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
$(\neg(r \leftrightarrow s)) \leftrightarrow (\neg r \leftrightarrow s)$		X		
$\forall x(p(x) \to p(p(x))) \to p(p(c))$	X			
$\neg\exists x\exists y(g(x) \doteq g(y))$				X
$\big(\exists x\exists y\ p(x,y)\big) \to \big(\exists x\ p(x,g(x))\big)$			X	
$\big(\forall x\ p(x,g(x))\big) \to \big(\exists y\ p(y,g(g(c)))\big)$		X		

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.	X	
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.	X	
Für ein aussagenlogisches Atom p ist die LTL-Formel $(\Diamond p) \wedge (\Diamond \neg p)$ unerfüllbar.		X
Die PL1-Variable x ist mit jedem PL1-Term t unifizierbar.		X
Das Erfüllbarkeitsproblem aussagenlogischer 3-KNF-Formeln ist NP-vollständig.	X	

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.

Widerspruchsbeweis!
Angenommen es gibt eine Klausel $C \in S$ und ein Literal L in C, so daß kein $D \in S$ existiert, in dem $\bar{L}$ vorkommt.
Sei C eine beliebige Klausel in S (es gibt eine, da S unerfüllbar ist).

Wegen der minimalen Unerfüllbarkeit ist $S \setminus \{C\}$ erfüllbar. Sei I eine erfüllende Interpretation für $S \setminus \{C\}$. Wir nehmen an, es gibt nun ein Literal L in C, so dass $\bar{L}$ in keiner Klausel in $S \setminus \{C\}$ vorkommt.

O.b.d.A. sei L positiv, für negative L ist der Beweis analog.

Wir bilden die neue Interpretation J:

$$J(K) = \begin{cases} I(K) & \text{falls } K \neq L \\ \mathbf{W} & \text{für } K = L \end{cases}$$

Da L disjunktiv in C vorkommt, gilt $J(C) = \mathbf{W}$.

Für jede Klausel D in S mit $D \neq C$ galt $I(D) = \mathbf{W}$. Da höchstens L, und nicht $\bar{L}$, in D vorkommen kann, gilt auch noch $J(D) = \mathbf{W}$. Somit ist J eine erfüllende Intepretation für C *und* $S \setminus \{C\}$, also ist die Menge S erfüllbar.

Dies ist ein Widerspruch zur Ausgangsannahme. Also muß $\bar{L}$ in einer Klausel D in S vorkommen.

3 Kurze konjunktive Normalform (7 Punkte)

Geben Sie für die Formel

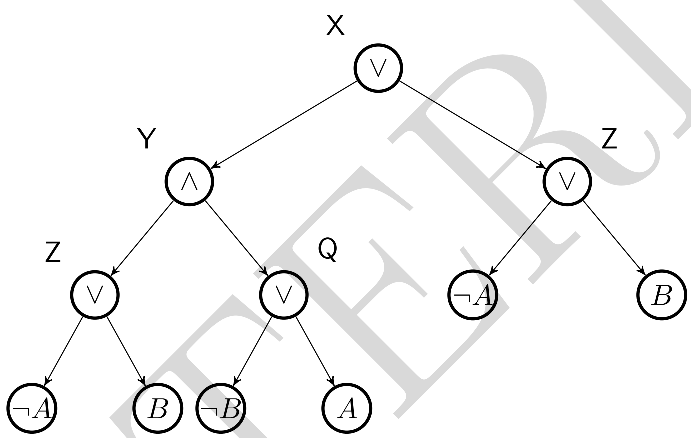
$$(\neg(A \leftrightarrow B)) \rightarrow (A \rightarrow B)$$

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

Als Erstes überführen wir die Formel in NNF: wir teilen die Äquivalenz in zwei Implikationen auf, ersetzen Implikationen durch Disjunktionen, und propagieren die Negationen nach innen. Ergebnis:

$$((\neg A \vee B) \wedge (\neg B \vee A)) \vee (\neg A \vee B)$$

Wir führen neue Variablen ein, die jeweils für den Wahrheitswert einer Teilformel stehen:



N.B. Die Variable Z wurde für zwei Vorkommen der selben Teilformel verwendet. Das ist zwar nicht zwingend, macht die Formel aber einfacher.

