# 1-INF-450 Logika pre informatikov

## Zimný semester 2013/14
## 9. prednáška

Ján Komara

# Obsah 9. 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 $\mathrm{N}^n$ je dobre založená (tiež noetheriánska), ak každá klesajúca postupnosť

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

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:

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

# Zopakovanie

## Regulárna rekurzia do dobre založenej relácie $\prec$

Sú to definície v tvare

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

pre ktoré sú splnené podmienky regularity.
Podmienky regularity majú tvar:

$$\Gamma^{\tau}_{f(\vec{\rho})}[f;\vec{x}] \to \vec{\rho}[f;\vec{x}] \prec \vec{x}$$

pre každú rekurzívnu aplikáciu $f(\vec{\rho})$ funkcie $f$, ktorá je strážená podmienkou $\Gamma^{\tau}_{f(\vec{\rho})}$ v terme $\tau$.
Splnenie podmienok pre funkciu definovanou pridruženou rovnosťou

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

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 $x \dot{-} 1$ a je uzavretá na explicitné definície a regulárne rekurzívne definície s mierou.

## Obecne rekurzívne funkcie (definícia)

Trieda obecne rekurzívnych funkcií je najmenšia trieda funkcií, ktorá obsahuje funkciu nasledovníka $S(x) = x + 1$, funkciu predchodcu $x \dot{-} 1$ 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

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

kde $\varphi$ je ohraničená formula spĺňajúca podmienku regularity

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

Skrátený zápis

$$f(x_1, \ldots, x_n) = \mu y \left[\varphi[x_1, \ldots, x_n, y]\right].$$

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

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

# Zopakovanie

### μ-rekurzívne funkcie

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

### Veta

*Trieda obecne rekurzívnych funkcií je totožná s triedou μ-rekurzívnych funkcií.*

### Churchova téza (1936)

*Trieda intuitívne vypočítateľných funkcií nad oborom prirodzených čísel je totožná s triedou obecne 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:

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

  - Obecné podmienkové výrazy:

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

## Diskriminácia na konštantách

Rekurzívna definícia Fibonacciho postupnosti:

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

Program:

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

Preklad do definície s jednoduchými podmienkovými výrazmi:

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

# Deklaratívne programovanie

## Diskriminácia na konštantách

Rekurzívna definícia Fibonacciho postupnosti:

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

Program:

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

Preklad do definície s jednoduchými podmienkovými výrazmi:

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

## 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 $\max(x, y)$:

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

Program:

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

Preklad do definície s jednoduchým podmienkovým výrazom:

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

# Deklaratívne programovanie

## Podmienkové výrazy

Syntax:

$$
\begin{array}{ll}
\textbf{case} & \\
\quad \varphi_1[\vec{x}] \Rightarrow \rho_1[\vec{x}] & \\
\quad \vdots & \\
\quad \varphi_m[\vec{x}] \Rightarrow \rho_m[\vec{x}] & \\
\textbf{end} &
\end{array}
\qquad\qquad
\begin{array}{l}
\textbf{case} \\
\quad \chi_1[\vec{x}] = 1 \Rightarrow \rho_1[\vec{x}] \\
\quad \vdots \\
\quad \chi_m[\vec{x}] = 1 \Rightarrow \rho_m[\vec{x}] \\
\textbf{end}.
\end{array}
$$

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:

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

## Podmienkové výrazy

Syntax:

$$\begin{array}{l} \textbf{case} \\ \quad \varphi_1[\vec{x}] \Rightarrow \rho_1[\vec{x}] \\ \quad \vdots \\ \quad \varphi_m[\vec{x}] \Rightarrow \rho_m[\vec{x}] \\ \textbf{end} \end{array} \qquad \begin{array}{l} \textbf{case} \\ \quad \chi_1[\vec{x}] = 1 \Rightarrow \rho_1[\vec{x}] \\ \quad \vdots \\ \quad \chi_m[\vec{x}] = 1 \Rightarrow \rho_m[\vec{x}] \\ \textbf{end}. \end{array}$$

