

2-INF-264 Teória deklaratívneho programovania

Zimný semester 2015/16

8. prednáška

Ján Komara

Obsah 8. prednášky

Zopakovanie

Deklaratívne programovanie

Efektívne operácie na primitívne rekurzívnych indexoch

Záver

Zopakovanie

Cieľ predmetu

- ▶ Vybudovať matematické základy deklaratívnych programovacích jazykov.

Stručná osnova predmetu

- ▶ Primitívne rekurzívne funkcie
 - ▶ Aritmetizácia dátových štruktúr.
 - ▶ Regulárne rekurzívne definície s mierou.
- ▶ Obecne rekurzívne funkcie
 - ▶ Regulárne rekurzívne definície do dobre založených relácií:
 - ▶ Ackermannova funkcia (1928),
 - ▶ univerzálna funkcia pre primitívne rekurzívne funkcie.
- ▶ Čiastočne rekurzívne funkcie
 - ▶ Kleeneho prvá veta o rekurzii (veta o pevnom bode).
 - ▶ Kleeneho veta o normálnej forme.
 - ▶ Churchova téza a algoritmicky nerozhodnuteľné problémy.

Zopakovanie

Dobre založené relácie

Vravíme, že relácia $\vec{x} \prec \vec{y}$ na \mathbb{N}^n je dobre založená (tiež noetherianská), ak každá klesajúca postupnosť

$$\vec{x}_1 \succ \vec{x}_2 \succ \vec{x}_3 \succ \vec{x}_4 \succ \dots$$

je konečná. Tu $\vec{x} \succ \vec{y} \leftrightarrow \vec{y} \prec \vec{x}$.

Úplné dobre založené usporiadanie sa nazýva dobré usporiadanie.

Príklady

- ▶ Relácia $\vec{x} \prec \vec{y}$ indukovaná mierou $\mu[\vec{x}]$ do štandardného usporiadania prirodzených čísel:

$$\vec{x} \prec \vec{y} \leftrightarrow \mu[\vec{x}] < \mu[\vec{y}].$$

- ▶ Lexikografické usporiadanie dvojíc prirodzených čísel:

$$(a, b) <_{\text{lex}} (c, d) \leftrightarrow a < c \vee a = c \wedge b < d.$$

Zopakovanie

Regulárna rekúzia do dobre založenej relácie \prec

Sú to definície v tvare

$$f(\vec{x}) = \tau[f; \vec{x}], \quad (1)$$

pre ktoré sú splnené podmienky regularity.

Podmienky regularity majú tvar:

$$\Gamma_{f(\vec{\rho})}^{\tau}[f; \vec{x}] \rightarrow \vec{\rho}[f; \vec{x}] \prec \vec{x}$$

pre každú rekurzívnu aplikáciu $f(\vec{\rho})$ funkcie f , ktorá je strážená podmienkou $\Gamma_{f(\vec{\rho})}^{\tau}$ v terme τ .

Splnenie podmienok pre funkciu definovanou pridruženou rovnosťou

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

Táto funkcia je potom jediným riešením funkcionálnej rovnice (1).

Zopakovanie

Primitívne rekurzívne funkcie (charakterizácia)

Trieda primitívne rekurzívnych funkcií je najmenšia trieda funkcií, ktorá obsahuje funkciu nasledovníka $S(x) = x + 1$, funkciu predchodcu $x \div 1$ a je uzavretá na explicitné definície a regulárne rekurzívne definície s mierou.

Obecne rekurzívne funkcie (definícia)

Triedaobecne rekurzívnych funkcií je najmenšia trieda funkcií, ktorá obsahuje funkciu nasledovníka $S(x) = x + 1$, funkciu predchodcu $x \div 1$ a je uzavretá na explicitné definície a regulárne rekurzívne definície do dobre založených relácií.