Es ergibt sich eine Formel in definitorischer Form:

$$X \wedge \quad X \leftrightarrow (Y \vee Z) \wedge \quad Y \leftrightarrow (Z \wedge Q) \wedge \quad Z \leftrightarrow (\neg A \vee B) \wedge \quad Q \leftrightarrow (\neg B \vee A)$$

Nach Umwandlung der Definitionen in KNF, erhält man die Formel in kurzer KNF:

$$\begin{array}{lrrrr} X \wedge & (\neg X \vee Y \vee Z) \wedge & (\neg Y \vee Z) \wedge & (\neg Z \vee \neg A \vee B) \wedge & (\neg Q \vee \neg B \vee A) \wedge \\ & (X \vee \neg Y) \wedge & (\neg Y \vee Q) \wedge & (Z \vee A) \wedge & (Q \vee B) \wedge \\ & (X \vee \neg Z) \wedge & (\neg Z \vee \neg Q \vee Y) \wedge & (Z \vee \neg B) \wedge & (Q \vee \neg A) \end{array}$$

Es ist auch möglich, die Umwandlung nach NNF wegzulassen, und die Formel direkt nach KKNF zu übersetzen. Das entspricht zwar nicht dem Algorithmus aus der Vorlesung, wurde aber hier auch als korrekt angerechnet.

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.

$$\boxed{\forall x.\, L(x,x)}$$

---

b. Kein Mensch liebt alle Menschen.

$$\boxed{\neg\exists x.\, \forall y.\, L(x,y)}$$

---

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

$$\boxed{\forall x.\, (L(x,x) \to \neg\exists y.\, (L(y,x) \land \neg x \doteq y))}$$

---

d. Jeder Mensch liebt höchstens einen anderen Menschen.

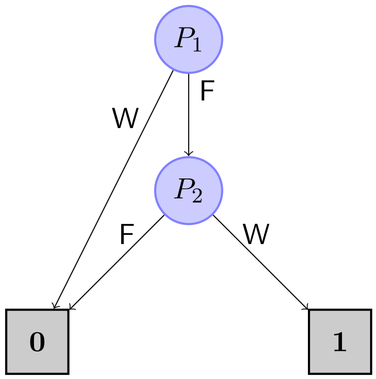
$$\boxed{\forall x, y, z.\ (L(x,y) \land L(x,z) \land \neg x \doteq y \land \neg x \doteq z \to y \doteq z)}$$

---

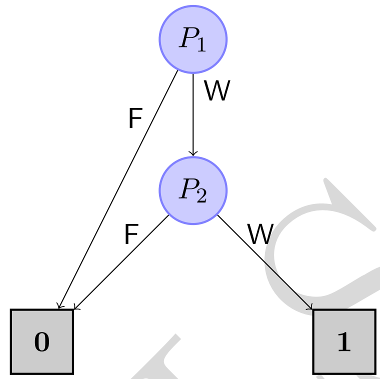
5 Shannongraphen (3+2 Punkte)

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

G_1 :

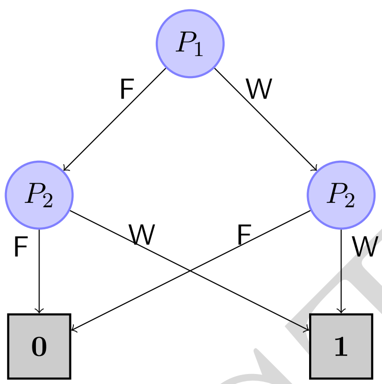


G_2 :



