Humboldt-Universität zu Berlin
Lehrstuhl Logik in der Informatik
Prof. Dr. Nicole Schweikardt 10. November 2015

Logik in der Informatik

Wintersemester 2015/2016

Übungsblatt 4

Abgabe: *bis 19. Nov. 2015, 13.[15] Uhr (vor der Vorlesung oder im Briefkasten zwischen den Räumen 3.401 und 3.402 im Johann von Neumann-Haus (Rudower Chaussee 25))*

Aufgabe 1: (40 Punkte)

(a) Stellen Sie für die Klauselmenge

$$\Gamma_1 \ := \ \{ \ \{Q,S\},\ \{\neg Q,\neg S\},\ \{S,\neg R\},\ \{\neg S,R\},\ \{\neg Q,\neg R\} \ \}$$

eine aussagenlogische Formel φ_1 in KNF auf, so dass für jede Interpretation $\mathcal{I}$ gilt:

$$\mathcal{I} \models \varphi_1 \quad \Longleftrightarrow \quad \mathcal{I} \models \Gamma_1 \ .$$

(b) Sei Γ_1 die Klauselmenge aus Aufgabenteil (a) und sei

$$\Gamma_2 \ := \{ \ \{P,\neg Q,R\},\ \{P,\neg Q,\neg R\},\ \{\neg P,\neg Q\},\ \{Q,S\},\ \{Q,\neg S\} \ \} \ ,$$
$$\Gamma_3 \ := \{ \ \{P,Q,R,\neg S\},\ \{\neg P,\neg S\},\ \{Q,\neg R\},\ \{\neg Q,R,S\},\ \{\neg Q,\neg S\},\ \{S\} \ \} \ ,$$

wobei P,Q,R,S unterschiedliche Aussagensymbole aus AS sind. Geben Sie für jede der drei Klauselmengen jeweils ein Modell oder eine Resolutionswiderlegung an. Bei einer Resolutionswiderlegung gehen Sie analog zu Beispiel 2.59 vor und wählen entweder die graphische Darstellung oder die Resolutionswiderlegung als Auflistung mit rechtsseitigen Begründungen.

(c) Seien Q,R,S,T,U,W unterschiedliche Aussagensymbole aus AS. Wenden Sie den DPLL-Algorithmus auf die folgende Klauselmenge Γ an. Erklären Sie dabei Schritt für Schritt, wie der Algorithmus vorgeht.

$$\Gamma \ := \Big\{ \ \{\neg R,T,W\},\ \{\neg R,\neg S,\neg W\},\ \{\neg R,\neg T\},\ \{\neg Q,S,T\},\ \{\neg Q,R,\neg S\},$$
$$\{R,S,W\},\ \{R,\neg T,\neg W\},\ \{Q,U\},\ \{S,\neg U,\neg W\},\ \{Q,W\},\ \{Q,\neg S,\neg U\} \ \Big\}$$

Hinweis: Um Ihnen selbst und unseren Korrektoren die Arbeit zu erleichtern, wählen Sie bitte in Zeile 4 des DPLL-Algorithmus nicht-negierte Literale, und zwar in alphabetischer Reihenfolge. Ebenso wählen Sie bitte bei der Anwendung der Vereinfachungsheuristiken die Literale in alphabetischer Reihenfolge.

Geben Sie wie in Beispiel 2.64 entsprechend die entstehende Klauselmenge und die benutzte Vereinfachungsheuristik an.

Aufgabe 2: (15 Punkte)

Betrachten Sie die Einschränkung des aussagenlogischen Erfüllbarkeitsproblems auf Formeln in DNF, d.h.: Die Eingabe besteht aus einer aussagenlogischen Formel φ in DNF, und die Aufgabe ist, zu entscheiden, ob φ erfüllbar ist.

Finden Sie heraus, ob dieses Problem effizient gelöst werden kann. Falls „ja“, geben Sie einen Polynomialzeit-Algorithmus zur Lösung des Problems an; falls „nein“, weisen Sie nach, dass das Problem NP-vollständig ist.