Obecne rekurzívne funkcie, ktoré nie sú primitívne rekurzívne

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

Zopakovanie

Regulárna minimalizácia

Sú to definície funkcií v tvare

$$f(x_1, \dots, x_n) = \text{najmenšie číslo } y \text{ také, že platí } \varphi[x_1, \dots, x_n, y],$$

kde φ je ohraničená formula spĺňajúca podmienku regularity

$$\forall x_1 \dots \forall x_n \exists y \varphi[x_1, \dots, x_n, y].$$

Skrátený zápis

$$f(x_1, \dots, x_n) = \mu y [\varphi[x_1, \dots, x_n, y]].$$

V nasledujúcej definícii pod operátorom regulárnej minimalizácie rozumieme regulárnu minimalizáciu v tomto špeciálnom tvare:

$$f(x_1, \dots, x_n) = \mu y [g(y, x_1, \dots, x_n) = 1].$$

Zopakovanie

μ -rekurzívne funkcie

Trieda μ -rekurzívnych funkcií je najmenšia primitívne rekurzívne uzavretá trieda funkcií, ktorá je uzavretá na operátor regulárnej minimalizácie.

Veta

Trieda obecné rekurzívnych funkcií je totožná s triedou μ -rekurzívnych funkcií.

Churchova téza (1936)

Trieda intuitívne vypočítateľných funkcií nad oborom prirodzených čísel je totožná s triedou obecné rekurzívnych funkcií.

Deklaratívne programovanie

Programy deklaratívnej paradigmy

Regulárne rekurzívne definície do dobre založených relácií:

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

Základné konštrukcie pri tvorbe programov

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

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

- ▶ Obecné podmienkové výrazy:

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

Deklaratívne programovanie

Diskriminácia na konštantách

Rekurzívna definícia Fibonacciho postupnosti:

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

Program:

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

Preklad do definície s jednoduchými podmienkovými výrazmi:

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

Deklaratívne programovanie

Diskriminácia na konštantách

Rekurzívna definícia Fibonacciho postupnosti:

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

Program:

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

Preklad do definície s jednoduchými podmienkovými výrazmi:

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

Deklaratívne programovanie

Dichotomická diskriminácia

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

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

Program:

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

Preklad do definície s jednoduchým podmienkovým výrazom:

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

Deklaratívne programovanie

Dichotomická diskriminácia

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

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

Program:

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

Preklad do definície s jednoduchým podmienkovým výrazom:

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

Deklaratívne programovanie

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):

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

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

Deklaratívne programovanie

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 $\varphi_i[\vec{x}]$:

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

Sémantika:

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

Deklaratívne programovanie

Monadická diskriminácia

Rekurzívna definícia umocňovania:

$$x^y = \text{case}$$
$$y = 0 \Rightarrow 1$$
$$y = z + 1 \Rightarrow x \cdot x^z$$
$$\text{end.}$$

Program:

$$x^y = \text{case}$$
$$(y =_* 0) = 1 \Rightarrow 1$$
$$(y \neq_* 0) = 1 \wedge y \dot{-} 1 = z \Rightarrow x \cdot x^z$$
$$\text{end.}$$

Preklad do definície s jednoduchým podmienkovým výrazom:

$$x^y = \text{if } (y =_* 0) \neq 0 \text{ then } 1 \text{ else } x \cdot x^{y \dot{-} 1}.$$

Deklaratívne programovanie

Monadická diskriminácia

Rekurzívna definícia umocňovania:

$$x^y = \text{case}$$
$$y = 0 \Rightarrow 1$$
$$y = z + 1 \Rightarrow x \cdot x^z$$
$$\text{end.}$$

Program:

$$x^y = \text{case}$$
$$(y =_* 0) = 1 \Rightarrow 1$$
$$(y \neq_* 0) = 1 \Rightarrow \text{let } z = y \div 1 \text{ in } x \cdot x^z$$
$$\text{end.}$$

Preklad do definície s jednoduchým podmienkovým výrazom:

$$x^y = \text{if } y \neq 0 \text{ then } x \cdot x^{y \div 1} \text{ else } 1.$$

