Funkcionální programování

Rekurzivní datové typy

Petr Pudlák

4. ledna 2012

Osnova

Datový typ $C\ a \approx ((C\ a) \to a)$

Povolení obecných rekurzivních datových typů vede samo o sobě k možnosti sestrojovat divergující/obecně rekurzivní funkce:

$$\textbf{data}\ Bad\ a = C\ (Bad\ a \to a)$$

$$xx :: Bad\ a \to a$$
$$xx\ (x@(C\ x')) = x'\ x$$

Každý typ obydlíme divergujícím termem $\Omega = (\lambda x.xx)(\lambda x.xx)$:
Poznámka: GHC 6.12.2 tuto funkci dokonce ani nezkompiluje, protože se (mylně) domnívá, že neobsahuje rekurzi, a tak se do nekonečna snaží funkci optimalizovat.

$$\Omega :: a$$
$$\Omega = xx\ (C\ xx)$$

Sestrojíme $\mathbb{Y}$ kombinátor, který v netypovaném lambda kalkulu odpovídá $\mathbb{Y} = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$:

$$\mathbb{Y} :: (a \to a) \to a$$
$$\mathbb{Y}\ f = \textbf{let}\ fxx = f \circ xx$$
$$\qquad\qquad \textbf{in}\ fxx\ (C\ fxx)$$

Osnova

Striktní a totální funkce

Cvičení

Jaký je rozdíl mezi striktní a totální funkcí (tedy funkcí definovanou pro všechny argumenty)?

Striktní a totální funkce

Cvičení

Jaký je rozdíl mezi striktní a totální funkcí (tedy funkcí definovanou pro všechny argumenty)?

Pro striktní funkci platí, že

$$x = \bot \Rightarrow f(x) = \bot$$

Pro totální funkci platí, že

$$f(x) = \bot \Rightarrow x = \bot$$

Kvaziuspořádání, částečně uspořádaná množina

Definition (Kvaziuspořádání (preorder))

Kvaziuspořádání je binární relace $\leq$, která je reflexivní a tranzitivní. Tedy:

$$x \leq x$$

$$x \leq y \text{ a } y \leq z \text{ pak } x \leq z$$

Definition (Uspořádaná množina (partially ordered set / poset))

Uspořádaná množina je množina $\mathbf{L}$, na které je dáno antisymetrické kvaziuspořádání $\leq$, tedy:

$$x \leq y \text{ a } y \leq x \text{ pak } x = y$$

Svaz

Definice pomocí uspořádané množiny

Definition

Svaz je uspořádaná množina $(\mathbf{L}, \leq)$, na které jsou definovány binární operace $\vee$ a $\wedge$, pro které platí:

- $x \vee y$ je supremum množiny $\{x, y\}$ vzhledem k $\leq$.
- $x \wedge y$ je infimum množiny $\{x, y\}$ vzhledem k $\leq$.

Cvičení

Dokažte:

1. *Operace $\vee$ a $\wedge$ jsou komutativní, asociativní a idempotentní.*
2. *$x \leq y$ právě když $x \wedge y = x$.*

Svaz

Algebraická definice

Definition

Svaz je množina **L**, na které jsou definovány binární operace $\vee$ a $\wedge$, pro které platí:

Komutativita:	$x \vee y = y \vee x$	$x \wedge y = y \wedge x$
Asociativita:	$x \vee (y \vee z) = (x \vee y) \vee z$	$x \wedge (y \wedge z) = (x \wedge y) \wedge z$
Absorpce:	$x \vee (x \wedge y) = x$	$x \wedge (x \vee y) = x$

Na **L** definujme uspořádání $\leq$ tak, že $x \leq y$ právě když $x \wedge y = x$.

Cvičení

1. *Dokažte, že takto definované operace $\vee$ a $\wedge$ jsou idempotentní.*
2. *Dokažte, že obě definice svazu jsou ekvivalentní.*

Úplný svaz

Definition

Úplný svaz je takový svaz, ve kterém existuje supremum a infimum každé (i prázdné) množiny. Značení:

$$\text{supremum množiny } W = \bigvee \mathbf{W}$$

$$\text{infimum množiny } W = \bigwedge \mathbf{W}$$

Cvičení

Co můžeme říci o:

1. $\bigvee\{\}$
2. $\bigwedge\{\}$

Úplný svaz

Definition

Úplný svaz je takový svaz, ve kterém existuje supremum a infimum každé (i prázdné) množiny. Značení:

$$\text{supremum množiny } W = \bigvee \mathbf{W}$$

$$\text{infimum množiny } W = \bigwedge \mathbf{W}$$

Cvičení

Co můžeme říci o:

1. $\bigvee\{\}$
2. $\bigwedge\{\}$

Poznámka

Často se značí $0 = \bigvee\{\}$ *a* $1 = \bigwedge\{\}$.

