Univerzitet u Beogradu
Matematički fakultet
Katedra za računarstvo i informatiku

Šifra predmeta: R265
14.03.2023.

Uvod u interaktivno dokazivanje teorema

Vežbe 4

Zadatak 1 *Intuicionistička pravila prirodne dedukcije u logici prvog reda*

Diskutovati o pravilima uvođenja i pravilima eliminacije prirodne dedukcije u logici prvog reda. Pomoću ključne reči *thm* ispitati svako pravilo prirodne dedukcije. Primeniti odgovarajuće pravilo prirodne dedukcije na jednostavnim formulama i diskutovati o cilju koga treba dokazati pre i posle primene tog pravila.

Za logiku prvog reda pored pravila prirodne dedukcije iskazne logike, važe i pravila uvođenja i elimenacije kvantifikatora.

Uvođenje univerzalnog kvantifikatora: *allI*

thm *allI*

lemma $\forall\ x.\ P\ x$
  apply (*rule allI*)

Eliminacija univerzalnog kvantifikatora: *allE*

thm *allE*

lemma $\forall\ x.\ P\ x \Longrightarrow A$
  apply (*erule-tac* $x = t$ **in** *allE*)

Uvođenje egzistencijalnog kvantifikatora: *exI*

thm *exI*

lemma $\exists\ x.\ P\ x$
  apply (*rule-tac* $x = t$ **in** *exI*)

Eliminacija egzistencijalnog kvantifikatora: *exE*

thm *exE*

lemma $\exists\ x.\ P\ x \Longrightarrow A$
  apply (*erule exE*)

Zadatak 2 *Dokazi u prirodnoj dedukciji*

Pokazati da su sledeće formule valjane u logici prvog reda. Dozvoljeno je korišćenje samo intuicionističkih pravila prirodne dedukcije.

lemma $(\forall\ x.\ Man\ x \longrightarrow Mortal\ x) \wedge Man\ Socrates \longrightarrow Mortal\ Socrates$
  apply (*rule impI*)

```
  apply (erule conjE)
  apply (erule-tac x = Socrates in allE)
  apply (erule impE)
   apply assumption +
  done

lemma de-Morgan-1: (∃ x. ¬ P x) ⟶ ¬ (∀ x. P x)
  apply (rule impI)
  apply (rule notI)
  apply (erule exE)
  apply (erule-tac x = x in allE)
  apply (erule notE)
  apply assumption
  done

lemma de-Morgan-2: (∀ x. ¬ P x) ⟶ (∄ x. P x)
  apply (rule impI)
  apply (rule notI)
  apply (erule exE)
  apply (erule-tac x = x in allE)
  apply (erule notE)
  apply assumption
  done

lemma de-Morgan-3: (∄ x. P x) ⟶ (∀ x. ¬ P x)
  apply (rule impI)
  apply (rule allI)
  apply (rule notI)
  apply (erule notE)
  apply (rule-tac x = x in exI)
  apply assumption
  done

lemma (∃ x. P x) ∧ (∀ x. P x ⟶ Q x) ⟶ (∃ x. Q x)
  apply (rule impI)
  apply (erule conjE)
  apply (erule exE)
  apply (erule-tac x = x in allE)
  apply (rule-tac x = x in exI)
  apply (erule impE)
   apply assumption +
  done

lemma (∀ m. Man m ⟶ Mortal m) ∧
       (∀ g. Greek g ⟶ Man g) ⟶
       (∀ a. Greek a ⟶ Mortal a)
  apply (rule impI)
  apply (erule conjE)
  apply (rule allI)
  apply (rule impI)
  apply (erule-tac x = a in allE) +
```

```
  apply (erule impE) +
   apply assumption +
  done
```

Dodatni primeri:

```
lemma (∀ a. P a ⟶ Q a) ∧ (∀ b. P b) ⟶ (∀ x. Q x)
  apply (rule impI)
  apply (erule conjE)
  apply (rule allI)
  apply (erule-tac x = x in allE) +
  apply (erule impE)
   apply assumption +
  done

lemma (∃ x. A x ∨ B x) ⟶ (∃ x. A x) ∨ (∃ x. B x)
  apply (rule impI)
  apply (erule exE)
  apply (erule disjE)
   apply (rule disjI1)
   apply (rule-tac x = x in exI)
   apply assumption
  apply (rule disjI2)
  apply (rule-tac x = x in exI)
  apply assumption
  done

lemma (∀ x. A x ⟶ ¬ B x) ⟶ (∄ x. A x ∧ B x)
  apply (rule impI)
  apply (rule notI)
  apply (erule exE)
  apply (erule conjE)
  apply (erule-tac x = x in allE)
  apply (erule impE)
   apply assumption
  apply (erule notE)
  apply assumption
  done
```

Formulisati i dokazati naredna tvrđenja.

