Technische Universität München **SS 2000**
Institut für Informatik **16. 5. 2000**
Prof. Tobias Nipkow, Ph. D. **Aufgabenblatt 4**

Praktikum Spezifikation und Verifikation

1 Natürliches Schließen

In diesem Aufgabenblatt geht es um den Kalkül des natürlichen Schießens, mit dessen Hilfe einige Lemmas der reinen Aussagen-Logik bewiesen werden sollen.

Für die Beweise gelten die folgenden Spielregeln:

- Es dürfen nur die Lemmas
  notI: $(A \Longrightarrow False) \Longrightarrow \neg A$,
  notE: $[\![\neg A; A]\!] \Longrightarrow B$,
  conjI: $[\![A; B]\!] \Longrightarrow A \wedge B$,
  conjE: $[\![A \wedge B; [\![A; B]\!] \Longrightarrow C]\!] \Longrightarrow C$,
  disjI1: $A \Longrightarrow A \vee B$,
  disjI2: $A \Longrightarrow B \vee A$,
  disjE: $[\![A \vee B; A \Longrightarrow C; B \Longrightarrow C]\!] \Longrightarrow C$,
  impI: $(A \Longrightarrow B) \Longrightarrow A \longrightarrow B$,
  impE: $[\![A \longrightarrow B; A; B \Longrightarrow C]\!] \Longrightarrow C$,
  mp: $[\![A \longrightarrow B; A]\!] \Longrightarrow B$
  iffI: $[\![A \Longrightarrow B; B \Longrightarrow A]\!] \Longrightarrow A = B$,
  iffE: $[\![A = B; [\![A \longrightarrow B; B \longrightarrow A]\!] \Longrightarrow C]\!] \Longrightarrow C$
  und für die klassischen Beweise zusätzlich
  classical: $(\neg A \Longrightarrow A) \Longrightarrow A$
  verwendet werden.

- Es dürfen nur die Methoden ***rule***, ***erule*** und ***assumption*** verwendet werden.

1.1 Einige konstruktive Lemmas

lemma `I: "A --> A"` **oops**

lemma `"A & B --> A"` **oops**

lemma `"A & B --> B & A"` **oops**

lemma `"(A & B) --> (A | B)"` **oops**

lemma `K: "A --> B --> A"` **oops**

lemma `"(A | B) = (B | A)"` **oops**

lemma `"(A | A) = (A & A)"` **oops**

lemma `S: "(A --> B --> C) --> (A --> B) --> A --> C"` **oops**

lemma `"(A --> B) --> (A' --> B') --> A & A' --> B & B'"` **oops**

lemma `"(A --> B) --> (B --> C) --> A --> C"` **oops**

1.2 Einige klassische lemmas

lemma `"~ ~ A --> A"` **oops**

lemma `"(~ A --> B) --> (~ B --> A)"` **oops**

lemma `"((A --> B) --> A) --> A"` **oops**

lemma `"~(A --> ~B) --> A"` **oops**

lemma `"(~ (A & B)) = (~A | ~B)"` **oops**

lemma `"(A = B) = (B = (A::bool))"` **oops**

lemma `"~ (A = (~ A))"` **oops**

lemma `"A | ~ A"` **oops**

▷ **Abgabe: 23. 5. 2000**