Karlsruher Institut für Technologie

Klausur Formale Systeme

Fakultät für Informatik

WS 2013/2014

Prof. Dr. Peter H. Schmitt

11. April 2014

Vorname: **vorname**
Name: **nachname**
Matrikel-Nr.: **matrikelnr**

Platz-Nr.: **platz**
Code: **nonce**

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (10)	A2 (6)	A3 (7)	A4 (6)	A5 (5)	A6 (13)	A7 (6)	A8 (7)	Σ (60)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung (5+5 Punkte)

a. Kreuzen Sie in der folgenden Tabelle alles Zutreffende an.

Für jede korrekte Antwort gibt es einen Punkt, **für jede falsche Antwort wird ein *halber Punkt* abgezogen!** (Dabei werden jedoch keinesfalls weniger als 0 Punkte für jede der zwei Teilaufgaben vergeben.)

Hinweise:

- „PL1" steht für „Prädikatenlogik erster Stufe (mit Gleichheit $\doteq$)", wie sie in der Vorlesung vorgestellt wurde. Auf diese beziehen sich in Teilaufgabe a. auch die Begriffe „erfüllbar", „allgemeingültig" und „unerfüllbar".
- p, r und s sind Prädikatensymbole, g ist ein Funktionssymbol, c ist ein Konstantensymbol, x und y sind Variablen.
- Es gelten die üblichen Klammereinsparungsregeln.

	keine Formel der PL1	allgemeingültig	erfüllbar, aber nicht allgemeingültig	unerfüllbar
$(\neg(r \leftrightarrow s)) \leftrightarrow (\neg r \leftrightarrow s)$				
$\forall x(p(x) \rightarrow p(p(x))) \rightarrow p(p(c))$				
$\neg\exists x\exists y(g(x) \doteq g(y))$				
$\big(\exists x\exists y\ p(x,y)\big) \rightarrow \big(\exists x\ p(x,g(x))\big)$				
$\big(\forall x\ p(x,g(x))\big) \rightarrow \big(\exists y\ p(y,g(g(c)))\big)$				

b. Bitte kreuzen Sie in der folgenden Tabelle das Zutreffende an. Für korrekte Antworten erhalten Sie einen Punkt, **für falsche Antworten wird *ein Punkt* abgezogen**. Dabei werden jedoch nie weniger als 0 Punkte für diese Teilaufgabe vergeben.

	Richtig	Falsch
Allgemeingültigkeit aussagenlogischer Formeln in KNF ist in linearer Zeit entscheidbar.		
Sei M eine Menge von PL1-Formeln und F eine PL1-Formel (jeweils ohne freie Variablen). $M \models F$ gilt genau dann, wenn $M \cup \{\neg F\}$ unerfüllbar ist.		
Für ein aussagenlogisches Atom p ist die LTL-Formel $(\lozenge p) \wedge (\lozenge \neg p)$ unerfüllbar.		
Die PL1-Variable x ist mit jedem PL1-Term t unifizierbar.		
Das Erfüllbarkeitsproblem aussagenlogischer 3-KNF-Formeln ist NP-vollständig.		

2 Beweisaufgabe (6 Punkte)

Definition. Eine unerfüllbare Klauselmenge S heißt *minimal unerfüllbar*, wenn jede echte Teilemenge $S_0 \subset S$ erfüllbar ist.

Sei S eine minimal unerfüllbare aussagenlogische Klauselmenge.

Zeigen oder widerlegen Sie: Für jede Klausel $C \in S$ und jedes Literal L in C gibt es eine Klausel $D \in S$, so dass $\bar{L}$ in D vorkommt.

3 Kurze konjunktive Normalform (7 Punkte)

Geben Sie für die Formel

$$(\neg(A \leftrightarrow B)) \to (A \to B)$$

eine erfüllbarkeitsäquivalente Formel in **kurzer konjunktiver Normalform** an. Machen Sie dabei Ihre Vorgehensweise deutlich, sowie, welche eingeführte Variable welcher Teilformel entspricht.

