Formální Metody a Specifikace (LS 2011)
Přednáška 4:
Logické teorie a modelování datových struktur

Stefan Ratschan

Katedra číslicového návrhu
Fakulta informačních technologií
České vysoké učení technické v Praze

11. březen 2011

PRA HA
PRA GUE
PRA GA
PRA G

EVROPSKÁ UNIE

Datové typy

Následující algoritmus:

Input: $x_1, \dots, x_n$
return $f(x_1, \dots, x_n)$

přičemž f je term jako např.: x^2, $a[2i]$, $\mathrm{rest}(x)$ atd.

Předpokládáme specifikaci:

- $I(x)$: x je dovolený vstup
- $O(x, x')$: x' je dovolený výstup pro vstup x

Správnost algoritmu:

$$\forall x \forall x' \, . \, [I(x) \wedge x' = f(x)] \Rightarrow O(x, x')$$

Důkaz správnosti

Stačí zjistit jestli

$$\models \forall x \forall x' \,.\, [I(x) \wedge x' = f(x)] \Rightarrow O(x, x') \ \ ?$$

... a kvůli úplnosti predikátové logiky prvního řádu (viz. Gödel)
vždy můžeme najít příslušný důkaz?

Navíc: Pro libovolní program stačí jen používat příslušnou funkci f.

Tudíž: Můžeme dokázat správnost libovolných programů, a
Temelín nikdy nebude explodovat kvůli chybějící čárce v programu?

A teď nás vyházejí z ráje ...

Chování libovolných programů lze formálně popsat,
takovou funkcí f (zbytek přednášky), ale ...

Datové struktury

Programy i specifikace obsahují termy jako $a[2i]$, $\mathrm{rest}(x)$

$$\forall x \forall x' \,.\, [I(x) \wedge x' = f(x)] \Rightarrow O(x, x') \ \ ?$$

Zkusíme všechny možnosti?

Potřebujeme konečný a objektivní doklad správnosti (tj. důkaz!)

Musíme formálně popsat chování takových datových struktur, např.

- integer (pozor: množina integerů není množina celých čísel)
- float (pozor: množina floatů není množina reálných čísel!)
- pole
- seznamy

Příklad: Páry

Typy: $\mathcal{P}$, prvky: Typ $\mathcal{T}$

Funkční symboly:

- pair: $\mathcal{T} \times \mathcal{T} \to \mathcal{P}$
- fst: $\mathcal{P} \to \mathcal{T}$
- snd: $\mathcal{P} \to \mathcal{T}$

Specifikace/algoritmus:

- $I(p) :\Leftrightarrow \exists x, y \,.\, p = pair(x, y)$
- $O(p, p') :\Leftrightarrow \exists x, y \,.\, p = pair(x, y) \Rightarrow p' = pair(y, x)$
- $f(p) \ = pair(snd(p), fst(p))$

Zkusíme dokázat $\models \forall p \forall p' \,.\, [I(p) \land p' = f(p)] \Rightarrow O(p, p')$

Chybí chování párů:

$$\forall x, y \,.\, \mathrm{fst}(\mathrm{pair}(x, y)) = x$$

$$\forall x, y \,.\, \mathrm{snd}(\mathrm{pair}(x, y)) = y$$

Modelování datových struktur

$$\models T \Rightarrow \left[\forall x \forall x' \,.\, [I(x) \wedge x' = f(x)] \Rightarrow O(x, x')\right]$$

přičemž T popisuje chování datových struktur. V našem případě:

$$T :\Leftrightarrow [\forall x, y \,.\, \mathrm{fst}(\mathrm{pair}(x, y)) = x] \wedge [\forall x, y \,.\, \mathrm{snd}(\mathrm{pair}(x, y)) = y]$$

Takový popis se jmenuje *axiomatizace*.

Jak můžeme axiomatizovat jiné datové struktury?

Axiomatizace podle algebry

Např: Podle algebry jsou celá čísla grupou.

Můžeme používat axiomy teorie grup:

- Asociativita: $\forall x, y, z \,.\, x + (y + z) = (x + y) + z$
- 0 je neutrálním prvkem: $\forall x \,.\, 0 + x = x \wedge x + 0 = x$
- Existence inverzních prvků: $\forall x \exists v \,.\, x + v = 0 \wedge v + x = 0$

Problémy:

- Příliš slabý (takto sice lze dokázat $x + 1 - 1 = x$ ale $x + 2 - 1 - 1 = x$ ne)
- Běžné datové struktury často nesplňují axiomy klasické algebry

Ale: Parametrické datové typy (např: seznam s prvky z grupy).