Knaster-Tarského věta

Theorem (Knaster-Tarski)

Buď $\mathbf{L}$ *úplný svaz a buď* $\mathfrak{F} : \mathbf{L} \to \mathbf{L}$ *funkce zachovávající uspořádání. Pak:*

1. *Množina* $\mathbf{K} = \{x \in \mathbf{L} \mid \mathfrak{F}(x) = x\}$ *pevných bodů funkce* $\mathfrak{F}$ *je úplný podsvaz* $\mathbf{L}$. *Zejména existují*
   - *její nejmenší prvek* $0_K = \bigvee_K \{\}$, *který budeme značit také* $\mu\mathfrak{F}$, *a*
   - *její největší prvek* $1_K = \bigwedge_K \{\}$, *který budeme značit také* $\nu\mathfrak{F}$.
2. *Jestliže* $x \geq \mathfrak{F}(x)$ *pak* $0_K \leq x$.
3. *Jestliže* $x \leq \mathfrak{F}(x)$ *pak* $1_K \geq x$.

Knaster-Tarského věta I

Důkaz

1. Dokážeme nejprve, že existuje nejmenší pevný bod $\mathfrak{F}$ (tedy 0_K). Existence největšího pevného bodu (1_K) se dokáže duálně.
2. Buď $\mathbf{D} = \{x \in \mathbf{L} \mid x \geq \mathfrak{F}(x)\}$.
   $\mathbf{D}$ je neprázdná, neboť $1_L \in \mathbf{D}$.
   $\mathbf{D}$ obsahuje všechny pevné body $\mathfrak{F}$.
3. Jelikož $\mathfrak{F}$ je monotónní, platí, že je-li $x \in \mathbf{D}$, pak $\mathfrak{F}(x) \in \mathbf{D}$.
4. Buď $u = \bigwedge_L \mathbf{D}$ a buď $x \in \mathbf{D}$ libovolné. Pak $x \geq u$, tedy $x \geq \mathfrak{F}(x) \geq \mathfrak{F}(u)$, tedy $\mathfrak{F}(u)$ je dolní závora $\mathbf{D}$. Jelikož u je největší dolní závora $\mathbf{D}$ (infimum), musí být $u \geq \mathfrak{F}(u)$ a tedy $u \in \mathbf{D}$. Pak musí i $\mathfrak{F}(u) \in \mathbf{D}$, tudíž $\mathfrak{F}(u) = u$ a u je proto nejmenším pevným bodem $\mathfrak{F}$.

Knaster-Tarského věta II

Důkaz

5. Zbývá dokázat, že $\mathbf{K}$ je to úplný svaz.
   Buď $\mathbf{W} \subseteq \mathbf{K}$ a buď $w = \bigwedge_L \mathbf{W}$.
   Je-li $z \in \mathbf{K}$ a nějaké $x \leq z$ pak $\mathfrak{F}(x) \leq \mathfrak{F}(z) = z$. Proto je-li nějaké $x \leq w$ pak i $\mathfrak{F}(x) \leq w$. Funkce $\mathfrak{F}$ je tedy uzavřená na podsvazu $\mathbf{J} = \{x \in \mathbf{L} \mid x \leq w\}$.
   Podle předchozí části existuje t největší pevný bod $\mathfrak{F}$ na svazu $\mathbf{J}$.
   Zřejmě $t = \bigwedge_K \mathbf{W}$.

Knaster-Tarského věta pro svaz podmnožin

Aplikací předchozí věty na svaz podmnožin dané množiny dostáváme tyto věty:

Theorem

Buď $\mathbf{U}$ *množina a buď* $\mathfrak{F} : \mathcal{P}(\mathbf{U}) \to \mathcal{P}(\mathbf{U})$ *taková, že* $A \subseteq B$ *implikuje* $\mathfrak{F}(A) \subseteq \mathfrak{F}(B)$.
Pak existují nejmenší $\mu\mathfrak{F} \subseteq \mathbf{U}$ *a největší* $\nu\mathfrak{F} \subseteq \mathbf{U}$ *pevné body* $\mathfrak{F}$.

Theorem (Princip indukce)

Buď $\mathbf{P} \subseteq \mathbf{U}$ *množina těch prvků z* $\mathbf{U}$, *které mají nějakou vlastnost* P.
Jestliže $\mathfrak{F}(\mathbf{P}) \subseteq \mathbf{P}$ *pak* $\mu\mathfrak{F} \subseteq \mathbf{P}$, *čili všechny prvky z* $\mu\mathfrak{F}$ *mají vlastnost* P.

