, 1, 2, 3, 4, \dots\}$
- *Dátové štruktúry* kódujeme do N v štýle jazyka LISP s pomocou vhodnej párovacej funkcie
- *Programy* sú vlastnosti totálnych funkcií nad oborom N spĺňajúce určité vstupné podmienky
- *Formálny systém* je druhorádová formalizácia aritmetiky

Stručná osnova predmetu

- Primitívne rekurzívne funkcie
- Obecne rekurzívne funkcie
- Čiastočne rekurzívne funkcie
  - rekurzívne nerozhodnuteľné problémy – problém zastavenia

Primitívne rekurzívne funkcie

Typy rekurzie

Primitívna rekurzia

- Príklad (umocňovanie)

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

Obecná rekurzia

- Rekurzia s mierou
  - Euklidov algoritmus
  - Fibonacciho postupnosť

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

- Transfinitná rekurzia
  - Ackermannova funkcia (1928)
  - Univerzálna funkcia pre primitívne rekurzívne funkcie

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)

$$\mathsf{I}_i^n(x_1, \ldots, x_n) = x_i$$

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

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

$$h(x, z, y) = \mathsf{S}\,\mathsf{I}_2^3(x, z, y)$$
$$0 + y = \mathsf{I}(y)$$
$$\mathsf{S}(x) + y = h(x, x + y, y)$$

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

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

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

$$0 \cdot y = \mathsf{Z}(y) \qquad \mathsf{S}(x) \cdot y = h(x, x \cdot y, y)$$

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:

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

Umocňovanie

$$x^0 = 1$$

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

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

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

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

$$f(\mathsf{S}(y), x) = h(y, f(y, x), x)$$

$$x^y = f(I_2^2(x, y), I_1^2(x, y))$$

Záver

1. cvičenie

- Začína hneď po prednáške v miestnosti I-H3
- Pracujte vo dvojiciach
- Úlohy odovzdať najneskôr do 12:00 v pondelok budúci týždeň

2. prednáška

- Univerzálna funkcia pre primitívne rekurzívne funkcie