Ako za svaki broj koji nije paran važi da je neparan;
i ako za svaki neparan broj važi da nije paran;
pokazati da onda za svaki broj važi da nije istovremeno i paran i neparan

```
lemma
    (∀ x. ¬ Paran x ⟶ Neparan x) ∧
    (∀ x. Neparan x ⟶ ¬ Paran x) ⟶
    (∀ x. ¬ (Paran x ∧ Neparan x))
  apply (rule impI)
  apply (erule conjE)
  apply (rule allI)
  apply (rule notI)
  apply (erule conjE)
```

```
  apply (erule-tac x = x in allE) +
  apply (erule impE) +
    apply assumption +
  apply (erule impE)
   apply assumption
  apply (erule notE)
  apply assumption
  done
```

Ako svaki konj ima potkovice;
i ako ne postoji čovek koji ima potkovice;
i ako znamo da postoji makar jedan čovek;
dokazati da postoji čovek koji nije konj.

```
lemma (∀ x. Konj x ⟶ Potkovice x) ∧
       (∄ x. Covek x ∧ Potkovice x) ∧
       (∃ x. Covek x) ⟶
       (∃ x. Covek x ∧ ¬ Konj x)
  apply (rule impI)
  apply (erule conjE) +
  apply (erule exE)
  apply (erule-tac x = x in allE)
  apply (rule-tac x = x in exI)
  apply (rule conjI)
   apply assumption
  apply (rule notI)
  apply (erule impE)
   apply assumption
  apply (erule notE)
  apply (rule-tac x = x in exI)
  apply (rule conjI)
   apply assumption +
  done
```

Ako je svaki kvadrat romb;
i ako je svaki kvadrat pravougaonik;
i ako znamo da postoji makar jedan kvadrat;
onda postoji makar jedan romb koji je istovremeno i pravougaonik.

```
lemma (∀ x. Kvadrat x ⟶ Romb x) ∧
       (∀ x. Kvadrat x ⟶ Pravougaonik x) ∧
       (∃ x. Kvadrat x) ⟶
       (∃ x. Romb x ∧ Pravougaonik x)
  apply (rule impI)
  apply (erule conjE) +
  apply (erule exE)
  apply (erule-tac x = x in allE) +
  apply (rule-tac x = x in exI)
  apply (rule conjI)
   apply (erule impE)
    apply assumption +
  apply (erule impE) +
    apply assumption +
  apply (erule impE)
```

apply *assumption* +
done

Ako je relacija R simetrična, tranzitivna
i ako za svako x postoji y koje je sa njim u relaciji,
onda je relacija R i refleksivna.

Savet: Pomoću ključne reči *definition* definisati osobinu refleksivnosti, tranzitivnosti i simetricnosti. Ta formulisati tvđenje i dokazati ga. Podsetiti se ključne reči *unfolding* za raspisivanje definicije.

definition *reflexive* $R \equiv \forall\ x.\ R\ x\ x$

definition *transitive* $R \equiv \forall\ x\ y\ z.\ R\ x\ y \wedge R\ y\ z \longrightarrow R\ x\ z$

definition *symmetric* $R \equiv \forall\ x\ y.\ R\ x\ y \longleftrightarrow R\ y\ x$

lemma *symmetric* R $\wedge$ *transitive* R $\wedge$
$(\forall\ x.\ \exists\ y.\ R\ x\ y) \longrightarrow$
reflexive R
unfolding *reflexive-def transitive-def symmetric-def*
apply (*rule impI*)
apply (*erule conjE*) +
apply (*rule allI*)
apply (*erule-tac* $x = x$ **in** *allE*) **back back**
apply (*erule exE*)
apply (*erule-tac* $x = x$ **in** *allE*)
apply (*erule-tac* $x = x$ **in** *allE*)
apply (*erule-tac* $x = y$ **in** *allE*)
apply (*erule-tac* $x = y$ **in** *allE*)
apply (*erule-tac* $x = x$ **in** *allE*)
apply (*erule iffE*)
apply (*erule impE*)
apply (*rule conjI*)
apply *assumption*
apply (*erule impE*)
apply *assumption* +
done

Zadatak 3 *Klasična pravilo prirodne dedukcije: ccontr.*

Diskutovati zašto sledeće tvrđenje može biti dokazano samo intuicionističkim pravilima prirodne dedukcije, dok to ne važi za tvrđenje nakon njega. Primetiti razliku između pravila *notI* i *ccontr*.

