Formální Metody a Specifikace (LS 2011)
Přednáška 3:
Specifikace programů a základy praktické logiky 2

Stefan Ratschan

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

4. březen 2011

Motivace

Pro ověřování správnosti algoritmu,
musíme nejdřív specifikovat žádoucí chování

Zatím: Specifikace vstupů/výstupů

Precizní, jednoznačný, univerzální jazyk: predikátová logika prvního řádu

Minulá přednáška:

- Syntax: správný tvar logických formulí
- Sémantika: formule platí ($\models$)
- Pravidla ekvivalence

Ověřování správnosti triviálního algoritmů

Dnes: Následující algoritmus:

Input: x

return x

Příklad specifikace:

- Vstup: $x \in \mathbb{N}$, x dělitelné 4
- Výstup: $x \in \mathbb{N}$, x dělitelné 2

Dělitelnost v predikátové logice: x dělitelné y: $\exists k \,.\, yk = x$

$$\forall x \,.\, [\exists k \,.\, 4k = x] \Rightarrow [\exists k \,.\, 2k = x]$$

Interpretace $\mathcal{N}$: dává všem symbolům obvyklý význam v teorii $\mathbb{N}$

Chceme vědět: $\mathcal{N} \models \forall x \,.\, [\exists k \,.\, 4k = x] \Rightarrow [\exists k \,.\, 2k = x]$

Problém: Museli bychom ověřit nekonečný počet čísel.

Důkazy

Důkaz: Doklad že určitá formule platí, který je:

- Konečný
- Objektivní: Kdokoli může ověřit správnost důkazu a, tím pádem, platnost formule.

Pokud se důkaz píše formálně,
dokonce počítač může ověřit jeho správnost (viz. proof-carrying code)

Jak ověřit že něco platí?

Už víme, že můžeme dokázat $A \Leftrightarrow B$ tím,
že dokážeme že A je ekvivalentně B.

Ale bohužel to nestačí (metoda není úplná).

Dnes: Metoda důkazů která je úplná:
Pro každou formuli ϕ tak, že $\models \phi$, existuje důkaz.

Neformální verze přirozené dedukce (natural deduction)

Alternativy (vhodný pro studium základy matematiky, ale ne pro každodenní dokazování):

- Hilbertův kalkulus (Hilbert calculus)
- Sekvenční kalkulus (sequent calculus)

Metoda důkazů

Když chceme něco dokázat,
používáme seznam skutečností které už známe, a

- přidáme další skutečnosti ("usoudíme", "předpokládáme")
- a zjednodušujeme to, co chceme dokázat, až do okamžiku kdy

to, co chceme dokázat je známo.

Přitom: Vždy můžeme něco nahradit ekvivalentním

Důkazní pravidla—výrokový případ bez negací

Jak zjednodušovat to, co chceme dokázat?

- $A \wedge B$: Nejdřív dokážeme A, pak B
- $A \vee B$: Předpokládáme $\neg A$ a dokážeme B
- $A \Rightarrow B$: Předpokládáme A a dokážeme B.

Jak produkovat nové známé věci?

- $A \wedge B$: Usoudíme i A i B
- $A \vee B$: Píšeme nejdřív "Případ A:", předpokládáme A a dokončíme důkaz. Pak píšeme "Případ B:", předpokládáme B, a dokončíme důkaz
- $A \Rightarrow B$: Pokud známe i A pak usoudíme B

Příklad

Příklad:

$$[p \Rightarrow q] \Rightarrow \big[\, [p \wedge r] \Rightarrow [q \wedge r] \,\big]$$

Důkaz je celý proces který jsem ukázal na tabuli.

Tento proces můžeme i popsat písemně na papíře:

Chceme dokázat $[p \Rightarrow q] \Rightarrow \big[\, [p \wedge r] \Rightarrow [q \wedge r] \,\big]$. Předpokládáme $p \Rightarrow q$ a dokážeme $[p \wedge r] \Rightarrow [q \wedge r]$. Pro důkaz formule $[p \wedge r] \Rightarrow [q \wedge r]$ předpokládáme $p \wedge r$ a dokážeme $q \wedge r$. Pro důkaz formule $q \wedge r$ dokážeme i q i r:

- Důkaz q: Máme předpoklad $p \wedge r$. Z toho plyne i předpoklad p i r. Z předpokladů p a $p \Rightarrow q$ plyne q, což skončí tuto část důkazu.
- Důkaz r: Už jsme zjistili že z předpokladu $p \wedge r$ plyne r, což skončí i tuto část důkazu.