4 Formalisieren in Prädikatenlogik (1+1+2+2=6 Punkte)

Formalisieren Sie die vier folgenden Aussagen über die Liebe in Prädikatenlogik erster Stufe mit Gleichheit. Benutzen Sie dafür jeweils die folgenden interpretierten Symbole:

- Prädikatensymbol (zweistellig) $L(\cdot,\cdot)$ mit der Semantik $(x, y) \in \mathcal{I}(L)$ gdw. x liebt y.

Sie können davon ausgehen, dass alle Formeln mit der Menge der Menschen als Universum interpretiert werden. Weitere Annahmen über die Interpretation $\mathcal{I}$ als die obigen dürfen Sie nicht machen.

a. Jeder Mensch liebt sich selbst.

b. Kein Mensch liebt alle Menschen.

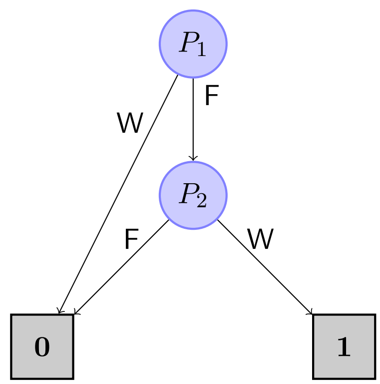
c. Jeder Mensch, der sich selbst liebt, wird von keinem anderen geliebt.

d. Jeder Mensch liebt höchstens einen anderen Menschen.

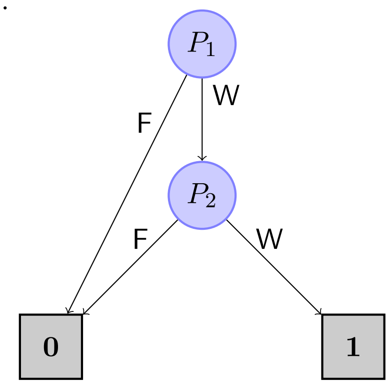
5 Shannongraphen (3+2 Punkte)

Gegeben seien die folgenden beiden Shannnongraphen bezüglich der Variablen-Ordnung $P_1 < P_2$.

G_1 :

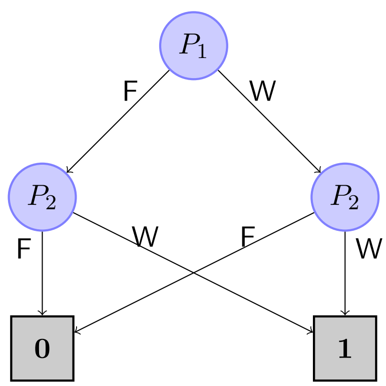


G_2 :



a. Begründen Sie, warum folgender Shannnongraph die Disjunktion von G_1 und G_2 ist.

G_3 :



Shannongraphen (Fortführung)

b. Konstruieren Sie zu dem Shannnongraphen G_3 aus Teilaufgabe (a) den reduzierten Shannongraphen (mit der gleichen Variablen-Ordnung $P_1 < P_2$). Verwenden Sie das Verfahren aus der Vorlesung. Geben Sie alle Zwischenschritte an, d. h. geben Sie nach jedem Reduktionsschritt den daraus resultierenden Graphen an. Es gibt für jeden korrekten Zwischenschritt Punkte, d. h. insbesondere, die volle Punktzahl kann nur erreicht werden, wenn alle Zwischenschritte korrekt angegeben sind.

6 Resolution (13 Punkte)

Beweisen Sie mit Hilfe des Resolutionskalküls für die PL1, dass die unten stehende Menge prädikatenlogischer Klauseln unerfüllbar ist. Darin ist p ein zweistelliges Prädikatensymbol, f und g sind zweistellige Funktionssymbole, a und b sind Konstantensymbole und x, y, z sind Variablen.