Axiomatizace seznamů

Typy: $\mathcal{L}$, prvky: $\mathcal{T}$

Funkční symboly:

- $\mathrm{cons} : \mathcal{T} \times \mathcal{L} \to \mathcal{L}$
- $\mathrm{first} : \mathcal{L} \to \mathcal{T}$
- $\mathrm{rest} : \mathcal{L} \to \mathcal{L}$
- $\mathrm{empty} :\to \mathcal{L}$

Axiomy ($l \in \mathcal{L}$, $x \in \mathcal{T}$):

- $\forall l, x \,.\, \mathrm{first}(\mathrm{cons}(x, l)) = x$
- $\forall l, x \,.\, \mathrm{rest}(\mathrm{cons}(x, l)) = l$
- $\forall l, x \,.\, l \neq \mathrm{empty}() \Rightarrow \mathrm{cons}(\mathrm{first}(l), \mathrm{rest}(l)) = l$
- $\forall l, x \,.\, \mathrm{cons}(x, l) \neq \mathrm{empty}()$

Axiomatizace seznamů: poznámky

- $\forall l, x \,.\, \mathrm{first}(\mathrm{cons}(x, l)) = x$
- $\forall l, x \,.\, \mathrm{rest}(\mathrm{cons}(x, l)) = l$
- $\forall l, x \,.\, l \neq \mathrm{empty}() \Rightarrow \mathrm{cons}(\mathrm{first}(l), \mathrm{rest}(l)) = l$
- $\forall l, x \,.\, \mathrm{cons}(x, l) \neq \mathrm{empty}()$

Chování first(empty()), rest(empty()) není specifikován

Mohli bychom např. zavést konstantu error: $\mathrm{first}(\mathrm{empty}()) = \mathrm{error}$ atd.

Nemáme funkci pro délku seznamu (ještě nemáme přirozená čísla)

Axiomatizace polí

Typy: $\mathcal{A}_n$, prvky: $\mathcal{T}$ (délku polí n volíme předem)

Funkční symboly:

- new $:\to \mathcal{A}_n$
- $\cdot[\cdot] : \mathcal{A}_n \times \{1, \ldots, n\} \to \mathcal{T}$
- write $: \mathcal{A}_n \times \{1, \ldots, n\} \times \mathcal{T} \to \mathcal{A}_n$

Axiomy ($a, b \in \mathcal{A}_n, v \in \mathcal{T}, i, j \in \{1, \ldots, n\}$):

- $\forall a, v, i, j \;.\; i = j \Rightarrow \mathrm{write}(a, i, v)[j] = v$
- $\forall a, v, i, j \;.\; i \neq j \Rightarrow \mathrm{write}(a, i, v)[j] = a[j]$
- $\forall a, b \;.\; [\bigwedge_{i \in \{1,\ldots,n\}} a[i] = b[i]] \Leftrightarrow a = b$

Axiomatizace přirozených čísel

Typ: $\mathcal{N}$

Funkční symboly:

- $0 :\rightarrow \mathcal{N}$
- $1 :\rightarrow \mathcal{N}$
- $+ : \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{N}$
- $\cdot : \mathcal{N} \times \mathcal{N} \rightarrow \mathcal{N}$

Axiomy (Peanova aritmetika)

- $\forall x \;.\; x + 1 \neq 0$
- $\forall x, y \;.\; x + 1 = y + 1 \Rightarrow x = y$
- $\forall x \;.\; x + 0 = x$
- $\forall x, y \;.\; x + (y + 1) = (x + y) + 1$
- $\forall x \;.\; x0 = 0$
- $\forall x, y \;\; x(y + 1) = xy + x$
- ...

Axiomatizace přirozených čísel

Pro každou formuli F s přesně jednou volnou proměnnou x, další axiom

$$\Big[F[x \leftarrow 0] \wedge \big[\forall x \,.\, F \Rightarrow F[x \leftarrow x+1]\big]\Big] \Rightarrow \forall x \,.\, F$$

Jinak řečeno: indukce

Pro každou formuli F? Nekonečný počet axiomů!

Bohužel žádná konečná axiomatizace neexistuje, a

bohužel to je jen začátek problémů,

a bohužel je za to vinen Brňák.

Neúplnost

První Gödelova věta o neúplnosti:
Nechť T je logická teorie dostatečně silná pro vyjádření aritmetiky přirozených čísel. Pak existuje formule v, která není v T dokazatelná ani vyvratitelná. [Gödel, 1931]

Máme štěstí v neštěstí: v praxi to není žádný problém.

