Übungsblatt 08

Übung

Abgabe bis Di., 11.06., 14 Uhr

Allgemein

Die Aufgaben müssen in **Dreiergruppen** abgegeben werden. Vierergruppen sind ebenfalls möglich.

Es ist **wichtig**, dass Sie sich an folgendes **Verfahren für die Abgabe** halten.
Die Lösungen werden in geeigneter Form in der Stud.IP-Veranstaltung Ihrer Übungsgruppe über das Vips-Modul hochgeladen. Sie müssen diese Abgaben nicht mit Markdown+AsciiMath erstellen. Sie können Ihre Bearbeitungen auch mit LaTeX formatieren, es ist aber auch die direkte Eingabe von Text oder der Upload von Text- und Bilddateien in gängigen Formaten möglich.

Weitere Hinweise zur Abgabe der Lösungen finden Sie in den Aufgabenstellungen.

Aufgabe 1 – 15 Punkte

Prädikatenlogik I

Betrachten Sie folgende prädikatenlogische Formel.

$$\forall X : \Big(P(X) \implies \Big(\forall S : \Big(\exists R : \Big(\Big(Q(f(R), T) \land Q(S, R)\Big) \lor O(T)\Big)\Big)\Big)\Big)$$

Gegeben ist eine Sprache über der Individuenmenge $D = [-10..10] \subset \mathbb{Z}$ (also den ganzen Zahlen von -10 bis 10 inklusive) mit folgender Signatur.

Die arithmetischen Operationen und die Größenrelationen über D sind analog zu jenen über den ganzen Zahlen $\mathbb{Z}$ und werden als bekannt vorausgesetzt.

- Prädikate $\mathcal{P} = \{O, P, Q\}$
  - O ist einstellig. $O(X)$ ist genau dann wahr, wenn $X = 9$.
  - P ist einstellig. $P(X)$ ist genau dann wahr, wenn $X \leq 0$.
  - Q ist zweistellig. $Q(X, Y)$ ist genau dann wahr, wenn $X \geq Y$.
- Funktoren $\mathcal{F} = \{f\}$
  - f ist einstellig.

$$f(X) = \begin{cases} 10 & \text{für } X = -10 \\ X - 1 & \text{sonst} \end{cases}$$

Bestimmen Sie die Menge $L(T) \subseteq D$ der Werte für T, mit denen die Formel den Wahrheitswert **wahr** annimmt.
(15 Punkte)

Aufgabe 2 – 15 Punkte

Prädikatenlogik II

Betrachten Sie folgende prädikatenlogische Formel.

$$\forall S : \Big(\exists R : \Big(\big(Q(T, f(R)) \vee Q(R, S)\big) \vee O(f(T))\Big)\Big)$$

Gegeben ist eine Sprache über der Individuenmenge

$$D = \{\texttt{Alice}, \texttt{Bob}, \texttt{Charlie}, \texttt{Diane}\}$$

mit folgender Signatur.

- Prädikate $\mathcal{P} = \{O, Q\}$
  - O ist einstellig. $O(X)$ ist genau dann wahr, wenn $X = \texttt{Charlie}$.
  - Q ist zweistellig. $Q(X, Y)$ ist genau dann wahr, wenn einer der folgenden Fälle eintritt.
    * $(X, Y) = (\texttt{Bob, Alice})$
    * $(X, Y) = (\texttt{Bob, Bob})$
    * $(X, Y) = (\texttt{Alice, Charlie})$
- Funktoren $\mathcal{F} = \{f\}$
  - f ist einstellig. Es gilt Folgendes.
    * $f(\texttt{Alice}) = \texttt{Bob}$
    * $f(\texttt{Bob}) = \texttt{Alice}$
    * $f(\texttt{Charlie}) = \texttt{Diane}$
    * $f(\texttt{Diane}) = \texttt{Diane}$

Bestimmen Sie die Menge $L(T) \subseteq D$ der Werte für T, mit denen die Formel den Wahrheitswert **wahr** annimmt.
(15 Punkte)

Praktische Übung

Abgabe der Prüfsumme bis Di., 11.06., 14 Uhr

Testat ab Di., 11.06. 18 Uhr

Hilfe zum Bearbeiten der praktischen Übungen können Sie grundsätzlich jeden Tag in den Rechnerübungen bekommen. Die Testate finden ebenfalls in **Dreiergruppen** und Vierergruppen statt. Dabei sind die Gruppen identisch zu denen, die auch die theoretischen Aufgaben zusammen bearbeiten. In diesem Fall reserviert nur ein Gruppenmitglied einen Termin. Es ist ausreichend, wenn nur **eine** Person aus der Gruppe eine Prüfsumme abgibt.

Abgabe der Prüfsumme

- Siehe vorherige Übungen.
- Übermitteln Sie die Prüfsumme mit dem Test *GdPI 08 - Testat*.