Důkazní pravidla—výrokový případ

Jak zjednodušovat to, co chceme dokázat?

- $A \wedge B$: Nejdřív dokážeme A, pak B
- $A \vee B$: Předpokládáme $\neg A$ a dokážeme B
- $A \Rightarrow B$: Předpokládáme A a dokážeme B.
- $\neg A$: Předpokládáme A a zkusíme najít spor

Spor: Situace kdy pro určité P, i P i $\neg P$ jsou známé skutečnosti.

Jak produkovat nové známé věci?

- $A \wedge B$: Usoudíme i A i B
- $A \vee B$: Píšeme nejdřív "Případ A :", předpokládáme A a dokončíme důkaz. Pak píšeme "Případ B :", předpokládáme B, a dokončíme důkaz
- $A \Rightarrow B$: Pokud známe i A pak usoudíme B
- $\neg A$: Pokud zkusíme najít spor, pak místo toho dokážeme A .

Příklad

$$\big[\,[p \Rightarrow q] \wedge [p \Rightarrow \neg q]\,\big] \Rightarrow \neg p$$

Důkaz v písemně formě:

Předpokládáme $[p \Rightarrow q] \wedge [p \Rightarrow \neg q]$ a dokážeme $\neg p$. Pro důkazu $\neg p$ předpokládáme p a zkusíme najít spor.

Z předpokladu $[p \Rightarrow q] \wedge [p \Rightarrow \neg q]$ známe i předpoklad $p \Rightarrow q$ i předpoklad $p \Rightarrow \neg q$. Z předpokladů p a $p \Rightarrow q$ plyne q, a z předpokladů p a $p \Rightarrow \neg q$ plyne $\neg q$. Tudíž máme i předpoklad q i $\neg q$ což je hledaný spor. Konec důkazu.

Důkazní pravidla včetně kvantifikátorů 1

Jak zjednodušovat to co chceme dokázat?

- $\exists x \,.\, A$: Vybereme (intuicí, kreativitou, anebo jinou speciální spojení s Bohem) term t, a dokážeme $A[x \leftarrow t]$
- $\forall x \,.\, A$: Vybereme novou konstantu a, píšeme "Nechť a je libovolné ale pevné" a dokážeme $A[x \leftarrow a]$
- $A \wedge B$: Nejdřív dokážeme A, pak B
- $A \vee B$: Předpokládáme $\neg A$ a dokážeme B
- $A \Rightarrow B$: Předpokládáme A a dokážeme B.
- $\neg A$: Předpokládáme A a zkusíme najít spor

Důkazní pravidla včetně kvantifikátorů 2

Jak produkovat nové známé věci?

- $\exists x \,.\, A$: Vybereme novou konstantu a, píšeme "nechť a je tak, že $A[x \leftarrow a]$", a přídáme $A[x \leftarrow a]$ do našeho seznamu známých skutečností
- $\forall x \,.\, A$: Vybereme (intuicí, kreativitou, anebo jinou specialní spojení s Bohem) term t a usoudíme $A[x \leftarrow t]$
- $A \wedge B$: Usoudíme i A i B
- $A \vee B$: Píšeme nejdřív "Případ A :", předpokládáme A a dokončíme důkaz. Pak píšeme "Případ B :", předpokládáme B, a dokončíme důkaz
- $A \Rightarrow B$: Pokud známe i A pak usoudíme B
- $\neg A$: Pokud zkusíme najít spor, pak místo toho dokážeme A .

Příklad

$$\Big[[\forall x \,.\, P(x) \Rightarrow Q(x)] \wedge [\exists x \,.\, P(x)]\Big] \Rightarrow [\exists x \,.\, Q(x)]$$

Předpokládáme $[\forall x \,.\, P(x) \Rightarrow Q(x)] \wedge [\exists x \,.\, P(x)]$, tj. předpokládáme i $\forall x \,.\, P(x) \Rightarrow Q(x)$ i $\exists x \,.\, P(x)$. Chceme dokázat $\exists x \,.\, Q(x)$.