Další teorie

Rozšíření teorie přirozených čísel:

$\leq$?

$x \leq y :\Leftrightarrow \exists k \,.\, x + k = y$

Celá čísla: $x - y$

Obrovský počet dalších teorií, např. teorie reálných čísel.

Teorie množin

Axiomatizace: Zermelo-Fraenkel set theory (ZFC) (C: axiom of choice)

Bohužel příliš složitá pro tuto přednášku (viz. internet)

Místo toho: pár základních pravidel pro každodenní použití:

Množiny vytvoříme takto: $\{x \mid A\}$, přičemž A je logická formule

Definice: Pro množiny S a T,

- $S \cap T \doteq \{x \mid x \in S \land x \in T\}$
- $S \cup T \doteq \{x \mid x \in S \lor x \in T\}$
- $S \setminus T \doteq \{x \mid x \in S \land x \notin T\}$
- Prázdná množina je množina $\emptyset$ tak, že $\neg \exists x \,.\, x \in \emptyset$

Equivalence: Pro množiny S a T,

- $S \subseteq T$ je ekvivalentně $\forall x \,.\, x \in S \Rightarrow x \in T$
- $S = T$ je ekvivalentně $\forall x \,.\, x \in S \Leftrightarrow x \in T$
- $a \in \{x \mid A\}$ je ekvivalentně $A[x \leftarrow a]$

Množiny

Navíc platí:

- Pro množinu S, $\{x \mid x \in S\} = S$

Často se používá notace:

- $\{a\}$ pro $\{x \mid x = a\}$
- $\{a_1, \dots, a_n\}$ pro $\{x \mid x = a_1 \vee \cdots \vee x = a_n\}$
- $\forall x \in S \,.\, A$ pro $\forall x \,.\, x \in S \Rightarrow A$
- $\exists x \in S \,.\, A$ pro $\exists x \,.\, x \in S \wedge A$

Diskuse

V teorii množin můžeme formalizovat všechno co jsme viděli až dosud!

Zejména: V teorii množin můžeme formalizovat přirozená čísla

Důsledek: zase ten Gödel!

Zase máme štěstí v neštěstí:
V praxi predikátová logika prvního řádu + teorie množin stačí pro vybudování celé matematiky (všeho co lze formalizovat).

Aspoň až dosud ....

Pokud teorie množin je tak mohutná, že umí všechno modelovat ...
Proč jsem vůbec diskutovali jiné teorie?

Decision procedures

Určité teorie T jsou *rozhodnutelné*, existuje algoritmus se specifikací

- Vstup: formule ϕ v logické teorie T
- Výstup: **T** pokud $T \models \phi$, jinak **F**.

$T \models \phi$ vs. $\models T \Rightarrow \phi$ (viz. "sémantický důsledek")

Ve většině naších teorií se jedná o rozhodnutelné teorie (demo)

V takových případech
můžeme dokázat správnost algoritmu ze začátku automaticky.

$$\forall x \forall x' \,.\, [I(x) \wedge x' = f(x)] \Rightarrow O(x, x')$$

Ale: Peanova aritmetika, teorie množin nejsou rozhodnutelné [Church, 1936, Turing, 1937]

Diskuse

Proč vlastně potřebujeme

- celá/přirozená čísla (počítačové integery jsou jen konečnou podmnožinou),
- reálná čísla (v počítači máme čísla s pohyblivou čárkou), a
- a dynamické datové struktury (počítač má jen konečnou paměť)?

V určitém počítači máme jen konečný počet bitů,
tj. algoritmus má jen konečné chování,
museli bychom jen to všechno zkusit!

Ve verifikaci hardwaru se to částečně dělá,
ale v případě software s tím obvykle nemáme šanci.

Viz. přednáška MI-MAS: Modelování a analýza systémů

Combination procedures

Programy obvykle nepoužívají
jen integery, anebo jen seznamy, jen pole atd.

Obvykle máme směs různých datových struktur.

Combination procedures:
Algoritmy které v určitých případech
umí rozhodnout kombinace různých teorií

Uvidíme (možná na konci semestru).

Předpověď přednášek

$$\models T \Rightarrow \left[\forall x \forall x' \,.\, [I(x) \land x' = f(x)] \Rightarrow O(x, x')\right]$$

Literature I

Alonzo Church. An unsolvable problem of elementary number theory. *Am. J. Math.*, 58:345–363, 1936.

Kurt Gödel. Über formal unentscheidbare Sätze der Pxrincipia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik*, 38:173–198, 1931.

A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.