Deklaratívne programovanie

Párová diskriminácia

Rekurzívna definícia párovej veľkosti prirodzeného čísla:

$$\begin{aligned} |x|_p &= \text{case} \\ &\quad x = 0 \Rightarrow 0 \\ &\quad x = \langle v, w \rangle \Rightarrow |v|_p + |w|_p + 1 \\ &\text{end.} \end{aligned}$$

Program:

$$\begin{aligned} |x|_p &= \text{case} \\ &\quad (x =_* 0) = 1 \Rightarrow 0 \\ &\quad (x \neq_* 0) = 1 \Rightarrow \text{let } v = \pi_1(x) \wedge w = \pi_2(x) \text{ in} \\ &\quad \quad |v|_p + |w|_p + 1 \\ &\text{end.} \end{aligned}$$

Preklad do definície s jednoduchým podmienkovým výrazom:

$$|x|_p = \text{if } x \neq 0 \text{ then } |\pi_1(x)|_p + |\pi_2(x)|_p + 1 \text{ else } 0.$$

Deklaratívne programovanie

Porovnávanie so vzorom ('pattern matching')

Formula $\varphi[\vec{x}; \vec{y}]$ spĺňa podmienku jednoznačnosti

$$\varphi[\vec{x}; \vec{y}] \wedge \varphi[\vec{x}; \vec{z}] \rightarrow \bigwedge_{i=1}^m y_i = z_i.$$

Tu \vec{x} resp. \vec{y} sú vstupné resp. výstupné premenné φ .

Charakteristický term $\chi[\vec{x}]$ pre porovnávanie so vzorom:

$$\chi[\vec{x}] = 1 \vee \chi[\vec{x}] = 0 \quad \exists \vec{y} \varphi[\vec{x}; \vec{y}] \leftrightarrow \chi[\vec{x}] = 1.$$

Svedčiace termy $\vec{\omega}[\vec{x}]$ pre výstupné premenné:

$$\exists \vec{y} \varphi[\vec{x}; \vec{y}] \leftrightarrow \varphi[\vec{x}; \vec{\omega}[\vec{x}]].$$

Platí

$$\varphi[\vec{x}; \vec{y}] \leftrightarrow \chi[\vec{x}] = 1 \wedge \bigwedge_{i=1}^m \omega_i[\vec{x}] = y_i.$$

Deklaratívne programovanie

Podmienkové výrazy

Syntax:

case

$$\varphi_1[\vec{x}; \vec{y}_1] \Rightarrow \rho_1[\vec{x}, \vec{y}_1]$$

\vdots

$$\varphi_m[\vec{x}; \vec{y}_m] \Rightarrow \rho_m[\vec{x}; \vec{y}_m]$$

end

case

$$\chi_1[\vec{x}] = 1 \wedge \vec{\omega}_1[\vec{x}] = \vec{y}_1 \Rightarrow \rho_1[\vec{x}, \vec{y}_1]$$

\vdots

$$\chi_m[\vec{x}] = 1 \wedge \vec{\omega}_m[\vec{x}] = \vec{y}_m \Rightarrow \rho_m[\vec{x}, \vec{y}_m]$$

end.

Podmienka úplnosti a jednoznačnosti (výlučnosti):

$$\bigvee_{i=1}^m \exists \vec{y}_i \varphi_i[\vec{x}; \vec{y}_i] \wedge \bigwedge_{i,j=1}^m \forall \vec{y}_i \forall \vec{y}_j (\varphi_i[\vec{x}; \vec{y}_i] \wedge \varphi_j[\vec{x}; \vec{y}_j] \rightarrow \rho_i[\vec{x}, \vec{y}_i] = \rho_j[\vec{x}, \vec{y}_j])$$

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

Deklaratívne programovanie

Podmienkové výrazy

Syntax:

case

$$\varphi_1[\vec{x}; \vec{y}_1] \Rightarrow \rho_1[\vec{x}, \vec{y}_1]$$

\vdots

$$\varphi_m[\vec{x}; \vec{y}_m] \Rightarrow \rho_m[\vec{x}; \vec{y}_m]$$

end

case

$$\chi_1[\vec{x}] = 1 \wedge \vec{\omega}_1[\vec{x}] = \vec{y}_1 \Rightarrow \rho_1[\vec{x}, \vec{y}_1]$$

\vdots

$$\chi_m[\vec{x}] = 1 \wedge \vec{\omega}_m[\vec{x}] = \vec{y}_m \Rightarrow \rho_m[\vec{x}, \vec{y}_m]$$

end.

