Humboldt-Universität zu Berlin
Lehrstuhl Logik in der Informatik
Prof. Dr. Nicole Schweikardt

10. November 2023

Logik in der Informatik

Wintersemester 2023/2024

Übungsblatt 4

Abgabe: *bis 20. November 2023, 13.00 Uhr*

Aufgabe 1: **(Moodle-Quiz)**

Absolvieren Sie das Quiz 4 auf der Moodle-Plattform.

Aufgabe 2: **(Präsenzaufgabe)**

Auch im 50. Jahrtausend ist *Contra-Bo* für die Server der inzwischen virtuellen Realitätsbetreiber ***B**luesky*, ***D**iaspora*, ***M**astodon* und ***T**hreads* verantwortlich. Kürzlich konnte *Contra-Bo* ein unendlich großes Rechenzentrum mit Server $\langle i, j\rangle$ für alle $i, j \in \mathbb{Z}$ erwerben. Entsprechend existieren nun auch unendlich viele Aussagensymbole $B_{i,j}, D_{i,j}, M_{i,j}, T_{i,j}$ für alle $i, j \in \mathbb{Z}$.

(a) Stellen Sie unendliche Formelmengen Φ_1 und Φ_2 auf, die die in Aufgabe 2 **(a)** und **(b)** von Blatt 2 beschriebenen Bedingungen repräsentieren, allerdings für das neue Rechenzentrum.

(b) Warum kann die Bedingung aus Aufgabe 2 **(d)** von Blatt 2 nicht durch eine unendliche Formelmenge über den gegebenen Aussagensymbolen repräsentiert werden?

(c) Um überall Server für alle Kunden in der Nähe zu haben, soll auf jedem drei mal drei Server großen Abschnitt jeder Kunde durch mindestens einen Server vertreten sein. Stellen Sie eine Formelmenge Φ_3 auf, die diese Bedingung repräsentiert.

(d) Einige Serverkennungen haben inzwischen sentimentalen Wert bei den Kunden und sollen deswegen auch im neuen Rechenzentrum für den entsprechenden Kunden reserviert werden. Sei $\Phi := \Phi_1 \cup \Phi_2 \cup \Phi_3$ und sei Ψ die Formelmenge

$$\begin{aligned}\Psi \;:=\; & \Big\{B_{i,j} : \text{Bluesky möchte Server } \langle i,j\rangle \text{ reservieren}\Big\} \\ \cup\; & \Big\{D_{i,j} : \text{Diaspora möchte Server } \langle i,j\rangle \text{ reservieren}\Big\} \\ \cup\; & \Big\{M_{i,j} : \text{Mastodon möchte Server } \langle i,j\rangle \text{ reservieren}\Big\} \\ \cup\; & \Big\{T_{i,j} : \text{Threads möchte Server } \langle i,j\rangle \text{ reservieren}\Big\}\end{aligned}$$

Nun stellt sich für die *Contra-Bo* die Frage, ob es einen Nutzungsplan gibt, der die zusätzlichen Vorlieben der Kunde respektiert, d.h. ob $\Phi \cup \Psi$ erfüllbar ist.

Zeigen Sie: Es gibt genau dann einen korrekten Nutzungsplan für das gesamte Rechenzentrum, wenn es für jeden endlichen quadratischen Abschnitt einen solchen Nutzungsplan gibt.

Aufgabe 3: (40 Punkte)

(a) Wandeln Sie analog zu Beispiel 2.52 die Formel

$$\varphi \ := \ \big(\neg(P \vee Q) \rightarrow (\neg S \wedge T)\big)$$

mit dem Tseitin-Verfahren in eine erfüllbarkeitsäquivalente Formel φ_K in 3-KNF um.
Achtung: Halten Sie sich strikt an die Art der Notation und Zeilenaufteilung von Beispiel 2.52. Dies beinhaltet folgende Eigenschaften:

- Die Subformeln von φ erhalten aufsteigend Bezeichner der Form ψ_i, wobei $i \in \mathbb{N}_{\geq 1}$ (φ erhält keinen zusätzlichen Bezeichner).
- Negierte Aussagensymbole und φ bilden hier keine eigene Subformel. Im Gegensatz dazu bilden aber negierte Formeln, die aus mehr als nur einem Aussagensymbol bestehen, eine eigene Subformel.
- Der Index i entspricht der Position der Subformeln in der Traversierung des Syntaxbaumes von φ nach der pre-order Tiefensuche. Eine solche Traversierung listet zunächst die Wurzel des Baumes auf, dann die Traversierung des kompletten linken Teilbaumes (so dieser existiert) und dann die des rechten Teilbaumes (so dieser existiert).
- Die neuen Aussagensymbole sind entsprechend aus der Menge $\{X_\varphi, X_{\psi_1}, X_{\psi_2}, \dots\}$ zu wählen. Für jede Subformel wird in φ' eine neue Zeile begonnen und rechtsseitig die passende Begründung angegeben.
- In φ_K entspricht die Zeilenaufteilung der Zeilenaufteilung von φ'.

