Übungsblatt 08

**Der Umfang dieses Übungsblatts ist reduziert,
damit mehr Zeit für die Bearbeitung der Praktischen Übung bleibt.**

Übung

Abgabe bis Di., 20.06., 8 Uhr

Allgemein

Ihre Lösungen werden nur korrigiert, wenn Sie einer Übungsgruppe angehören, d.h. Teilnehmer einer Stud.IP-Veranstaltung *GdPI - Übung - (⟨Termin⟩)* sind.

Nur wer in Ilias etwas abgibt kann Punkte bekommen. Siehe dazu die Anmerkungen auf Übungsblatt 02.

Benutzen Sie Markdown mit AsciiMath um die Lösung der Aufgaben zu strukturieren und zu kodieren. Fassen Sie die **Lösungen aller Aufgaben in einer Datei** zusammen.

Formatieren Sie Ihre Abgabe so, dass **die einzelnen Aufgaben klar durch passende Überschriften voneinander** getrennt sind.

Geben Sie die Datei über das Lernmodul *GdPI 08 - Markdown+AsciiMath* ab.

Aufgabe 1 – 15 Punkte

Prädikatenlogik-1 (gdpi-l-k4mja0ntaz)

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. $f(X) = X - 1$.

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-2 (gdpi-l-y3mtu0mtqy)

Betrachten Sie folgende prädikatenlogische Formel.

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

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., 20.06., 8 Uhr

Testat ab Fr., 23.06.

Hilfe zum Bearbeiten der praktischen Übungen können Sie grundsätzlich jeden Tag in den Rechnerübungen bekommen.

Am **Mo., 19.06.**, 10-12 Uhr online (3 Tutor*innen) und 18-20 Uhr in Präsenz (2 Tutor*innen), finden **keine Testate** statt. Diese Rechnerübungen sind ausschließlich **für Fragen** reserviert.

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 – 25 Punkte

Aussagenlogik

Zur Darstellung eines Literal, d.h. einer atomare Formel oder der Negation einer atomaren Formel, wird ein Tupel von String (`str`) und Boolean (`bool`) benutzt. Der String ist die atomaren 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)
```

Um eine kompakte Stringrepräsentation für Literale zu erhalten kann dir Funktion `showLit` verwendet werden.

```
>>> def showLit(lit):
...     s=""
...     if not lit[1]:
...         s+="~"
...     s+=lit[0]
...     return s
...
>>> print(showLit(('A', False)) + "\n" + showLit(('B', True)))
~A
B
```

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)]]
  ```

1. Definiere Sie Funktion `showClause` und `showCNF`, die Stringrepräsentation von Klauseln und Formel 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)

2. 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 Formeln 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)
```

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 1 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)

3. Bestimmen Sie eine erfüllende und eine nicht erfüllende Belegung für die in Aufgabenteil 1 definierte Formel `c`.
(4 Punkte)

4. 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
(6 Punkte)