Theorem (Princip koindukce)

Buď $\mathbf{P} \subseteq \mathbf{U}$ *množina těch prvků z* $\mathbf{U}$, *které mají nějakou vlastnost* P.
Jestliže $\mathbf{P} \subseteq \mathfrak{F}(\mathbf{P})$ *pak* $\mathbf{P} \subseteq \nu\mathfrak{F}$, *čili všechny prvky s vlastností* P *jsou v* $\nu\mathfrak{F}$.

Knaster-Tarského věta pro svaz podmnožin

Monotónní množinové funkce

Je-li $\mathfrak{F}$ definována po prvcích, tedy

$$\mathfrak{F}(\mathbf{A}) = \bigcup_{x \in \mathbf{A}} \mathfrak{F}'(a)$$

kde $\mathfrak{F}' : \mathbf{U} \to \mathcal{P}(\mathbf{U})$, pak $\mathfrak{F}$ je zřejmě monotónní vzhledem k $\subseteq$.
Pišme v takovém případě zkráceně $\mathfrak{F}(x)$ místo $\mathfrak{F}(\{x\})$.

Osnova

Svaz a monotónní funkce na rekurzivních typech

- Mějme typ $\tau : * \to *$ složený z ADT.
- Buď α proměnná parametrizující τ.
- Pro jednoduchost předpokládejme, že τ neobsahuje $\to$.
- Buď C (rekurzivní) typ na nejvyšší úrovni.
- Buď $\mathbf{K}$ množina jeho konstruktorů $\mathbf{K} = \{C_1, \ldots, C_n\}$.
- Buď $\mathbf{U}$ množina všech hodnot tohoto typu, které můžeme zkonstruovat. $\mathbf{U}$ bude obsahovat všechny stromy, konečné i nekonečné, jejichž uzly jsou konstruktory z $\mathbf{K}$, a nebo $\perp$, a každý uzel má „pověšený" podstromy v místech, kde je v konstruktoru α.
- Buď $\mathfrak{F}$ funkce z $\mathcal{P}(\mathbf{U})$ do $\mathcal{P}(\mathbf{U})$, kde $\mathfrak{F}(\mathbf{A})$ obsahuje všechny hodnoty, které získáme dosazením prvků z $\mathbf{A}$ za α do konstruktorů z $\mathbf{K}$, a všechny možné hodnoty v ostatních argumentech.
  Jelikož $\mathfrak{F}$ je definována po prvcích, je monotónní vzhledem k $\subseteq$.

μ a ν rekurzivních typech

- Podle Knaster-Tarského věty pak existují nejmenší a největší pevné body, tedy množiny uzavřené na aplikaci dalšího konstruktoru.
- Platí:
  1. Nejmenší pevný bod $\mu\mathfrak{F}$ je množina obsahující právě konečné stromy, které neobsahují $\perp$.
     Datový typ odpovídající této množině značíme $\mu\tau$ a nazýváme *induktivní (někdy rekurzivní) datový typ*.
  2. Největší pevný bod $\nu\mathfrak{F}$ je množina obsahující právě ty konečné i nekonečné stromy, které neobsahují $\perp$.
     Datový typ odpovídající této množině značíme $\nu\tau$ a nazýváme *koinduktivní (někdy korekurzivní) datový typ*.
- Obojí dokážeme indukcí podle hloubky stromu z toho, že množiny jsou pevné body $\mathfrak{F}$.

Označení definičního oboru a oboru hodnot funkce

Nahlížejme nadále na typy jako na množiny možných hodnot.

Definition

Pro funkci $g : \alpha \to \beta$ označme

- $\mathrm{dom}\, g = \{x \in \mathbf{U} \mid gx \in \beta\}$
- $\mathrm{rng}\, g = \{y \in \mathbf{U} \mid$ ex. $x \in \alpha$ tž. $gx = y\}$.

Definition

Řekneme, že funkce $g : \alpha \to \beta$ je totální, pokud $\alpha \subseteq \mathrm{dom}\, g$ nebo ekvivalentně $\mathrm{rng}\, g \subseteq \beta$.

Důkazy vlastností funkcí indukcí I

Theorem

Buď $g :: \mu\,\tau \to c$ rekurzivní funkce. Zapišme g jako

$$g = fix\ g'$$

kde $g' :: (\mu\,\tau \to c) \to (\mu\,\tau \to c)$ a $fix :: (a \to a) \to a$ je kombinátor rekurze Y.
Nechť platí, že:

- *Při každém rekurzivním volání g se dosazuje podkomponenta původního parametru.*
- *Funkce g' je totální (vzhledem k druhém parametru a k výsledkům rekurzivního volání).*

Pak g je totální.