Podmienka úplnosti a jednoznačnosti:

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

# Deklaratívne programovanie

## Podmienkové výrazy

Syntax:

$$
\begin{array}{l}
\textbf{case} \\
\quad \varphi_1[\vec{x}] \Rightarrow \rho_1[\vec{x}] \\
\quad \vdots \\
\quad \varphi_m[\vec{x}] \Rightarrow \rho_m[\vec{x}] \\
\textbf{end}
\end{array}
\qquad\qquad
\begin{array}{l}
\textbf{case} \\
\quad \chi_1[\vec{x}] = 1 \Rightarrow \rho_1[\vec{x}] \\
\quad \vdots \\
\quad \chi_m[\vec{x}] = 1 \Rightarrow \rho_m[\vec{x}] \\
\textbf{end}.
\end{array}
$$

Podmienka úplnosti a výlučnosti:

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

# Deklaratívne programovanie

## Monadická diskriminácia

Rekurzívna definícia umocňovania:

$$
\begin{aligned}
x^y = \textbf{case}\\
&\quad y = 0 \Rightarrow 1\\
&\quad y = z + 1 \Rightarrow x \cdot x^z\\
&\textbf{end}.
\end{aligned}
$$

Program:

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

Preklad do definície s jednoduchým podmienkovým výrazom:

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

# Deklaratívne programovanie

## Monadická diskriminácia

Rekurzívna definícia umocňovania:

$$
\begin{aligned}
x^y = \textbf{case}\\
&\quad y = 0 \Rightarrow 1\\
&\quad y = z + 1 \Rightarrow x \cdot x^z\\
&\textbf{end}.
\end{aligned}
$$

Program:

$$
\begin{aligned}
x^y = \textbf{case}\\
&\quad (y =_* 0) = 1 \Rightarrow 1\\
&\quad (y \neq_* 0) = 1 \Rightarrow \textbf{let } z = y \dot{-} 1 \textbf{ in } x \cdot x^z\\
&\textbf{end}.
\end{aligned}
$$

Preklad do definície s jednoduchým podmienkovým výrazom:

$$x^y = \textbf{if } y \neq 0 \textbf{ then } x \cdot x^{y \dot{-} 1} \textbf{ else } 1.$$

# Deklaratívne programovanie

## Párová diskriminácia

Rekurzívna definícia párovej veľkosti prirodzeného čísla:

$$
\begin{aligned}
|x|_{\mathrm{p}} = \ & \mathbf{case} \\
& \quad x = 0 \Rightarrow 0 \\
& \quad x = \langle v, w \rangle \Rightarrow |v|_{\mathrm{p}} + |w|_{\mathrm{p}} + 1 \\
& \mathbf{end}.
\end{aligned}
$$

Program:

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

Preklad do definície s jednoduchým podmienkovým výrazom:

$$|x|_{\mathrm{p}} = \mathbf{if}\ x \neq 0\ \mathbf{then}\ |\pi_1(x)|_{\mathrm{p}} + |\pi_2(x)|_{\mathrm{p}} + 1\ \mathbf{else}\ 0.$$

# Deklaratívne programovanie

## Porovnávanie so vzorom ('pattern matching')

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

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

Tu $\vec{x}$ resp. $\vec{y}$ sú vstúpne resp. výstupné premenné $\varphi$.
Charakteristický term $\chi[\vec{x}]$ pre porovnávanie so vzorom:

$$\chi[\vec{x}] = 1 \lor \chi[\vec{x}] = 0 \qquad \exists \vec{y}\, \varphi[\vec{x};\vec{y}] \leftrightarrow \chi[\vec{x}] = 1.$$

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

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

Platí

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

# 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**.

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

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

Sémantika:

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

# 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:

$$\bigvee_{i=1}^{m} \exists \vec{y}_i \varphi_i[\vec{x}; \vec{y}_1] \wedge \bigwedge_{i,j=1}^{m} \forall \vec{y}_i \forall \vec{y}_j \big(\varphi_i[\vec{x}; \vec{y}_i] \wedge \varphi_j[\vec{x}; \vec{y}_j] \to \rho_i[\vec{x}, \vec{y}_i] = \rho_j[\vec{x}, \vec{y}_j]\big).$$