Machen Sie bei jedem Resolutionsschritt erkenntlich, aus **welchen Ausgangsklauseln** die Resolvente entsteht und **welche Substitution** Sie dazu verwendet haben.

$$\Big\{ \underset{(1)}{\{\neg p(x,y)\,,\, p(f(x,a),y)\}} \;,\; \underset{(2)}{\{\neg p(f(x,y),z)\,,\, p(x,f(y,z))\}} \;,\; \underset{(3)}{\{\neg p(x,f(y,z))\,,\, p(z,f(x,y))\}} \;,$$
$$\underset{(4)}{\{p(g(b,x),a)\}} \;,\; \underset{(5)}{\{\neg p(a,f(a,g(x,a)))\}} \Big\}$$

7 Spezifikation mit der Java Modeling Language (2+1+3 Punkte)

Gegeben ist die Klasse

```
public class ArrayList {
    private Object[] arr;
    private int size;
    //@ invariant 0 <= size && size <= arr.length;

    /*@ public normal_behaviour
      @
      @
      @
      @   (1)
      @
      @
      @
      @   (2)
      @
      @
      @
      @   (3)
      @
      @
      @*/
    public insertAt(int pos, Object o) {
        // ...
    }

    // ...
}
```

Diese Klasse modelliert eine Liste von Referenzen vom Type *Object*, die in einem Array gespeichert werden. Die Länge der Liste steht im Feld `size`.

Die Methode `insertAt` hat die folgende informelle Spezifikation:

(1) Die Methode macht keine Garantien, wenn der Parameter `pos` einen negativen Wert oder einen Wert (echt) größer als die Länge der Liste annimmt.

(2) Nach Ausführen der Methode ist die Länge der Liste um eins größer als vor der Methode.

(3) Alle Einträge im Array hinter `pos` sind gegenüber dem Vorzustand um eins nach hinten geschoben.

Formalisieren Sie diesen Methodenvertrag von `insertAt` in JML und tragen Sie die Formalisierung der Elemente (1), (2) und (3) oben ein.

8 LTL und Büchi-Automaten (3+4 Punkte)

Gegeben sei die Signatur

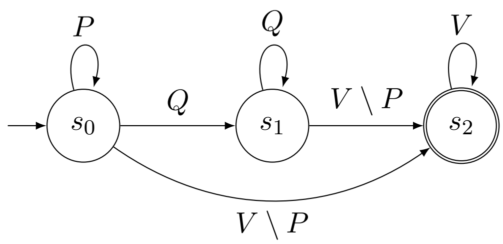
$$\Sigma = \{p, q\} \text{ und das zugehörige Alphabet } V = \mathbb{P}(\Sigma) = \{\emptyset, \{p\}, \{q\}, \{p, q\}\} \ .$$

Es werden die aus der Vorlesung bekannten Abkürzungen verwendet:

$$P = \{\{p\}, \{p, q\}\}, \quad Q = \{\{q\}, \{p, q\}\}, \quad PQ = \{\{p, q\}\}$$

a. Geben Sie eine LTL-Formel A_0 über der Signatur Σ an, welche genau in den *omega-Strukturen* wahr ist, die der folgende Büchi-Automat $\mathcal{A}_0$ über dem Alphabet V akzeptiert (d.h., so dass $L^\omega(\mathcal{A}_0) = \{\xi \in V^\omega : \xi \models A_0\}$ gilt).

$\mathcal{A}_0$:



A_0 :

b. Geben Sie einen Büchi-Automaten $\mathcal{A}_1$ über dem Alphabet V an, der genau die LTL-Formel

$$(\Box p) \ \mathbf{V} \ q$$

akzeptiert, d.h., so dass $L^\omega(\mathcal{A}_1) = \{\xi \in V^\omega : \xi \models (\Box p) \, \mathbf{V} \, q\}$ gilt.

$\mathcal{A}_1$: