

Perlen der Informatik 3

8. Übung

Aufgabe 8.1 Aussagenlogik in Isabelle/Isar

Finde einen strukturierten Isabelle/Isar-Beweis für folgenden Aussagen:

- (a) $(A \rightarrow B \rightarrow C) \rightarrow (A \rightarrow B) \rightarrow (A \rightarrow C)$
- (b) $A \wedge (B \vee C) \iff (A \wedge B) \vee (A \wedge C)$
- (c) $\neg(A \vee B) \rightarrow (\neg A \wedge \neg B)$

Benutze dafür nur `proof (rule ...)`, `by (rule ...)` als Beweismethoden.

Aufgabe 8.2 Prädikatenlogik in Isabelle/Isar

- (a) $(\forall x. P(x) \rightarrow Q(x)) \wedge \neg Q(a) \rightarrow \neg P(a)$
- (b) $(\exists x. P(x)) \wedge (\forall x. P(x) \rightarrow Q(x)) \longrightarrow (\exists x. Q(x))$
- (c) $(\exists x. D(x)) \longrightarrow (\forall y. D(y))$ („Drinker’s Principle“)

Aufgabe 8.3 Intervall-Listen

Wir wollen Mengen von natürlichen Zahlen als geordnete Liste von Intervallen darstellen, und betrachten die Operation, die eine Zahl in eine derart repräsentierte Menge einfügt.

Gegeben sei `type_synonym interval = nat × nat`.

- (a) Intervalle müssen sortiert und optimal sein, d.h. $[(0, 1), (2, 3)]$ ist nicht erlaubt! Definiere die Invariante als Funktion `inv :: interval list ⇒ bool`.
- (b) Definiere die *Abstraktionsfunktion* für Intervall-Listen: `set_of :: interval list ⇒ nat set`.
- (c) Definiere eine Einfügeoperation `ins :: nat ⇒ interval list ⇒ interval list`, die die Invariante erhält.
- (d) Beweise die Korrektheit der Einfügeoperation: `inv xs ⇒ set_of (ins x xs) = insert x (set_of xs)`
- (e) *Bonusaufgabe*: Beweise, dass die Einfügeoperation die Invariante erhält.