Důkazy vlastností funkcí indukcí II

Důkaz:

- Jelikož g' je totální a odebírá z argumentu konstruktor, než ho předá do rekurzivního volání, platí, že $\mathfrak{F}(\operatorname{dom} h) \subseteq \operatorname{dom}(g'\,h)$.
- A protože $g = g'\ g$, platí, že $\mathfrak{F}(\operatorname{dom} g) \subseteq \operatorname{dom} g$.
- Pak podle Knaster-Tarského věty je $\mu\mathfrak{F} \subseteq \operatorname{dom} g$.

Zobecnění induktivních datových typů

- Rekurze se v definici induktivního datové typů nesmí vyskytovat vlevo od $\rightarrow$.
  (Jinak lze nekontrolovaně konstruovat $\perp$ či obecnou rekurzi.)
- Funkce na více argumentech musí do rekurzivního volání rozebrat alespoň jeden z konstruktorů.

Katamorfismy I

- Funkce, které nezkoumají rekurzivní argumenty konstruktorů, a pouze je předávají do rekurzivního volání, nazvěme *katamorfismy*.
- Katamorfismy jsou zobecnění *foldr* funkce na libovolné induktivně rekurzivní typy.
- Pro typ $\mu\tau$ lze jeho katamorfismy popsat funkcí $\tau c \to c$ a z této funkce zkonstruovat katamorfismus pomocí obecné funkce $\text{fold}_\tau : (\tau c \to c) \to \mu\tau \to c$.
- (Příklady.)

Katamorfismy II

Cvičení

Zapište obecnou funkci vyššího řádu pro popis katamorfismů (obdobu foldr) na datovém typu

$$\textbf{data}\ \mathit{Nat} = \mathit{Zero} \mid \mathit{Succ}\ \mathit{Nat}$$

Cvičení

Zapište funkce $+$ a $*$ jako katamorfismy pomocí této funkce.

Paramorfismy I

- Paramorfismy jsou zobecnění katamorfismů.
  "[Paramorhism] eats its argument and keeps it too."
- Pro typ $\mu\tau$ lze jeho paramorfismy popsat typem $\tau c \to \mu\tau \to c$ a z této funkce zkonstruovat paramorfismus pomocí obecné funkce $\text{fold'}_\tau : (\tau c \to \mu\tau \to c) \to \mu\tau \to c$.
- (Příklady – např. faktoriál.)
- Každý paramorfismus lze převádět na katamorfismus.

Paramorfismy II

Cvičení

Zapište obecnou funkci vyššího řádu pro popis katamorfismů (obdobu foldr) na datovém typu

$$\textbf{data}\ \mathit{Nat} = \mathit{Zero} \mid \mathit{Succ}\ \mathit{Nat}$$

Cvičení

Zapište funkci faktoriál jako paramorfismus pomocí této funkce.

Cvičení

Zapište funkci faktoriál jako katamorfismus.

Osnova

Důkazy vlastností funkcí koindukcí I

Theorem

Buď $g :: c \to \nu\ \tau$ rekurzivní funkce. Zapišme g jako

$$g = fix\ g'$$

Nechť platí, že:

- *Každé rekurzivním volání g je argumentem konstruktoru z $\mathbf{K}$.*
- *Funkce g' je totální.*

Pak pro obor hodnot g je obsažen v $\nu\ \tau$.

Důkazy vlastností funkcí koindukcí II

Důkaz:

- Jelikož g' je totální a „obaluje" výsledek z rekurzivního volání konstruktorem, platí, že $\mathrm{rng}(g'\,h) \subseteq \mathfrak{F}(\mathrm{rng}\,h)$.
- A protože $g = g'\,g$, platí, že $\mathrm{rng}\,g \subseteq \mathfrak{F}(\mathrm{rng}\,g)$.
- Pak podle Knaster-Tarského věty je $\mathrm{rng}\,g \subseteq \nu\mathfrak{F}$.

Anamorfismy

- *Anamorfismy* jsou duální ke katamorfismům.
- Pro typ $\nu\tau$ lze jeho anamorfismy popsat typem $c \to \tau c$.
- (Příklady – *zip*, *iterate*.)

Apomorfismy

- *Apomorfismy* jsou duální k paramorfismům.

Literatura I

[1] Benjamin C. Pierce. *Types and Programming Languages*. MIT Press, 2002. URL: http://www.cis.upenn.edu/~bcpierce/tapl.

[2] D. A. Turner. ?Total Functional Programming? In: *Journal of Universal Computer Science* 10.7 (2004), pp. 751–768.

[3] Varmo Vene. ?Categorical Programming With Inductive and Coinductive Types? PhD thesis. University of Tartu, 2000. URL: http://www.cs.ut.ee/~varmo/papers/thesis.pdf.

[4] Philip Wadler. ?Recursive types for free!? University of Glasgow; (Bohuzel jsem na tento text narazil az pote, co jsme to probirali.) 1990. URL: http://homepages.inf.ed.ac.uk/wadler/papers/free-rectypes/free-rectypes.txt.