Lösungen, die sich nicht an obige Formregeln halten, werden nicht oder nur teilweise korrigiert. Bei fehlerhaften Zeilen in φ' können eventuell die entsprechenden Zeilen in φ_K nicht korrigiert werden.

(b) Sei $n \in \mathbb{N}$ mit $n \geq 1$ und sei φ_n die in Satz 2.45 der Vorlesung betrachtete aussagenlogische Formel.

(i) Bestimmen Sie alle Interpretationen $\mathcal{I}$, für die gilt:

- $\mathcal{I}$ erfüllt φ_n und
- für jedes $i \in \{1, \dots, n\}$ existiert eine Interpretation, die sich von $\mathcal{I}$ nur dadurch unterscheidet, dass sie *genau* eines der beiden Aussagensymbole X_i, Y_i auf einen anderen Wahrheitswert abbildet als $\mathcal{I}$, und die φ_n *nicht* erfüllt.

(ii) Beweisen Sie Satz 2.45 der Vorlesung.
Hinweis: Eine Möglichkeit, dies zu zeigen, ist einen Beweis durch Widerspruch zu führen. Nehmen Sie dafür an, dass ψ_n eine zu φ_n äquivalente Formel in DNF ist, die aus weniger als 2^n konjunktiven Klauseln besteht. D.h. es gibt eine natürliche Zahl $N < 2^n$ und N konjunktive Klauseln $\kappa_1, \dots, \kappa_N$, so dass $\psi_n = \kappa_1 \vee \dots \vee \kappa_N$. Folgern Sie aus Aufgabenteil **(i)**, dass mindestens eine der Klauseln $\kappa_1, \dots, \kappa_N$ von mindestens zwei essentiell verschiedenen Interpretationen $\mathcal{I}$ aus **(i)** erfüllt wird. Leiten Sie daraus einen Widerspruch her.

(iii) Gibt es für alle $n \in \mathbb{N}$ mit $n \geq 1$ DNF-Formeln φ'_n der Länge $\mathcal{O}(n)$, so dass jede zu φ'_n äquivalente KNF-Formel mindestens 2^n disjunktive Klauseln hat? Beweisen Sie, dass Ihre Antwort korrekt ist.

Aufgabe 4: **(20 Punkte)**

Lesen Sie Kapitel 4 aus dem Buch “Learn Prolog Now!”.

Achtung: *Fertigen Sie Ihre Lösung für die Aufgabenteile (a) und (b) handschriftlich an (also wie Aufgabe 3) und reichen Sie diese bei Moodle ein. Die Lösung des Aufgabenteils (c) muss unter Beachtung der bekannten Abgabehinweise (siehe* `http://hu.berlin/prolog`*) für Prolog-Code in einem zusätzlichen Abgabefach in Moodle eingereicht werden!*

(a) Wie antwortet Prolog auf die folgenden Anfragen?

- **(i)** `?- [ a, b ] = [ X, Y ].`
- **(ii)** `?- [ X | []] = [c].`
- **(iii)** `?- [ [] | [ b, c ]] = [ X, _, Z ].`
- **(iv)** `?- [ H | T ] = [ a, b | [ c | [ d ]]].`

(b) Das Prädikat `member/2` wird in Abschnitt 4.2 des Buchs “Learn Prolog Now!” definiert. Zeichnen Sie den Suchbaum für die Anfrage

```
?- member(42, [42, ixs, X]).
```

(c) Definieren Sie *rekursiv* ein Prädikat `nimm/3`, so dass `nimm(E, X, Y)` genau dann erfolgreich ist, wenn `E` ein Element der Liste `X` ist und `Y` aus der Liste `X` durch Löschung eines Vorkommens von `E` entsteht. So sollte beispielsweise die Anfrage

```
?- nimm(E, [1, 2, 3], Y).
```

zu der folgenden Antworten führen:

```
E = 1,
Y = [2, 3] ;
E = 2,
Y = [1, 3] ;
E = 3,
Y = [1, 2] ;
false.
```