Aufgabe 3: **(20 Punkte)**

Im Folgenden betrachten wir einen *Baum* $\mathcal{B}$ mit der *abzählbar unendlichen Knotenmenge* $V := \mathbb{N}$. Die Wurzel von $\mathcal{B}$ ist dabei der Knoten $w := 0$. Die Kanten von $\mathcal{B}$ repräsentieren wir durch eine Funktion *Kinder*, die jedem Knoten $v \in V$ die Menge $\mathit{Kinder}(v)$ all seiner Kinder zuordnet. Wir nehmen an, dass $\mathcal{B}$ *endlich verzweigend* ist. Damit meinen wir, dass für jeden Knoten $v \in V$ die Menge $\mathit{Kinder}(v)$ endlich ist.

(a) Ein *Pfad* in $\mathcal{B}$ ist eine (endliche oder unendliche) Folge $(v_0, v_1, v_2, \ldots)$ von Knoten aus V, so dass gilt: $v_0 = w$ ist die Wurzel von $\mathcal{B}$, und für alle v_i, v_{i+1} auf dem Pfad ist $v_{i+1} \in \mathit{Kinder}(v_i)$.
Eine Interpretation $\mathcal{I} \colon \mathsf{AS} \to \{0, 1\}$ *repräsentiert einen Pfad* $P = (v_0, v_1, v_2, \ldots)$, falls für jedes $v \in V$ und das zugehörige Aussagensymbol $A_v \in \mathsf{AS}$ gilt:

$$\mathcal{I}(A_v) = 1 \quad \iff \quad v \in \{v_0, v_1, v_2, \ldots\}.$$

Das Aussagensymbol A_v repräsentiert also die Aussage *„Der Knoten v gehört zum Pfad P"*.
Geben Sie eine unendliche Formelmenge Φ an, so dass für jede Interpretation $\mathcal{I}$ gilt:

$$\mathcal{I} \models \Phi \quad \iff \quad \mathcal{I} \text{ repräsentiert einen Pfad unendlicher Länge in } \mathcal{B}.$$

(b) Ein endlicher Pfad $(v_0, v_1, v_2, \ldots, v_n)$ hat die Länge n. Wir sagen dass der Baum $\mathcal{B}$ *Pfade beliebiger endlicher Länge* enthält, wenn $\mathcal{B}$ für jedes $n \in \mathbb{N}$ einen Pfad der Länge n enthält.
Beweisen Sie das folgende Lemma von Dénes Kőnig (1936):

Kőnigs Lemma. *Wenn $\mathcal{B}$ Pfade beliebiger endlicher Länge enthält, dann enthält $\mathcal{B}$ einen Pfad unendlicher Länge.*

Hinweis: Benutzen Sie den *Endlichkeitssatz*.

Aufgabe 4: **(25 Punkte)**

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

Achtung: *Die Bearbeitung der Teilaufgaben (c) und (d) ist über das GOYA-System abzugeben! Lösungsansätze, die von SWI-Prolog nicht geladen werden können, werden nicht bewertet!*

(a) Wie antwortet Prolog auf die folgenden Anfragen?

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

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

(d) Wir kodieren aussagenlogische Literale wie folgt durch Prolog-Terme: Ist $i \in \mathbb{N}$, dann repräsentiert `pos(`i`)` das Literal A_i und `neg(`i`)` das Literal $\neg A_i$. Weiterhin kodieren wir Mengen von Literalen als Prolog-Listen. Beispielsweise repräsentieren wir $\{A_1, \neg A_2, \neg A_3\}$ durch `[pos(1), neg(2), neg(3)]`.
Schreiben Sie ein Prädikat `resolvente/3`, so dass Folgendes gilt: Unter der Annahme, dass `L1`, `L2` und `R` Mengen von Literalen repräsentieren, ist `resolvente(L1, L2, R)` erfüllt wenn `R` eine Resolvente von `L1` und `L2` ist. Beispielsweise sollte die Anfrage

```
?- resolvente([pos(1), neg(3), pos(4)], [pos(2), pos(3), neg(4)], R).
```

zu folgenden Ausgaben führen:

`R = [pos(1),pos(4),pos(2),neg(4)]` und `R = [pos(1),neg(3),pos(2),pos(3)]`

Hinweise: Nutzen Sie gegebenenfalls das Prädikat `nimm/3` aus Teilaufgabe **(c)**. Haben Sie **(c)** nicht gelöst, so können Sie die Online-Hilfe von SWI-Prolog nutzen, um sich mit dem vordefinierten Prädikat `select/3` vertraut zu machen. Nutzen Sie außerdem das vordefinierte Prädikat `union/3`.