Matematická notácia:

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

# Deklaratívne programovanie

## Podmienkové výrazy

Syntax:

$$\begin{array}{ll}
\textbf{case} & \textbf{case} \\
\quad \varphi_1[\vec{x};\vec{y}_1] \Rightarrow \rho_1[\vec{x},\vec{y}_1] & \quad \chi_1[\vec{x}] = 1 \wedge \vec{\omega}_1[\vec{x}] = \vec{y}_1 \Rightarrow \rho_1[\vec{x},\vec{y}_1] \\
\quad \vdots & \quad \vdots \\
\quad \varphi_m[\vec{x};\vec{y}_m] \Rightarrow \rho_m[\vec{x};\vec{y}_m] & \quad \chi_m[\vec{x}] = 1 \wedge \vec{\omega}_m[\vec{x}] = \vec{y}_m \Rightarrow \rho_m[\vec{x},\vec{y}_m] \\
\textbf{end} & \textbf{end}.
\end{array}$$

Podmienka úplnosti a výlučnosti:

$$\bigvee_{i=1}^{m} \exists \vec{y}_i \varphi_i[\vec{x};\vec{y}_1] \wedge \bigwedge_{\substack{i,j=1 \\ i\neq j}}^{m} \neg\big(\exists \vec{y}_i \varphi_i[\vec{x};\vec{y}_i] \wedge \exists \vec{y}_j \varphi_j[\vec{x};\vec{y}_j]\big).$$

Matematická notácia:

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

# Efektívne operácie na primitívne rekurzívnych indexoch

## Špecifikácia interpretra

Interpreter programovacieho jazyka p.r. odvodení 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 $x_1, \ldots, x_n$ platí rovnosť:

  $$\ulcorner f \urcorner \bullet \langle x_1, \ldots, x_n \rangle = f^{\mathcal{N}}(x_1, \ldots, x_n).$$

  Tu $\ulcorner f \urcorner \in \mathrm{N}$ je kód a $f^{\mathcal{N}} : \mathrm{N}^n \to \mathrm{N}$ interpretácia symbolu $f$.
- Pre každé číslo $e$, unárna funkcia $f$ definovaná vzťahom

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

  je primitívne rekurzívna funkcia.

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

# Efektívne operácie na primitívne rekurzívnych indexoch

## Implementácia interpretra

Diskriminácia podľa konštruktorov p.r. funkčných symbolov:

$$
\begin{array}{l}
e \bullet x = \textbf{case} \\
\qquad e = \boldsymbol{Z} \Rightarrow 0 \\
\qquad e = \boldsymbol{S} \Rightarrow x + 1 \\
\qquad e = \boldsymbol{I}^n_i \Rightarrow [x]^n_i \\
\qquad e = \langle g, gs \rangle \Rightarrow \big\langle g \bullet x, gs \bullet x \big\rangle \\
\qquad e = \boldsymbol{Comp}^n_m(h, gs) \Rightarrow h \bullet (gs \bullet x) \\
\qquad e = \boldsymbol{Rec}_n(g, h) \Rightarrow \\
\qquad\quad \textbf{case} \\
\qquad\qquad x = \langle 0, y \rangle \Rightarrow g \bullet y \\
\qquad\qquad x = \langle z + 1, y \rangle \Rightarrow h \bullet \big\langle z, \boldsymbol{Rec}_n(g, h) \bullet \langle z, y \rangle, y \big\rangle \\
\qquad\qquad \textbf{otherwise} \Rightarrow 0 \\
\qquad\quad \textbf{end} \\
\qquad \textbf{otherwise} \Rightarrow 0 \\
\textbf{end}.
\end{array}
$$

# Efektívne operácie na primitívne rekurzívnych indexoch

## Implementácia interpretra

Definícia interpretra pomocou vnorenej dvojitej rekurzie:

$$
\begin{aligned}
\boldsymbol{Z} \bullet x &= 0 \\
\boldsymbol{S} \bullet x &= x + 1 \\
\boldsymbol{I}_i^n \bullet x &= [x]_i^n \\
\langle g, gs \rangle \bullet x &= \big\langle g \bullet x, gs \bullet x \big\rangle \\
\boldsymbol{Comp}_m^n(h, gs) \bullet x &= h \bullet (gs \bullet x) \\
\boldsymbol{Rec}_n(g, h) \bullet \langle 0, y \rangle &= g \bullet y \\
\boldsymbol{Rec}_n(g, h) \bullet \langle x + 1, y \rangle &= h \bullet \big\langle x, \boldsymbol{Rec}_n(g, h) \bullet \langle x, y \rangle, y \big\rangle.
\end{aligned}
$$

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

$$
\begin{aligned}
\big(\boldsymbol{Rec}_n(g, h), \langle x, y \rangle\big) &<_{\mathrm{lex}} \big(\boldsymbol{Rec}_n(g, h), \langle x + 1, y \rangle\big) \\
\big(h, \big\langle x, \boldsymbol{Rec}_n(g, h) \bullet \langle x, y \rangle, y \big\rangle\big) &<_{\mathrm{lex}} \big(\boldsymbol{Rec}_n(g, h), \langle x + 1, y \rangle\big).
\end{aligned}
$$

# Efektívne operácie na primitívne rekurzívnych indexoch

## Primitívne rekurzívne indexy

Prirodzené číslo $e$ také, že

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

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 $f \in \mathrm{PR}^n$.

# Efektívne operácie na primitívne rekurzívnych indexoch

## Príklad

Primitívne rekurzívne odvodenie operácie sčítania

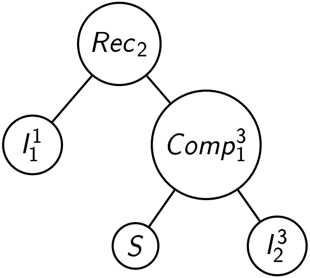
$$
\begin{aligned}
0 + y &= y \\
x + 1 + y &= x + y + 1
\end{aligned}
\qquad\qquad
\begin{aligned}
h(x, z, y) &= \mathsf{S}\,\mathsf{I}^3_2(x, z, y) \\
0 + y &= \mathsf{I}(y) \\
\mathsf{S}(x) + y &= h(x, x + y, y)
\end{aligned}
$$

Primitívne rekurzívny funkčný symbol

$$Rec_2(I^1_1, Comp^3_1(S, I^3_2))$$

a jeho aritmetizácia (p.r. index)

$$\boldsymbol{Rec}_2(\boldsymbol{I}^1_1, \boldsymbol{Comp}^3_1(\boldsymbol{S}, \boldsymbol{I}^3_2)).$$



# Efektívne operácie na primitívne rekurzívnych indexoch

## Konštantné funkcie

Špecifikácia unárnej p.r. funkcie $\boldsymbol{C}_m$:

$$Prf(1, \boldsymbol{C}_m) \wedge \forall x\, \boldsymbol{C}_m \bullet x = m.$$

Návod na riešenie:

$$\mathsf{C}_0 = \mathsf{Z} \qquad\qquad \mathsf{C}_{m+1}(x) = \mathsf{S}\,\mathsf{C}_m(x).$$

Špecifikácia binárnej p.r. funkcie $\boldsymbol{C}^n_m$:

$$Prf(n, \boldsymbol{C}^n_m) \wedge \forall x_1 \ldots \forall x_n\, \boldsymbol{C}^n_m \bullet \langle x_1, \ldots, x_n \rangle = m.$$

Návod na riešenie:

$$\mathsf{C}^n_m(x_1, \ldots, x_n) = \mathsf{C}_m\, \mathsf{I}^n_1(x_1, \ldots, x_n).$$