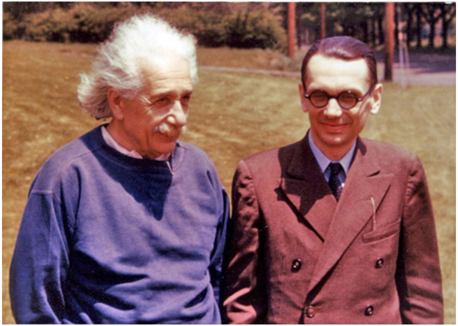
Kvůli předpokladu $\exists x \,.\, P(x)$ můžeme vybrat novou konstantu a tak, že $P(a)$. Kvůli předpokladu $\forall x \,.\, P(x) \Rightarrow Q(x)$ můžeme usoudit $P(a) \Rightarrow Q(a)$. Pro důkaz $\exists x \,.\, Q(x)$ stačí dokázat $Q(a)$ což plyne z předpokladů $P(a)$ a $P(a) \Rightarrow Q(a)$. Konec důkazu.

Důkazní pravidla pro rovnosti

Pro každý term t smíme předpokládat $t = t$.

Pokud víme $t_1 = t_2$,
pak můžeme vždy nahrazovat t_1 termem t_2 a obráceně.

Úplnost



Kurt Gödel, 1906 (Brno) — 1978 (Princeton): Disertace, 33 stránek:

Pro každou formuli ϕ, tak že $\models \phi$, existuje odpovídající důkaz.

Tentokrát: ... v jiném důkaznímu systému [Gödel, 1929].

Ale platí i pro náš systém.

Definice

Bylo by velmi obtížně kdybychom
pokaždé, když chceme vyjádřit že x je sudé,
psali formuli $\exists y \ . \ 2y = x$.

Podobně jako funkce/procedury v programování,
v logice můžeme definovat nové symboly

- abychom nemuseli znovu psát každý výskyt stejné formule
- abychom strukturovali a zpřehledňovali formule
- abychom se vyhýbali redundance

Můžeme definovat predikáty a funkční symboly.

Definice predikátů

$$\forall v_1 \dots v_n \, . \, p(v_1, \dots, v_n) :\Leftrightarrow \phi$$

přičemž

- p je (nový) predikát s aritou n,
- ϕ je formule která
  - neobsahuje p, a
  - neobsahuje jiné volné proměnné než $v_1, \dots, v_n$.

Příklady špatných definicí.

Pak $p(t_1, \dots, t_n)$ stojí pro $\phi[v_1 \leftarrow t_1, \dots, v_n \leftarrow t_n]$
a můžeme nahradit jednu stranu druhou.

Explicitní definice funkčních symbolů

$$\forall v_1 \dots v_n \,.\, f(v_1, \dots, v_n) := t$$

přičemž

- f je (nový) funkční symbol s aritou n,
- t je term
  - neobsahuje f, a
  - neobsahuje jiné proměnné než $v_1, \dots, v_n$.

Pak $f(t_1, \dots, t_n)$ stojí pro $t[v_1 \leftarrow t_1, \dots, v_n \leftarrow t_n]$
a můžeme nahradit jednu stranu druhou.

Implicitní definice funkčních symbolů

Příklad:

$$\forall x_1, x_2, y \;.\; y = NSD(x_1, x_2) :\Leftrightarrow$$
$$y \text{ dělí } x_1, y \text{ dělí } x_2, \neg\exists y' \;.\; y' > y, y' \text{ dělí } x_1, y' \text{ dělí } x_2$$

Obecně:

$$\forall v_1 \dots v_n, v \;.\; v = f(v_1, \dots, v_n) :\Leftrightarrow \phi$$

- f je (nový) funkční symbol s aritou n,
- ϕ je formule která
  - neobsahuje f, a
  - neobsahuje jiné volné proměnné než $v_1, \dots, v_n$.

Nahrazení je trochu složitější (nejdřív zavedení nového existenčního kvantifikátoru, pak nahrazení)

Definice v praxi

V matematických a informatických textech,
definice se obvykle používají neformálním způsobem ...

... což je v pořádku:
Naše formální diskuse nám pomáhá
správně číst a psát neformální definice.

Alternativní symboly označující definice: $\dot{\Leftrightarrow}, \doteq, \hat{=}, \ldots$

Chybí: rekursivní definice (používají se často pro seznamy, celá čísla atd.)

Předpověď přednášky

- Množiny, seznamy, pole, atd.
- Správnost programů
- Správnost programů
- Správnost programů
- Správnost programů
- ...

Literature I

Kurt Gödel. *Über die Vollständigkeit des Logikkalküls*. PhD thesis, Universität Wien, 1929.