Grundlagen der Logik und Logik-Programmierung SS 2012

Klausur [Probe]

Bitte vermerken Sie auf Ihrer Abgabe Ihren Namen und Ihre Matrikelnummer.

Aufgabe 1 Aussagenlogische Konsequenz

Zeigen oder widerlegen Sie, dass folgende logische Konsequenzen gelten:

1. $((A \wedge B) \rightarrow C) \models (A \rightarrow (B \rightarrow C))$,
2. $(A \rightarrow B) \models (B \rightarrow A)$,
3. $((A \rightarrow B) \wedge (B \rightarrow C) \wedge (C \rightarrow \neg A)) \models \bot$ (hier steht $\bot$ für *falsum*),
4. $((A \vee B) \wedge A) \models \neg B$.

Aufgabe 2 Aussagenlogische Resolution

Wenden Sie den Resolutionsalgorithmus an, um zu entscheiden, ob die KNF

$(A \vee B \vee \neg C) \wedge (A \vee C \vee B) \wedge (A \vee \neg D) \wedge (B \vee D) \wedge (\neg A \vee \neg B) \wedge (\neg A \vee B \vee D) \wedge (\neg D \vee \neg A)$

erfüllbar ist.

Aufgabe 3 Unifikation

Wenden Sie den Unifikationsalgorithmus aus der Vorlesung an, um zu entscheiden, ob die Terme

$$f(g(X, h(Z)), Z) \text{ und } f(g(h(Z), h(Z)), h(W))$$

unifizierbar sind und ggf. einen mgu zu berechnen. (Achtung: es ist durchaus von Bedeutung, dass die beiden Seiten gemeinsame Variablen verwenden!)

Aufgabe 4 Logische Programmierung und SLD-Resolution

Programmieren Sie in Prolog ein Ungeradheitsprädikat *odd* auf natürlichen Zahlen (definiert durch Null und Nachfolger wie in der Vorlesung) sowie ein Prädikat *allodd* auf Listen, das prüft, ob alle Einträge der Liste ungerade sind. Geben Sie zwei SLD-Refutationen gemäß der Prolog-Berechnungsregel für die Anfrage $\leftarrow \mathit{allodd}([X, Y \mid Z])$ an.

Aufgabe 5 Formale Deduktion in Logik erster Stufe

Geben Sie eine formale Herleitung mittels natürlicher Deduktion für das Trinkerparadoxon

$$\exists X. (p(X) \rightarrow \forall Y. p(Y))$$

an.

Klausur [Probe]

Bitte vermerken Sie auf Ihrer Abgabe Ihren Namen und Ihre Matrikelnummer.

Aufgabe 1 Aussagenlogische Konsequenz

Zeigen oder widerlegen Sie mittels Wahrheitstafeln, dass folgende logische Konsequenzen gelten:

1. $((A \wedge B) \rightarrow C) \models (A \rightarrow (B \rightarrow C))$,
2. $(A \rightarrow B) \models (B \rightarrow A)$,
3. $((A \rightarrow B) \wedge (B \rightarrow C) \wedge (C \rightarrow \neg A)) \models \bot$ (hier steht $\bot$ für *falsum*),
4. $((A \vee B) \wedge A) \models \neg B$.

Im negativen Fall reicht es, wenn Sie ein einzelnes Gegenbeispiel angeben.

Aufgabe 2 Aussagenlogische Resolution

Wenden Sie den Resolutionsalgorithmus an, um zu entscheiden, ob die KNF

$(A \vee B \vee \neg C) \wedge (A \vee C \vee B) \wedge (A \vee \neg D) \wedge (B \vee D) \wedge (\neg A \vee \neg B) \wedge (\neg A \vee B \vee D) \wedge (\neg D \vee \neg A)$

erfüllbar ist.

Aufgabe 3 Unifikation

Wenden Sie den Unifikationsalgorithmus aus der Vorlesung an, um zu entscheiden, ob die Terme

$$f(g(X, h(Z)), Z) \text{ und } f(g(h(Z), h(Z)), h(W))$$

unifizierbar sind und ggf. einen mgu zu berechnen. (Achtung: es ist durchaus von Bedeutung, dass die beiden Seiten gemeinsame Variablen verwenden!). Gefordert sind nicht nur die letztendliche Antwort, sondern auch die einzelnen vom Algorithmus durchgeführten Schritte.

Aufgabe 4 Logische Programmierung

Programmieren Sie in Prolog ein Prädikat *zip*/3, das eine Funktion implementiert, die die Elemente zweier Listen paarweise zu Zweierlisten verknüpft, d.h. es soll gelten

$$\mathit{zip}([a_1, \ldots, a_n], [b_1, \ldots, b_m], l) \iff l = [[a_1, b_1], \ldots, [a_k, b_k]]$$

mit $k = \min(n, m)$ – d.h. die übrigbleibenden Elemente der längeren Liste werden ignoriert. Programmieren Sie ferner eine Variante *zip2* von *zip*, in der die übrigbleibenden Elemente am Ende angehängt werden, d.h.

$$\mathit{zip2}([a_1, \ldots, a_n], [b_1, \ldots, b_m], l) \iff l = \begin{cases} [[a_1, b_1], \ldots, [a_m, b_m], a_{m+1}, \ldots, a_n] & (n \geq m) \\ [[a_1, b_1], \ldots, [a_n, b_n], b_{n+1}, \ldots, b_m] & (m > n). \end{cases}$$

Aufgabe 5 Formale Deduktion

Geben Sie eine formale Herleitung mittels natürlicher Deduktion für die Formel

$$((A \to (B \vee C)) \wedge (\neg(A \wedge B))) \to (A \to C)$$

an.

Aufgabe 6 Induktion

Sei $\Sigma = \{\mathit{fst}/2, \mathit{snd}/2\}$, und sei $\mathfrak{M}$ ein Σ-Modell, so dass für alle $x, y \in M$ gilt

$$\mathfrak{M}[\![\mathit{fst}]\!](x,y) = x \qquad \mathfrak{M}[\![\mathit{snd}]\!](x,y) = y.$$

(d.h. $\mathfrak{M} \models \forall X, Y.\, (\mathit{fst}(X,Y) = X \wedge \mathit{snd}(X,Y) = Y)$). Zeigen Sie durch Induktion über E, dass für jeden Term E mit $FV(E) = \{X_1, \dots, X_n\}$ ein $i \in \{1, \dots, n\}$ existiert, so dass für alle Umgebungen η gilt

$$\mathfrak{M}[\![E]\!]\eta = \eta(X_i)$$

(d.h. $\mathfrak{M} \models \forall X_1, \dots, X_n.\, E = X_i$).

Die Aufgaben sind untereinander gleich gewichtet; die Klausur ist mit der Hälfte der Punkte in jedem Fall bestanden.