

## Softwaretechnik und Programmiersprachen I - Sommersemester 2011

### Übungsblatt 4

Besprechung der Aufgaben in 25.12.02.55 am Mittwoch 14:30-16:00 Uhr oder Freitag 10:30-12:00 Uhr  
Bei Fragen wenden Sie sich bitte an John Witulski: John.Witulski@uni-duesseldorf.de.

#### Aufgabe 4.1

Formen Sie die folgende Aussage so um, dass nur Existenzquantoren vorkommen

$$\forall n \cdot n \in \mathbb{N} \implies (\exists m \cdot m \in \mathbb{N} \wedge m > n)$$

#### Aufgabe 4.2

Beweisen Sie die Aussage:

$$(\forall x \cdot (\neg P(x) \implies Q(x)) \wedge \exists x \cdot \neg Q(x)) \implies \exists x \cdot P(x)$$

#### Aufgabe 4.3

Beantworten Sie die Fragen mit Hilfe von Prolog:

1. Siegfried liebt Krimhild und mag Gunther.
2. Krimhild liebt Siegfried und haßt Brunhild.
3. Gunther liebt Brunhild und mag Krimhild und Hagen.
4. Brunhild haßt Siegfried, Gunther und Krimhild.
5. Hagen haßt Siegfried und alle, die Siegfried lieben.
6. Brunhild mag alle, die Siegfried hassen.
7. Alberich haßt alle, mit Ausnahme von sich selbst.

Fragen:

1. Wen mag Brunhild?
2. Wer hasst Siegfried?
3. Welche Paare könnte man bilden? (Ein Paar soll aus zwei Personen bestehen, die sich gegenseitig lieben)

#### Aufgabe 4.4

Übersetzen Sie die folgenden Prädikate in Textform. Welche Prädikate sind Tautologien, Widersprüche, erfüllbar? Sie dürfen benutzen, was Sie über natürliche Zahlen ( $\mathbb{N}$ ) wissen.

1.  $\forall n \cdot n \in \mathbb{N} \implies (\exists m \cdot m \in \mathbb{N} \wedge m > n)$
2.  $\exists m \cdot m \in \mathbb{N} \wedge (\forall n \cdot n \in \mathbb{N} \implies m > n)$
3.  $\exists p \cdot p \in \mathbb{N} \wedge p \geq 2 \wedge (\forall q \cdot q \in \mathbb{N} \wedge q > p \implies \exists r \cdot r \in \mathbb{N} \wedge r < q \wedge r \geq 2 \wedge q \bmod r = 0)$

#### Aufgabe 4.5

Übersetzen Sie die folgenden Aussagen in prädikatenlogische Formeln

1. Es gibt für jeden Topf einen Deckel
2. Immer wenn es regnet, habe ich keinen Schirm dabei
3. "Grau, teurer Freund, ist alle Theorie und grün des Lebens goldner Baum."<sup>1</sup>

#### Aufgabe 4.6

Implementieren Sie einen Binärbaum. Als erstes Argument erhalten Sie stets einen sortierten Binärbaum.

1. Schreiben Sie zunächst ein Prädikat `bintree_find(Baum, El)`, das ein Element im Baum sucht. Beachten Sie dabei, dass Sie in einen Binärbaum suchen.
2. Schreiben Sie nun das Prädikat `bintree_add/3`, das ein neues Element in den Baum einfügt. Z.B.: `bintree_add(Baum, 2, BaumNeu)`.
3. Es soll auch möglich sein, Elemente aus den Baum zu löschen. Schreiben Sie dazu das Prädikat `bintree_del/3`. Z.B.: `bintree_del(Baum, 2, BaumNeu)`.

---

<sup>1</sup>Mephistopheles in Faust I (Johann Wolfgang von Goethe)