Serge Autexier
Christoph Lüth

3. Übungsblatt

Ausgabe: 14.05.13
Abgabe: 27.05.13

3.1 *Vom Umgang mit Quantoren* *5 Punkte*

Zeigen Sie folgendes Lemma in Isabelle:

$$(\exists x.\phi(x) \wedge \psi) \longleftrightarrow (\exists x.\phi(x)) \wedge \psi \quad x \notin FV(\psi)$$

Hinweise:

(i) Formulieren Sie zuerst das Lemma in Isabelle.

(ii) Der Beweis wird übersichtlicher, wenn beide Richtungen der Äquivalenz als getrennte Lemmata gezeigt werden.

(iii) Entscheidend für das Gelingen des Beweises in Isabelle ist es, zum richtigen Zeitpunkt die Quantoren in Vorbedingung und Konklusion zu eliminieren, damit die Eigenvariablenbedingung (ausgedrückt durch den „Meta-Allquantor“ `!!x. A x`) die geeignete Instanttierung zuläßt.

(iv) Der Beweis von rechts nach links wird einfacher, wenn das Lemma in der Form `[| A; B |] ==> C` statt `A & B --> C` formuliert wird.

(v) Wenn Ihnen der Beweis in Isabelle nicht gelingt, versuchen Sie zuerst eine Herleitung als ND-Baum.

3.2 *Vom Umgang mit Zahlen und Gleichungen* *5 Punkte*

Zeigen Sie folgende Aussage (Kommutativität der Multiplikation) in der Peano-Arithmetik, zuerst auf Papier (als Gleichungsherleitung) und dann in Isabelle:

$$a \cdot b = b \cdot a$$

Folgende Lemmata könnten dabei hilfreich sein (die ggf. erst bewiesen werden müssen, oder in der Übung bewiesen wurden):

$$Z = b \cdot Z \tag{1}$$
$$b + b \cdot a = b \cdot Sa \tag{2}$$

3.3 *Vom Umgang mit Listen* *5 Punkte*

Ausgehend von der auf der Webseite vorgestellten Modellierung von Listen in der Theorie `VSList.thy` definieren Sie rekursiv eine Funktion `count x xs`, welche zählt, wie oft ein Element `x` in einer Liste `xs` auftritt.

Zeigen Sie — wie gewohnt erst als Gleichungsumformung und dann in Isabelle — dass ein Element in der Verkettung zweier Listen so oft auftritt wie es summiert in beiden Listen einzeln auftritt:

```
count x (as++ bs) = count x as+ count x bs
```

3.4 *Vom Umgang mit Graphen* *5 Punkte*

In dieser Aufgabe wollen wir die Modellierung von Graphen mit der Prädikatenlogik erster Stufe weiter betrachten, und in Isabelle formalisieren. Diese Teilaufgabe kommt ganz ohne Beweise aus, hier geht es mehr darum, einen Sachverhalt in dem uns momentan zur Verfügung stehenden logischen Rahmenwerk — Prädikatenlogik erster Stufe mit Induktion — zu formalisieren, und die Möglichkeiten und Grenzen dieser Formalisierung zu erkennen.

- *Elementare Konzepte*:

  Ein Graph soll modelliert werden durch drei Typen `G`, `V` und `E` für Graph, Knoten (*vertices*) und Kanten (*edges*), zusammen mit zwei Prädikaten `vert` und `edge`:

  $$\texttt{vert g v} \quad \longleftrightarrow \quad \texttt{v} \text{ ist ein Knoten in } \texttt{g}$$
  $$\texttt{edge g v w e} \quad \longleftrightarrow \quad \texttt{e} \text{ ist eine Kante in } \texttt{g} \text{ von } \texttt{v} \text{ nach } \texttt{w}$$

  Mit diesen Prädikaten und dem elementaren Datentyp `list` können wir dann das Konzept eines Pfades rekursiv definieren: `path g v w p` ist in einem Graphen `g` ein Pfad von dem Knoten `v` zum Knoten `w`, bestehend aus einer Liste von Kanten `p`, wenn:

  - `v` ein Knoten in `g` ist, und
  - entweder `p` ist leer und `v= w`, oder
  - `p` ist nicht leer und besteht aus einer Kante `e` und einem Rest `es`, und es gibt einen Knoten `u` in `g`, so dass `e` eine Kante von `v` nach `u` und `es` wiederum ein Pfad in `g` von `u` nach `w` ist.

  Formalisieren Sie diese Konzepte axiomatisch in Isabelle.

- *Fortgeschrittene Konzepte:*

  Ein *Zyklus* ist ein Pfad in `g` von `v` nach `v`; ein *Hamilton-Zyklus* ist ein Zyklus in `g`, in dem jeder Knoten in `g` genau einmal besucht wird.

  Um dieses Konzept in Isabelle zu formalisieren benötigen wir beispielsweise folgende Hilfsfunktionen:

  - ein Prädikat `all_vertices g vs`, welches wahr ist, wenn jeder Knoten `v` in `g` in der Liste `vs` genau einmal auftritt.
  - ein Prädikat `visited_vertices g vs es`, welches wahr ist, wenn `es` ein Pfad in `g` ist, welcher die Knoten `vs` besucht.

  Dann ist ein Hamilton-Zyklus ein Zyklus `es`, so dass es eine Liste `vs` von Knoten gibt, die alle Knoten von `g` enthält, und die von dem Pfad `es` besucht wird.

- *Und weiter?*

  (i) Sind die Modelle unserer Logik Graphen im mathematischen Sinne (mit einer Menge V von Knoten und einer Menge E von Kanten und zwei Abbildungen $\rho_s, \rho_t : E \to V$, die jeder Kante einen Start- und Zielknoten zuordnen)? Wenn ja, sind alle Modelle dieser Art, oder gibt es auch hier Nicht-Standard-Modelle?

  (ii) Leider können wir mit unserem Konzept von Graphen keinen Algorithmus angeben, der einen Hamiltonzyklus berechnet. Was genau fehlt uns dazu?