

Perlen der Informatik 2 5. Übung

1 Natürliches Schließen in der Aussagenlogik

Für die Beweise auf diesem Übungsblatt gelten die folgenden Spielregeln:

- Es dürfen nur die Methoden *rule*, *erule* und *assumption* verwendet werden (ggf. auch *drule* und *frule*).

- Es dürfen folgende Regeln verwendet werden:

TrueI: *True*,

FalseE: *False* $\implies A$,

notI: $(A \implies \text{False}) \implies \neg A$,

notE: $\neg A \implies A \implies B$,

conjI: $A \implies B \implies A \wedge B$,

conjE: $A \wedge B \implies (A \implies B \implies C) \implies C$,

disjI1: $A \implies A \vee B$,

disjI2: $A \implies B \vee A$,

disjE: $A \vee B \implies (A \implies C) \implies (B \implies C) \implies C$,

impI: $(A \implies B) \implies A \longrightarrow B$,

impE: $A \longrightarrow B \implies A \implies (B \implies C) \implies C$,

mp: $A \longrightarrow B \implies A \implies B$

iffI: $(A \implies B) \implies (B \implies A) \implies A \longleftrightarrow B$,

iffE: $A \longleftrightarrow B \implies (A \longrightarrow B \implies B \longrightarrow A \implies C) \implies C$

ccontr: $(\neg A \implies \text{False}) \implies A$

- Es dürfen alle Regeln verwendet werden, die unter den hier genannten Einschränkungen hergeleitet wurden.
- Andere Regeln dürfen nicht verwendet werden.

1.1 Mit *apply*-Skripten

Finden Sie Beweise im *apply*-Stil für die folgenden Aussagen:

$$(A \wedge B) \wedge C \longrightarrow A \wedge B \wedge C$$

$$(A \vee B) \vee C \longrightarrow A \vee B \vee C$$

$$(A \longrightarrow B) \longrightarrow \neg B \longrightarrow \neg A$$

1.2 Mit strukturierten Beweisen

1.2.1 intuitionistische Schlüsse

Finden Sie strukturierte Isar-Beweise für die folgenden Aussagen:

$$(A \longrightarrow B \longrightarrow C) \longrightarrow (A \longrightarrow B) \longrightarrow A \longrightarrow C$$

$$A \wedge (B \vee C) \longleftrightarrow A \wedge B \vee A \wedge C$$

$$\text{de_morgan_1: } \neg (A \vee B) \Longrightarrow \neg A \wedge \neg B$$

1.2.2 klassische Schlüsse

Beweise, die die nicht-intuitionistische Regel *ccontr* (classical contradiction) benötigen, sind meist schwieriger. Zeigen Sie:

$$\text{de_morgan_2: } \neg (A \wedge B) \Longrightarrow \neg A \vee \neg B$$

2 Eine unflätige Funktion

Hinweis: In dieser Aufgabe dürfen alle bekannten Methoden und Theoreme verwendet werden.

Die folgende Funktion stellt die Struktur einer vormals “flachgeklopften” Liste wieder her:

```
fun unflat :: "'b list list => 'a list => 'a list list" where
  "unflat (xs # xss) ys = (let
    ps = take (length xs) ys;
    qs = drop (length xs) ys
  in ps # unflat xss qs)"
| "unflat _ [] = []"
| "unflat [] ys = [ys]"
```

▷ Zeigen Sie:

$$\text{unflat } xss \text{ (concat } xss) = xss$$

▷ Zeigen Sie weiter:

$$\text{unflat (rev } xss) \text{ (rev (concat } xss)) = rev (map rev } xss)$$

Zeigen Sie dazu in einem strukturierten Isar-Beweis als Teilaussage (“**have**”):

$$\bigwedge yss. \text{unflat } xss \text{ (rev (concat (rev } yss))) = rev (map rev (rev } yss))$$

Setzen Sie dann (“**then have**”) für die gebundene Variable *yss* den Term *rev xss* ein (“.” ⇒ “**by this**”). Damit gelangen Sie ans Ziel.