Uvod u formalno dokazivanje teorema

Danijela Petrović

- Isabelle/HOL
- HOL – Higher Order-Logic
- Veza između funkcionalnog programiranja i logike
- Isar

- Rad u Isabelle predstavlja kreiranje teorija.
- **Teorija** predstavlja skup tipova, funkcija, teorema, ...

Izgled programa u Isabelle

- theory T
  imports B_1, ..., B_n
  begin
  declarations, definitions and proofs
  end

- Svaka teorija T mora da se nalazi u fajlu koji se naziva T.thy

Tipovi

- Osnovni tipovi: bool, nat, ...
- Složeni tipovi: list, set, ...
- Funkcionalni tipovi, označeni sa ⇒
- Tipovi vrijabli (promenljivih): `a, `b, …
- Korisnički definisani tipovi...

Definisanje korisničkih tipova

- Ključne reč: typedecl (nerekurzivni tipovi) i datatype (rekurzivni tipovi)
- Primer:

  typedecl addr – apstraktni tip za adrese

- Primer:

  datatype natlist = Prazno
  | Nadovezi nat natlist

- Nadovezi 3 (Nadovezi 4 (Nadovezi 5 Prazno))

Primer 2

- Skup iskaznih formula (ili jezik iskazne logike) nad skupom P je najmanji podskup skupa svih reci nad takav da vazi:
- iskazna slova (iz skupa P) i logicke konstante su iskazne formule;
- ako su A i B iskazne formule, onda su i (¬A), (A ˆ B), (A ˇ B), (A ⇒ B) i (A ⇔ B) iskazne formule.

Primer 2

- types variable = nat
- datatype Formula =
  T
  | F
  | Var variable
  | Neg Formula
  | And Formula Formula (infixl "And" 105)
  | Or Formula Formula (infixl "Or" 104)
  | Imp Formula Formula (infixl "Imp" 103)
  | Iff Formula Formula (infixl "Iff" 102)

Funkcije

- Notacija $[A_1, \ldots, A_n] \Rightarrow B$

  zamenjena je sa:

  $A_1 \Rightarrow \ldots \Rightarrow A_n \Rightarrow B$

Funkcije

- Ključna reč definition (nerekurzivna definicija)
- Primer:

```
definition zbir :: “nat ⇒ nat ⇒ nat”
where
“zbir x y = x + y”
```

Funkcije

- Ključna reč value
- Primer:

```
value “zbir 3 5”
```

Funkcije

- Ključna reč primrec (rekurzivna definicija)
- Primer:

```
primrec sum :: "nat ⇒ nat“
where  "sum 0 = 0“
       | "sum (Suc n) = sum n + (Suc n)"
```

Primer

- Interpretaciju Iv denisemo na sledeći način:

- Iv(p) = v(p), za svaki element p skupa P;
- Iv(T) = 1 i Iv(F) = 0;
- Iv(¬ A) = 1 ako je Iv(A) = 0 i Iv(¬ A) = 0 ako je Iv(A) = 1;
- Iv(A ˄ B) = 1 ako je Iv(A) = 1 i Iv(B) = 1; Iv(A ˄ B) = 0 inace;
- Iv(A ˅ B) = 0 ako je Iv(A) = 0 i Iv(B) = 0 ; Iv(A ˅ B) = 1 inace;
- Iv(A ⇒ B) = 0 ako je Iv(A) = 1 i Iv(B) = 0 ; Iv(A ⇒ B) = 1 inace;
- Iv(A ⇔ B) = 1 ako je Iv(A) = Iv(B); Iv(A ⇔ B) = 0 inace.

Primer

- primrec model :: "Formula ⇒ Valuation ⇒bool“
  where
  "model T valuation = True“
  | "model F valuation = False”
  | "model (Var p) valuation = (p ∈ valuation)“
  | "model (Neg A) valuation = (¬ model A valuation)“
  | "model (A And B) valuation = (model A valuation ˆ model B valuation)“
  | "model (A Or B) valuation = (model A valuation ˇ model B valuation)“
  | "model (A Imp B) valuation = (¬ model A valuation ˇ model B valuation)“
  | "model (A Iff B) valuation = (model A valuation = model B valuation)"

- Ključna reč infixl
- Primer:

```
definition
  models  :: "Valuation ⇒ Formula ⇒ bool"
  (infixl “ ╞" 101)
  where "v  ╞ A = model A v"
```

⟦	[|	\<lbrakk>
⟧	|]	\<rbrakk>
⟹	==>	\<Longrightarrow>
⋀	!!	\<And>
≡	==	\<equiv>
⇌	==	\<rightleftharpoons>
↽	=>	\<rightharpoonup>
↼	<=	\<leftharpoondown>
λ	%	\<lambda>
⇒	=>	\<Rightarrow>
∧	&	\<and>
∨	|	\<or>
⟶	-->	\<longrightarrow>
¬	~	\<not>
≠	~=	\<noteq>
∀	ALL, !	\<forall>
∃	EX, ?	\<exists>
∃!	EX!, ?!	\<exists>!
ε	SOME, @	\<epsilon>
∘	o	\<circ>
| |	abs	\<bar> \<bar>
≤	<=	\<le>
×	*	\<times>
∈	:	\<in>
∉	~:	\<notin>
⊆	<=	\<subseteq>
⊂	<	\<subset>
∪	Un	\<union>
∩	Int	\<inter>
⋃	UN, Union	\<Union>
⋂	INT, Inter	\<Inter>
*	^*	\<^sup>*
−1	^-1	\<inverse>

Mathematical Symbols, Their ASCII-Equivalents and Internal Names

Simlifikacija u Isabelle

- Ključna reč simp; moguće koristiti uz definicje novih tipova, funkcija, uz leme, teoreme...
- Primer:

```
definition models  :: "Valuation ⇒ Formula ⇒
bool"    (infixl "╞" 101)
 where [simp]: “v  ╞ A = model A v"
```

Simlifikacija u Isabelle

- Može se desiti da se simlifikacija ne može završiti, pravila se primenjuju slepo sleva na desno.
- Primer:

  f(x) = g(x), g(x) = f(x)

Teoreme

- Ključne reči lemma, theorem
- lemma (*ime :) "tvrđenje ....."*
- Primer:

  lemma "A $\rightarrow$A ˆ A"

  theorem pom: "A $\rightarrow$B ˇ A"

Teoreme

- U definiciji se mogu koristiti ključne reči assumes, shows
- Primer:

```
theorem thm_2_2:
assumes "tautology A" "tautology (A Imp B)“
shows "tautology B"
```

Tipični Isar dokazi

- **proof**

  assume formula$_0$

  have formula$_1$ **by** *simp*

  ...

  have formula$_n$ **by** *auto*

  show formula$_{n+1}$ **by** ....

  qed

auto i simp

- auto deluje na sve podciljeve
- simp deluje samo na prvi podcilj
- auto je širi od simp

Primeri

- ...

Zaključak