# Efektívne operácie na primitívne rekurzívnych indexoch

## Konštantné funkcie

Špecifikácia unárnej p.r. funkcie $\boldsymbol{C}_m$:

$$Prf(1, \boldsymbol{C}_m) \land \forall x\, \boldsymbol{C}_m \bullet x = m.$$

Riešenie:

$$\boldsymbol{C}_0 = \boldsymbol{Z} \qquad\qquad \boldsymbol{C}_{m+1} = \boldsymbol{Comp}^1_1(\boldsymbol{S}, \boldsymbol{C}_m).$$

Špecifikácia binárnej p.r. funkcie $\boldsymbol{C}^n_m$:

$$Prf(n, \boldsymbol{C}^n_m) \land \forall x_1 \ldots \forall x_n\, \boldsymbol{C}^n_m \bullet \langle x_1, \ldots, x_n \rangle = m.$$

Riešenie:

$$\boldsymbol{C}^n_m = \boldsymbol{Comp}^n_1(\boldsymbol{C}_m, \boldsymbol{I}^n_1).$$

# Efektívne operácie na primitívne rekurzívnych indexoch

## Iterácia unárnej funkcie

Špecifikácia unárnej p.r. funkcie $\boldsymbol{Iter}(e)$:

$$
\begin{gathered}
Prf(1, e) \rightarrow Prf\big(2, \boldsymbol{Iter}(e)\big) \\
\boldsymbol{Iter}(e) \bullet \langle 0, x\rangle = x \\
\boldsymbol{Iter}(e) \bullet \langle n+1, x\rangle = e \bullet \big(\boldsymbol{Iter}(e) \bullet \langle n, x\rangle\big).
\end{gathered}
$$

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

$$
\begin{gathered}
f^0(x) = x \\
f^{n+1}(x) = f\, f^n(x).
\end{gathered}
$$

Riešenie:

$$
\boldsymbol{Iter}(e) = \boldsymbol{Rec}_2\big(\boldsymbol{I}_1^1, \boldsymbol{Comp}_1^3(e, \boldsymbol{I}_2^3)\big).
$$

# Efektívne operácie na primitívne rekurzívnych indexoch

## Bezparametrická primitívna rekurzia

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

$$
\begin{aligned}
Prf(2, e) &\to Prf\big(1, \boldsymbol{Rec0}(m, e)\big) \\
\boldsymbol{Rec0}(m, e) \bullet 0 &= m \\
\boldsymbol{Rec0}(m, e) \bullet (x+1) &= e \bullet \big\langle x, \boldsymbol{Rec0}(m, e) \bullet x \big\rangle .
\end{aligned}
$$

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

$$
\begin{aligned}
f(0) &= m \\
f(x+1) &= h\big(x, f(x)\big).
\end{aligned}
$$

Riešenie:

$$
\boldsymbol{Rec0}(m, e) = \boldsymbol{Comp}_2^1\Big(\boldsymbol{Rec}_2\Big(\boldsymbol{C}_m, \boldsymbol{Comp}_2^3\big(e, \langle \boldsymbol{I}_1^3, \boldsymbol{I}_2^3 \rangle\big)\Big), \langle \boldsymbol{I}_1^1, \boldsymbol{C}_0 \rangle\Big).
$$

# Efektívne operácie na primitívne rekurzívnych indexoch

## Parametrická funkcia

Špecifikácia binárnej p.r. funkcie $e/x$:

$$Prf(2, e) \to Prf(1, e/x)$$
$$(e/x) \bullet y = e \bullet \langle x, y \rangle.$$

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

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

Platí teda

$$g(y) = f\big(\mathsf{C}_x(y), \mathsf{I}(y)\big).$$

Riešenie:

$$e/x = \boldsymbol{Comp}^1_2\big(e, \langle \boldsymbol{C}_x, \boldsymbol{I}^1_1 \rangle\big).$$

# Záver

## 8. cvičenie

- Výsledky s komentárom nájdete na webe.

## 9. cvičenie

- Začína v stredu o 09:50 v miestnosti F1-248.
- Pracujte samostatne.
- Úlohy odovzdať najneskôr do 12:00 v pondelok budúci týždeň.

## 10. prednáška

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

## 10. cvičenie

- Semestrálny test.