$\chi_i[\vec{x}]$ je charakteristický term pre porovnávanie so vzorom $\varphi_i[\vec{x}; \vec{y}_i]$:

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

Sémantika:

$$D\left(\chi_1[\vec{x}], \rho_1[\vec{x}, \vec{\omega}_1[\vec{x}]], \dots, D(\chi_m[\vec{x}], \rho_m[\vec{x}, \vec{\omega}_m[\vec{x}]], 0) \dots\right).$$

Deklaratívne programovanie

Podmienkové výrazy

Syntax:

case

$$\varphi_1[\vec{x}; \vec{y}_1] \Rightarrow \rho_1[\vec{x}, \vec{y}_1]$$

\vdots

$$\varphi_m[\vec{x}; \vec{y}_m] \Rightarrow \rho_m[\vec{x}, \vec{y}_m]$$

end

case

$$\chi_1[\vec{x}] = 1 \wedge \vec{\omega}_1[\vec{x}] = \vec{y}_1 \Rightarrow \rho_1[\vec{x}, \vec{y}_1]$$

\vdots

$$\chi_m[\vec{x}] = 1 \wedge \vec{\omega}_m[\vec{x}] = \vec{y}_m \Rightarrow \rho_m[\vec{x}, \vec{y}_m]$$

end.

Matematická notácia:

$$\mathcal{D}_{\chi_1, \dots, \chi_m}^{\vec{\omega}_1, \dots, \vec{\omega}_m}(\varphi_1[\vec{x}; \vec{y}_1], \rho_1[\vec{x}, \vec{y}_1], \dots, \varphi_m[\vec{x}; \vec{y}_m], \rho_m[\vec{x}, \vec{y}_m]).$$

Efektívne operácie na primitívne rekurzívnych indexoch

Špecifikácia interpretra

Interpreter programovacieho jazyka p.r. odvedení je binárna funkcia $e \bullet x$, ktorá má tieto dve základné vlastnosti:

- ▶ Pre každý n -árny p.r. funkčný symbol f a n -ticu čísel x_1, \dots, x_n platí rovnosť:

$$\lceil f \rceil \bullet \langle x_1, \dots, x_n \rangle = f^{\mathcal{N}}(x_1, \dots, x_n).$$

Tu $\lceil f \rceil \in \mathbb{N}$ je kód a $f^{\mathcal{N}} : \mathbb{N}^n \rightarrow \mathbb{N}$ interpretácia symbolu f .

- ▶ Pre každé číslo e , unárna funkcia f definovaná vzťahom

$$f(x) = e \bullet x$$

je primitívne rekurzívna funkcia.

Primitívne rekurzívne odvedenia sú reprezentované p.r. funkčnými symbolmi.

Efektívne operácie na primitívne rekurzívnych indexoch

Implementácia interpretra

Diskriminácia podľa konštruktorov p.r. funkčných symbolov:

$e \bullet x = \text{case}$

$$e = Z \Rightarrow 0$$

$$e = S \Rightarrow x + 1$$

$$e = I_i^n \Rightarrow [x]_i^n$$

$$e = \langle g, gs \rangle \Rightarrow \langle g \bullet x, gs \bullet x \rangle$$

$$e = \text{Comp}_m^n(h, gs) \Rightarrow h \bullet (gs \bullet x)$$

$$e = \text{Rec}_n(g, h) \Rightarrow$$

case

$$x = \langle 0, y \rangle \Rightarrow g \bullet y$$

$$x = \langle z + 1, y \rangle \Rightarrow h \bullet \langle z, \text{Rec}_n(g, h) \bullet \langle z, y \rangle, y \rangle$$

$$\text{otherwise} \Rightarrow 0$$

end

$$\text{otherwise} \Rightarrow 0$$

end.