Vorbereitung

Für eine aussagenlogische Formel

$$F = (L_{11} \vee L_{12} \vee \ldots \vee L_{1\,m_1}) \wedge (L_{21} \vee L_{22} \vee \ldots \vee L_{2\,m_2}) \wedge \ldots \wedge (L_{n1} \vee L_{n2} \vee \ldots \vee L_{n\,m_n})$$

in konjunktiver Normalform (KNF) mit Literalen L_{ij} sind die Mengen der Literale der einzelnen Disjunktionen die **Klausel** von F.

$$\begin{aligned} K_1 &= \{L_{11}, L_{12}, \ldots, L_{1\,m_1}\} \\ K_2 &= \{L_{21}, L_{22}, \ldots, L_{2\,m_2}\} \\ &\ldots \\ K_n &= \{L_{n1}, L_{n2}, \ldots, L_{n\,m_n}\} \end{aligned}$$

Die **Klauselmenge** von F ist die Menge aller Klauseln $\{K_1, K_2, \ldots, K_n\}$.

Durch Umkehrung diese Ansatzes kann aus einer Klauselmenge eine Formel in KNF mit dieser Klauselmenge konstruiert werden.

Bemerkung

Verschiedene Formeln in KNF können dieselbe Klauselmenge haben, aber aufgrund von Idempotenz und Kommutativität sind Formel mit derselben Klauselmenge äquivalent.

Aufgabe 1 Python – 30 Punkte

Aussagenlogik in Python

Zur Darstellung eines Literal, d.h. einer atomaren Formel oder der Negation einer atomaren Formel, wird ein Tupel von String (`str`) und Boolean (`bool`) benutzt. Der String ist die atomare Formel, der Boolean entspricht dem Wahrheitswert des Literals, wenn die atomare Formale mit `True` belegt wird.

Beispiel

Tupel für die Literale $\neg A$ und B.

```
('A', False)
('B', True)
```

1. Definiere Sie die Funktion `showLit` in Python, die Stringrepräsentationen von Literalen wie im folgenden Beispiel erzeugen.

   ```
   >>> print(showLit(('A', False)) + "\n" + showLit(('B', True)))
   ~A
   B
   ```

   (5 Punkte)

2. Eine Klausel ist eine Liste von Literalen. Eine Formel in KNF wird von der Klauselmenge repräsentiert und als Liste von Klauseln dargestellt.

   Beispiel

   Formel (KNF) $(\neg B) \wedge (A \vee B \vee \neg C)$

   - Klauseln $\{\neg B\}$, $\{A, B, \neg C\}$,

     ```
     [('B', False)]
     [('A', True), ('B', True), ('C', False)]
     ```

   - Klauselmenge $\{\{\neg B\}, \{A, B, \neg C\}\}$

     ```
     [[('B', False)], [('A', True), ('B', True), ('C', False)]]
     ```

   Definiere Sie Funktionen `showClause` und `showCNF` in Python, die Stringrepräsentationen von Klauseln und Formeln in KNF/Klauselmenge wie im folgenden Beispiel erzeugen.

   ```
   >>> a=[('B', False)]
   >>> b=[('A', True), ('B', True), ('C', False)]
   >>> c=[a, b]
   >>> print(showClause(a) + "\n" + showClause(b))
   ~B
   A v B v ~C
   >>> print(showCNF(c))
   (~B) ^ (A v B v ~C)
   ```

   (5 Punkte)

3. Ein Belegung der Literale/Klauseln/Formel ist eine Liste von Strings. Für eine Belegung wird die atomare Formel eines Literals, dessen String in der Liste enthalten ist, mit `True` belegt; die atomare Formel eines Literals, dessen String nicht in der Liste enthalten ist, mit `False`.

   Definieren Sie die Funktionen

   ```
   alphaLit(lit, alpha)
   alphaClause(clause, alpha)
   alphaCNF(cnf, alpha)
   ```

   in Python, die ein Literal/eine Klausel/eine Formel und eine Belegung übergeben bekommen und einen Wahrheitswert (`bool`) zurückliefern.

   Beispiel

   Sei `a`, `b` und `c` wie in Aufgabenteil 2 definiert.

   ```
   >>> alpha=['A','B']
   >>> alphaLit(('A', True), alpha)
   True
   >>> alphaLit(('B', False), alpha)
   False
   >>> alphaLit(('C', True), alpha)
   False
   >>> alphaClause(a, alpha)
   False
   >>> alphaClause(b, alpha)
   True
   >>> alphaCNF(c, alpha)
   False
   ```

   (10 Punkte)

4. Bestimmen Sie eine erfüllende und eine nicht erfüllende Belegung für die in Aufgabenteil 2 definierte Formel `c`.
   (4 Punkte)

5. Die Formel

$$F = (X \vee \neg Y) \wedge (\neg X \vee \neg Y \vee \neg Z) \wedge (\neg X \vee Z) \wedge (X \vee Y \vee Z) \wedge (Y \vee \neg Z)$$

aus Übung 07, Aufgabe 2 ist nicht erfüllbar. Beweisen Sie das, indem Sie alle möglichen Belegungen für die Literale von F testen.

Beschränken Sie sich, mit Ausnahme des Codes aus den vorstehenden Aufgabenteilen, auf höchstens 4 Codezeilen in Python.
(6 Punkte)