lemma $\langle A \longrightarrow \neg\ \neg\ A \rangle$
apply (*rule impI*)
apply (*rule notI*)
apply (*erule notE*)
apply *assumption*
done

thm *notI*
thm *ccontr*

lemma $\neg \neg A \longrightarrow A$
apply (*rule impI*)
apply (*rule ccontr*)
apply (*erule notE*)
apply *assumption*
done

Dokazati sledeća tvrđenja:

lemma $(\neg P \longrightarrow P) \longrightarrow P$
apply (*rule impI*)
apply (*rule ccontr*)
apply (*erule impE*)
apply *assumption*
apply (*erule notE*)
apply *assumption*
done

lemma $\neg (A \wedge B) \longrightarrow \neg A \vee \neg B$
apply (*rule impI*)
apply (*rule ccontr*)
apply (*erule notE*)
apply (*rule conjI*)
apply (*rule ccontr*)
apply (*erule notE*)
apply (*rule disjI1*)
apply *assumption*
apply (*rule ccontr*)
apply (*erule notE*)
apply (*rule disjI2*)
apply *assumption*
done

lemma $(\neg (\forall\ x.\ P\ x)) \longrightarrow (\exists\ x.\ \neg P\ x)$
apply (*rule impI*)
apply (*rule ccontr*)
apply (*erule notE*)
apply (*rule allI*)
apply (*rule ccontr*)
apply (*erule notE*)
apply (*rule-tac* $x = x$ **in** *exI*)
apply *assumption*
done

Dodatni primeri:

lemma $(\neg B \longrightarrow \neg A) \longrightarrow (A \longrightarrow B)$
apply (*rule impI*)
apply (*rule impI*)
apply (*rule ccontr*)
apply (*erule impE*)
apply *assumption*
apply (*erule notE*) **back**
apply *assumption*

```
  done

lemma (A ⟶ B) ⟶ (¬ A ∨ B)
  apply (rule impI)
  apply (rule ccontr)
  apply (erule impE)
   apply (rule ccontr)
   apply (erule notE)
   apply (rule disjI1)
   apply assumption
  apply (erule notE)
  apply (rule disjI2)
  apply assumption
  done

lemma (¬ P ⟶ Q) ⟷ (¬ Q ⟶ P)
  apply (rule iffI)
   apply (rule impI)
   apply (rule ccontr)
   apply (erule impE)
    apply assumption
  apply (erule notE)
   apply assumption
  apply (rule impI)
  apply (rule ccontr)
  apply (erule impE)
   apply assumption
  apply (erule notE)
  apply assumption
  done

lemma ((P ⟶ Q) ⟶ P) ⟶ P
  apply (rule impI)
  apply (rule ccontr)
  apply (erule impE)
   apply (rule impI)
   apply (rule ccontr)
   apply (erule notE)
   apply assumption
  apply (erule notE)
  apply assumption
  done
```

Zadatak 4 *Klasična pravilo prirodne dedukcije: classical.*

Pokazati naredna tvrđenja pomoću pravila *classical*. Zgodna alternativa ovog pravila je razdvajanje na slučajeve neke podformule.

```
thm classical

lemma P ∨ ¬ P
  apply (rule classical)
```

apply (*rule disjI1*)
apply (*rule ccontr*)
apply (*erule notE*)
apply (*rule disjI2*)
apply *assumption*
done

lemma $(A \longleftrightarrow (A \longleftrightarrow B)) \longrightarrow B$
apply (*rule impI*)
apply (*cases A*)
apply (*erule iffE*)
apply (*erule impE*)
apply *assumption*
apply (*erule iffE*)
apply (*erule impE*) **back**
apply *assumption* +
apply (*rule ccontr*)
apply (*erule iffE*)
apply (*erule impE*) **back**
apply (*rule iffI*)
apply (*erule notE*)
apply *assumption*
apply (*erule impE*)
apply (*erule notE*) **back**
apply *assumption*
apply (*erule notE*) **back**
apply *assumption*
apply (*erule notE*)
apply *assumption*
done

Paradoks pijanca:
Postoji osoba za koju važi, ako je on pijanac onda su i svi ostali pijanci.

lemma *drinker's-paradox*: $\exists\ x.\ drunk\ x \longrightarrow (\forall\ x.\ drunk\ x)$
apply (*cases* $\forall\ x.\ drunk\ x$)
apply (*rule exI*)
apply (*rule impI*)
apply *assumption*
apply (*rule ccontr*)
apply (*erule notE*)
apply (*rule allI*)
apply (*rule ccontr*)
apply (*erule notE*)
apply (*rule-tac* $x = x$ **in** *exI*)
apply (*rule impI*)
apply (*erule notE*)
apply *assumption*
done