Efektívne operácie na primitívne rekurzívnych indexoch

Implementácia interpretra

Definícia interpretra pomocou vnorenej dvojitej rekurzíe:

$$\mathbf{Z} \bullet x = 0$$

$$\mathbf{S} \bullet x = x + 1$$

$$\mathbf{I}_i^n \bullet x = [x]_i^n$$

$$\langle g, gs \rangle \bullet x = \langle g \bullet x, gs \bullet x \rangle$$

$$\mathbf{Comp}_m^n(h, gs) \bullet x = h \bullet (gs \bullet x)$$

$$\mathbf{Rec}_n(g, h) \bullet \langle 0, y \rangle = g \bullet y$$

$$\mathbf{Rec}_n(g, h) \bullet \langle x + 1, y \rangle = h \bullet \langle x, \mathbf{Rec}_n(g, h) \bullet \langle x, y \rangle, y \rangle.$$

Podmienky regularity pre funkciu $e \bullet x$ sú triviálne splnené, napr.

$$(\mathbf{Rec}_n(g, h), \langle x, y \rangle) <_{\text{lex}} (\mathbf{Rec}_n(g, h), \langle x + 1, y \rangle)$$

$$(h, \langle x, \mathbf{Rec}_n(g, h) \bullet \langle x, y \rangle, y \rangle) <_{\text{lex}} (\mathbf{Rec}_n(g, h), \langle x + 1, y \rangle).$$

Efektívne operácie na primitívne rekurzívnych indexoch

Primitívne rekurzívne indexy

Prirodzené číslo e také, že

$$\forall x_1 \dots \forall x_n f(x_1, \dots, x_n) = e \bullet \langle x_1, \dots, x_n \rangle,$$

sa nazýva primitívne rekurzívny index funkcie f .

Veta

Funkcia je primitívne rekurzívna práve vtedy, keď má primitívne rekurzívny index.

Dobre vytvorené primitívne rekurzívne indexy

Predikát $Prf(n, e)$ platí, ak číslo e je dobre vytvorený primitívne rekurzívny index nejakej n -árnej p.r. funkcie, t.j. $e = \ulcorner f \urcorner$ pre nejaké $f \in PR^n$.

Efektívne operácie na primitívne rekurzívnych indexoch

Príklad

Primitívne rekurzívne odvedenie operácie sčítania

$$0 + y = y$$

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

$$h(x, z, y) = S I_2^3(x, z, y)$$

$$0 + y = I(y)$$

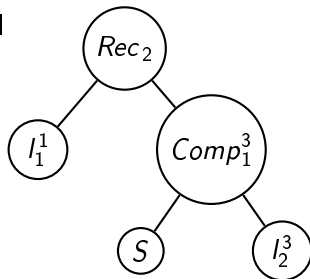
$$S(x) + y = h(x, x + y, y)$$

Primitívne rekurzívny funkčný symbol

$$Rec_2(I_1^1, Comp_1^3(S, I_2^3))$$

a jeho aritmetizácia (p.r. index)

$$Rec_2(I_1^1, Comp_1^3(S, I_2^3)).$$



Efektívne operácie na primitívne rekurzívnych indexoch

Konštantné funkcie

Špecifikácia unárnej p.r. funkcie C_m :

$$\text{Prf}(1, C_m) \wedge \forall x C_m \bullet x = m.$$

Návod na riešenie:

$$C_0 = Z \qquad C_{m+1}(x) = S C_m(x).$$

Špecifikácia binárnej p.r. funkcie C_m^n : pre každé $n \geq 1$ platí

$$\text{Prf}(n, C_m^n) \wedge \forall x_1 \dots \forall x_n C_m^n \bullet \langle x_1, \dots, x_n \rangle = m.$$

