
Übungen zur Vorlesung Logik
Blatt 3

Prof. Dr. Roland Meyer

Abgabe bis Freitag, 29.05.2015, 12:00 Uhr

Aufgabe 3.1 [Mehr Beweise im Kalkül \mathcal{F}_0 , $\frac{1}{2}$ Punkt pro Teilaufgabe]

Zeigen Sie:

- a) $\neg(q \rightarrow p) \vdash_{\mathcal{F}_0} \neg p$
- b) $\vdash_{\mathcal{F}_0} \neg(p \rightarrow p) \rightarrow \neg(p \rightarrow q)$
- c) $q, r \rightarrow \neg q \vdash_{\mathcal{F}_0} \neg r$
- d) $p \rightarrow (\neg q \rightarrow r), (\neg q \rightarrow r) \rightarrow (p \rightarrow \neg q), \neg r \vdash_{\mathcal{F}_0} \neg p$

Sie können die Lemmata und Beispiele aus der Vorlesung (bzw. vom Präsenzblatt 2), die Inkonsistenzregel und das Deduktionstheorem verwenden, *nicht jedoch* die Vollständigkeit von \mathcal{F}_0 .

Aufgabe 3.2 [Beweise im Gentzen-Sequenzkalkül, $\frac{2}{3}$ Punkt pro Teilaufgabe]

Zeigen Sie:

- a) $\neg(p \rightarrow q) \vdash_G q \rightarrow p$
- b) $\vdash_G (p \wedge q) \rightarrow p \vee r$
- c) $s \wedge r, r \rightarrow \neg(p \wedge q) \vdash_G \neg p, \neg q$

Hinweis: Notieren Sie die Beweise wie in der Vorlesung bottom-up und baumartig.

(Siehe auch entsprechendes Beispiel im PDF "4. Vollständigkeit, Sequenzen, Tableaus"[1] auf der Homepage.)

Aufgabe 3.3 [Online-Beweischecker, 1 Punkt]

Sie finden auf der Website zur Vorlesung einen Beweischecker für das Kalkül \mathcal{F}_0 (<https://concurrency.cs.uni-kl.de/sat/proofchecker/>). Testen Sie diesen, indem Sie einen der \mathcal{F}_0 -Beweise, die Sie bereits gesehen haben (z.B. die Lemmata aus der Vorlesung oder die Beweise zu Präsenzaufgabe 2.4) abtippen und überprüfen lassen.

Geben Sie uns in Ihrer Abgabe Feedback zum Beweischecker. Ist die Dokumentation ausreichend? Nennen Sie uns Probleme, Fehler und unerwartetes Verhalten, auf das Sie gestossen sind. Ihre Antwort sollte mindestens 8 Zeilen umfassen, um diese Aufgabe sinnvoll zu bearbeiten.

Bitte umdrehen!

[1] http://concurrency.cs.uni-kl.de/documents/Logik_SoSe_2015/vollstaendigkeit_sequenzen_tableaus_W5.pdf

Aufgabe 3.4 [Matching von Schemata, 1 Punkt pro Teilaufgabe]

Wenn Sie einen Beweischecker wie den oben erwähnten selbst programmieren wollen, muss dieser unter anderem gegebene Formeln auf Schemata matchen können, um zu entscheiden, ob eine Regel korrekt angewandt wurde. Das heisst, er muss, wenn ein Formelschema und eine Formel gegeben sind, entscheiden können, ob die Formel eine Instanz des Schemas ist.

Sei zum Beispiel das Schema

$$A \rightarrow (B \rightarrow A)$$

und die Formel

$$(p \rightarrow q) \rightarrow ((q \rightarrow p) \rightarrow (p \rightarrow q)).$$

gegeben.

- a) Überlegen Sie sich, wie sie dieses Problem algorithmisch lösen können und demonstrieren Sie dies am gegebenen Beispiel.
- b) Geben Sie einen Algorithmus an, der das allgemeine Problem möglichst effizient löst.

Hinweis: Sie können davon ausgehen, dass sowohl Formel als auch Schema vollständig geklammert sind. Überlegen Sie sich, wie Sie in diesem Fall die Struktur einer Formel durch einen Baum beschreiben können.

Abgabe: bis Freitag, 29.05.2015, 12:00 Uhr im Kasten neben Raum 34/401.4