



Artificial Intelligence, Wintersemester 2012/2013

Übungsblatt 12

Abgabe: 31.01.2013, Besprechung: 01.02.2013

Aufgabe 1 Prädikatenlogik erster Stufe [4 Punkte]

Mathematische Eigenschaften können auch als prädikatenlogische Formeln mit dem Prädikatensymbol $<$, den Funktionssymbolen $+$ und \times , und den Konstantensymbolen 0 und 1 ausgedrückt werden. Zusätzliche Prädikate können auch mit der Äquivalenzrelation (\Leftrightarrow) definiert werden.

- (a) Schreiben Sie die Eigenschaft " x ist gerade." als prädikatenlogische Formel.
- (b) Schreiben Sie die Eigenschaft " x ist eine Primzahl." als prädikatenlogische Formel.
- (c) Die Goldbachsche Vermutung besagt, dass jede gerade Zahl größer 2 als die Summe zweier Primzahlen ausgedrückt werden kann. Schreiben Sie diese Vermutung als prädikatenlogische Formel.

Aufgabe 2 Unifikation [5 Punkte]

Geben Sie für die folgenden Paare von atomaren Aussagen den kleinsten gemeinsamen Unifikator an, falls er existiert. Falls nicht, geben Sie den Grund für die Nichtunifizierbarkeit an.

- (a) $P(A, A, B), P(x, y, z)$.
- (b) $Q(y, G(A, B)), Q(G(x, x), y)$.
- (c) $Older(Father(y), y), Older(Father(x), Jerry)$.
- (d) $Knows(Father(y), y), Knows(x, x)$.
- (e) $F(H(y), x, a, G(x)), F(H(G(x)), H(G(z)), b, y)$

Aufgabe 3 Forward Chaining [5 Punkte]

Gegeben seien folgende Axiome:

1. $0 \leq 4$.
2. $5 \leq 9$.
3. $\forall x \quad x \leq x$.
4. $\forall x \quad x \leq x + 0$.
5. $\forall x \quad x + 0 \leq x$.
6. $\forall x, y \quad x + y \leq y + x$.
7. $\forall w, x, y, z \quad w \leq y \wedge x \leq z \Rightarrow w + x \leq y + z$.
8. $\forall x, y, z \quad x \leq y \wedge y \leq z \Rightarrow x \leq z$.

Beweisen Sie die Aussage $5 \leq 4 + 9$ mittels Forward Chaining. Achten Sie dabei darauf, dass Sie nur die hier gegebenen Axiome verwenden und nichts, was Sie evtl. sonst über Arithmetik wissen. Zeigen Sie nur die Schritte, die zum Erfolg führen, also keine irrelevanten Schritte.

Aufgabe 4 Skolemisierung ODER Prolog [6 Punkte]

Bei dieser Aufgabe müssen Sie nur a) **oder** b) bearbeiten, nicht beide.

a) Die folgenden Aufgaben beschäftigen sich mit **Substitution** und **Skolemisierung**

- (i) Unter der Voraussetzung $\forall x \exists y P(x, y)$ ist es nicht gültig, darauf zu schließen, dass $\exists q P(q, q)$. Geben Sie ein Beispiel für ein Prädikat P an, das die erste Aussage erfüllt, die zweite aber nicht.
- (ii) Angenommen, eine Inferenzmaschine wurde falsch geschrieben, weil der Occurs Check fehlt, so dass sie es erlaubt ein Literal wie $P(x, F(x))$ mit $P(q, q)$ zu unifizieren. (Die meisten Standard-Implementierungen von Prolog erlauben dies.) Zeigen Sie, dass solch eine Inferenzmaschine es erlaubt aus $\forall x \exists y P(x, y)$ die Schlussfolgerung $\exists q P(q, q)$ zu ziehen.
- (iii) Nehmen Sie an, dass eine Prozedur, die prädikatenlogische Formeln erster Stufe in Klausel-Form bringt, $\forall x \exists y P(x, y)$ fälschlicherweise zu $P(x, Sk_0)$ skolemisiert, d.h. es ersetzt y durch eine Skolem-Konstante statt durch eine Skolem-Funktion von x . Zeigen Sie, dass eine Inferenzmaschine, die eine solche Prozedur benutzt, es auch erlaubt aus der Voraussetzung $\forall x \exists y P(x, y)$ die Schlussfolgerung $\exists q P(q, q)$ zu inferieren.
- (iv) Ein häufiger Fehler von Studenten ist es, anzunehmen, dass es bei der Unifikation erlaubt ist, einen Term durch eine Skolem-Konstante zu ersetzen statt durch eine Variable. Es wird zum Beispiel behauptet, dass die Formeln $P(Sk_1)$ und $P(A)$ unifiziert werden können durch die Substitution $\{Sk_1/A\}$. Geben Sie ein Beispiel an, bei dem dies zu einer ungültigen Inferenz führt.

b) Schreiben Sie in **Prolog** die Quicksort-Sortieroutine der Form `quicksort(List, SortedList)`. Laden Sie sich hierzu GNU Prolog herunter (<http://www.gprolog.org/>).

Definieren Sie hierzu ein Prädikat `split(X, List, Small, Big)`, welches alle Elemente der Liste `List`, die kleiner oder gleich `X` sind, in die Liste `Small` hinzufügt, und alle größer `X` in die Liste `Big`. Nutzen Sie außerdem das bereits definierte Prädikat `append(L1, L2, L3)`, welches die Liste `L2` an `L1` anhängt und in `L3` schreibt.

Senden Sie Ihre kommentierte Implementierung sowie Ihre Ergebnisse an florian.mittag@uni-tuebingen.de mit dem Betreff „Abgabe KI-Uebung 12“. Bitte geben Sie in jeder Datei Ihren vollständigen Namen an, und senden Sie nur die Quellcode-Dateien ein, die Sie laut Aufgabenstellung modifiziert haben.