Návod na riešenie:

$$C_m^n(x_1, \dots, x_n) = C_m \uparrow_1^n(x_1, \dots, x_n).$$

Efektívne operácie na primitívne rekurzívnych indexoch

Konštantné funkcie

Špecifikácia unárnej p.r. funkcie C_m :

$$\text{Prf}(1, C_m) \wedge \forall x C_m \bullet x = m.$$

Riešenie:

$$C_0 = Z \qquad C_{m+1} = \text{Comp}_1^1(S, C_m).$$

Špecifikácia binárnej p.r. funkcie C_m^n : pre každé $n \geq 1$ platí

$$\text{Prf}(n, C_m^n) \wedge \forall x_1 \dots \forall x_n C_m^n \bullet \langle x_1, \dots, x_n \rangle = m.$$

Riešenie:

$$C_m^n = \text{Comp}_1^n(C_m, I_1^n).$$

Efektívne operácie na primitívne rekurzívnych indexoch

Iterácia unárnej funkcie

Špecifikácia unárnej p.r. funkcie $Iter(e)$:

$$Prf(1, e) \rightarrow Prf(2, Iter(e))$$

$$Iter(e) \bullet \langle 0, x \rangle = x$$

$$Iter(e) \bullet \langle n + 1, x \rangle = e \bullet (Iter(e) \bullet \langle n, x \rangle).$$

Návod na riešenie. Ak e je p.r. index unárnej funkcie f , potom $Iter(e)$ je p.r. index jej iterácie $f^n(x)$:

$$f^0(x) = x$$

$$f^{n+1}(x) = f f^n(x).$$

Riešenie:

$$Iter(e) = Rec_2(I_1^1, Comp_1^3(e, I_2^3)).$$

Efektívne operácie na primitívne rekurzívnych indexoch

Bezparametrická primitívna rekurgia

Špecifikácia binárnej p.r. funkcie $\mathbf{Rec0}(m, e)$:

$$\mathit{Prf}(2, e) \rightarrow \mathit{Prf}(1, \mathbf{Rec0}(m, e))$$

$$\mathbf{Rec0}(m, e) \bullet 0 = m$$

$$\mathbf{Rec0}(m, e) \bullet (x + 1) = e \bullet \langle x, \mathbf{Rec0}(m, e) \bullet x \rangle.$$

Návod na riešenie. Ak e je p.r. index binárnej funkcie h , potom $\mathbf{Rec0}(m, e)$ je p.r. index unárnej funkcie f :

$$f(0) = m$$

$$f(x + 1) = h(x, f(x)).$$

Riešenie:

$$\mathbf{Rec0}(m, e) = \mathit{Comp}_2^1 \left(\mathit{Rec}_2 \left(C_m, \mathit{Comp}_2^3(e, \langle I_1^3, I_2^3 \rangle) \right), \langle I_1^1, C_0 \rangle \right).$$

Efektívne operácie na primitívne rekurzívnych indexoch

Parametrická funkcia

Špecifikácia binárnej p.r. funkcie e/x :

$$\text{Prf}(2, e) \rightarrow \text{Prf}(1, e/x)$$

$$(e/x) \bullet y = e \bullet \langle x, y \rangle.$$

Návod na riešenie. Ak e je p.r. index binárnej funkcie f a $x \in \mathbb{N}$, potom e/x je p.r. index unárnej funkcie g :

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

Platí teda

$$g(y) = f(C_x(y), I_1(y)).$$

Riešenie:

$$e/x = \text{Comp}_2^1(e, \langle C_x, I_1^1 \rangle).$$

Záver

7. cvičenie

- ▶ Úlohy odovzdať najneskôr do 12:00 v nedeľu tento týždeň.

8. cvičenie

- ▶ Pracujte samostatne.
- ▶ Úlohy odovzdať najneskôr do 12:00 v nedeľu budúci týždeň.

9. prednáška

- ▶ Kleeneho prvá veta o rekurzii (veta o pevnom bode).
- ▶ Čiastočne rekurzívne funkcie.

9. cvičenie

- ▶ Semestrálny test.