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

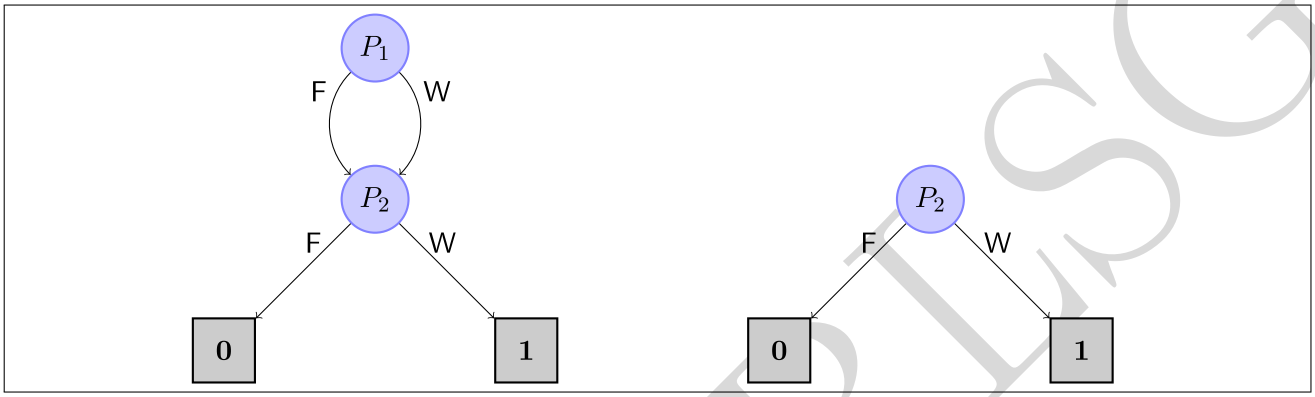
G_3 :



Die Menge aller Pfade zur **1** in G_3 entspricht genau der Vereinigung der Pfade zur **1** von G_1 und G_2.
Bekanntermaßen definiert die Disjunktion aller Pfade zur **1** eine disjunktive Normalform (DNF) eines Shannongraphen. Dabei verknüpft man die genommenen Kanten eines Pfades konjunktiv: Bei Kanten-Markierung W mit dem Literal des verlassenen Knotens, bei Kanten-Markierung F mit der Negation des entsprechenden Literals.
Folglich entspricht die DNF von G_3 genau der Disjunktion der DNF von G_1 und G_2.

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.

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

(4′)	$\{p(g(b,x'),a)\}$		Variante von (4)
(6)	$\{p(f(g(b,x'),a),a)\}$	(1), (4′)	$\sigma = \{x/g(b,x'), y/a\}$
(7)	$\{p(g(b,x'),f(a,a))\}$	(2), (6)	$\sigma = \{x/g(b,x'), y/a, z/a\}$
(8)	$\{p(a,f(g(b,x'),a))\}$	(3), (7)	$\sigma = \{x/g(b,x'), y/a, z/a\}$
(9)	$\{p(a,f(a,g(b,x')))\}$	(3), (8)	$\sigma = \{x/a, y/g(b,x'), z/a\}$
(10)	$\square$	(5), (9)	$\sigma = \{x/b, x'/a\}$

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) requires pos >= 0 && pos <= size;
      @
      @
      @
      @   (2) ensures size == \old(size) + 1;
      @
      @
      @
      @   (3) ensures (\forall int i; pos < i && i < size; arr[i] == \old(arr[i-1]));
      @
      @
      @*/
    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

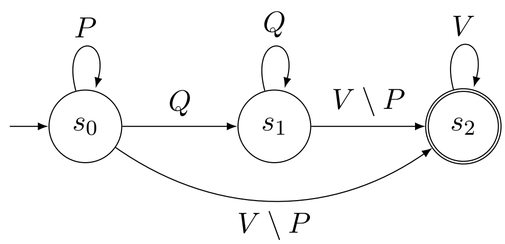
$\Sigma = \{p, q\}$ und das zugehörige Alphabet $V = \mathbb{P}(\Sigma) = \{\emptyset, \{p\}, \{q\}, \{p, q\}\}$.

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 :

$\boxed{p \, \mathbf{U} \, (q \, \mathbf{U} \, \neg p) \equiv \Diamond \neg p}$

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.

$\mathcal{A}_1$: