Klausur Formale Systeme

Fakultät für Informatik

SS 2017

Prof. Dr. Bernhard Beckert

3. August 2017

Name: ______________________________

Vorname: ______________________________

Matrikel-Nr.: ______________________________

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (14)	A2 (6)	A3 (6)	A4 (8)	A5 (11)	A6 (8)	A7 (7)	Σ (60)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung (5+5+4 = 14 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 diese Teilaufgabe 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 ist ein Prädikatensymbole, f ist ein Funktionssymbol und x, y sind Variablen.
- Es gelten die üblichen Klammereinsparungsregeln.

	keine Formel der PL1	allgemeingültig	erfüllbar, aber nicht allgemeingültig	unerfüllbar
$\exists x\, \exists y\, f(x) \doteq f(y)$				
$\forall f\, \exists p\, \forall x\, \forall y\, p(x, y) \leftrightarrow (f(x) \doteq y)$				
$(\forall x\, p(x)) \rightarrow (\exists x\, p(x))$				
$\exists x \exists y(\neg(x \doteq y) \wedge (p(x, y) \leftrightarrow (x \doteq y)) \wedge p(x, y))$				
$(\forall x\, \exists y\, p(x, y)) \rightarrow (\exists x\, \forall y\, p(x, y))$				

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
Zu jeder Substitution σ gibt es eine Substitution ρ, so dass die Komposition $\rho \circ \sigma$ die identische Substitution ist.		
Es gibt eine erfüllbare Formel ϕ der Prädikatenlogik erster Stufe, für die gilt: Alle Modelle von ϕ haben unendliche Universen.		
Enthält eine aussagenlogische Klauselmenge keine Klausel, die nur aus negativen Literalen besteht, dann ist die Klauselmenge erfüllbar.		
Das Weglassen von Regeln aus einem logischen Kalkül erhält seine Vollständigkeit.		
Gilt nach der Ausführung einer Methode die in JML spezifizierte Nachbedingung `a == \old(a)`, dann muss auch die Nachbedingung `a.next == \old(a.next)` gelten.		

Fortsetzung 1 Zur Einstimmung

c. Gegeben sei folgende aussagenlogische Tableauregel:

$$\begin{array}{c|c} \multicolumn{2}{c}{1\ A \vee B} \\ \hline 1\ A & 0\ A \\ 0\ B & 1\ B \end{array}$$

Zeigen Sie anhand eines Beispiels, dass die Regel nicht korrekt ist.

2 Normalformen (6 Punkte)

Beim automatischen Beweisen werden die logischen Formeln vor der Anwendung der Kalkülregeln oft in einer Normalform transformiert (z.B. Pränex- oder Klauselnormalform). Nennen Sie zwei Vorteile und zwei Nachteile der Verwendung von Normalformeln.

Vorteile:

Nachteile:

3 Modallogik (2+2+2 = 6 Punkte)

Zeichnen Sie zu jeder der folgenden modallogischen Formeln jeweils eine Kripkestruktur mit **mindestens drei Welten**, so dass die entsprechende Formel in **jeder** Welt wahr ist. Dabei ist P eine aussagenlogische Variable. Die Erreichbarkeitsbeziehung und der Wahrheitswert von P sollen für jede Welt erkennbar sein.

a. $P \wedge \Box\neg P$

b. $(\neg P \rightarrow \Diamond P) \wedge (P \rightarrow \Diamond\neg P)$

c. $(\Diamond\Diamond\Diamond P) \wedge (P \rightarrow \Diamond\Diamond\neg P)$

4 Formalisieren in PL1 (2+2+2+2 = 8 Punkte)

Gegeben sei die prädikatenlogische Signatur $\Sigma = (\{\mathit{weiß}, \mathit{schwarz}\}, \{\mathit{nachfolger}\}, \alpha)$. Sie enthält die einstelligen Prädikatensymbole $\mathit{weiß}(\cdot)$ und $\mathit{schwarz}(\cdot)$ und das einstellige Funktionssymbl $\mathit{nachfolger}(\cdot)$.

Zur Auswertung der Formeln werden nur solche Interpretationen (D, I) über Σ verwendet, in denen

- das Universum D eine Menge von Objekten ist,
- das Prädikat *weiß* für ein Objekt wahr ist, wenn das Objekt weiß ist,
- das Prädikat *schwarz* für ein Objekt wahr ist, wenn das Objekt schwarz ist,
- die einstellige Funktion *nachfolger* den Nachfolger eines Objekts zurück gibt.

Geben Sie jeweils eine Formel der Prädikatenlogik mit Gleichheit über Σ an, die folgende Sachverhalte darstellt:

a. Jedes Objekt ist entweder weiß oder schwarz.

b. Je zwei verschiedene Objekte haben nicht denselben Nachfolger

c. Jedes schwarze Objekt hat einen weißen Nachfolger, jedes weiße Objekt hat einen schwarzen Nachfolger.

d. Jede Sequenz von drei nacheinanderfolgenden Objekten enthält keinen Zyklus.

5 Beweiskalküle (5+6 = 11 Punkte)

a. Zeigen Sie mit Hilfe des Resolutionskalküls für die Prädikatenlogik, dass folgende Klauselmenge unerfüllbar ist. Notieren Sie bei jedem Schritt die Klauseln auf denen die Resolutionsregel angewandt wurde, sowie **die verwendete Substitution**.

(1) $\{\neg p(d, f(x_1)),\ \neg p(x_1, f(d))\}$

(2) $\{p(x_2, f(x_2)),\ q(d, x_2)\}$

(3) $\{\neg q(x_3, d)),\ p(g(x_4), x_3)\}$

(4) $\{\neg p(g(c)), x_5),\ \neg q(d, d)\}$

Fortsetzung 5 Beweiskalküle

b. Vervollständigen und schließen Sie den folgenden Tableau-Beweis.

Notieren Sie dabei:

- den Regeltyp $(\alpha, \beta, \gamma, \delta)$ und die Formel, auf die eine Regel angewendet wird,
- bei Abschlüssen die beiden Partner,
- sowie die schließende Substitution.

$$1\ \forall x\,\forall y\,\exists z\, p(x,z) \land p(z,y) \qquad (1)$$

$$|$$

$$1\ \forall x\,\forall y\,\forall z\,((p(x,y) \land p(y,z)) \to p(x,z)) \qquad (2)$$

$$|$$

$$0\ p(c,c) \qquad (3)$$

6 Spezifikation mit der Java Modeling Language

(4+4 = 8 Punkte)

a. Geben Sie in natürlicher Sprache wieder, was der folgende JML-Methodenvertrag für die Methode `m` aussagt:

```
public class A {
   int[] a;
  /*@ public normal_behaviour
    @   requires (\forall int i; 0 <= i && i < a.length; 0 <= a[i]);
    @   ensures \result == (\sum int i; 0 <= i && i < a.length;
    @                                 a[i] < 100 ? a[i] : 0);
    @   assignable \nothing;
    @*/
  public int m() { ... }
}
```

Fortsetzung 6 Spezifikation mit der Java Modeling Language

b. Gegeben sei die folgende Java-Klasse `DoubleLinkedList`, mit der eine doppelt-verkettete Liste realisiert wird. Hierbei bezeichnet `head` den Beginn der Liste, das Array `elem` enthält alle Elemente der Liste, inklusive dem `head`-Element. Das Feld `len` bezeichnet die Länge der Liste. Außerdem sind alle Einträge vom Typ `Node` und können jeweils einen Vorgänger (Feld `l`) und einen Nachfolger (Feld `r`) haben.

```
final class DoubleLinkedList {
  final static class Node { /*@ nullable @*/ Node l, r; }
  /*@ nullable @*/ Node head;
  Node[] elem;
  int len;

  /*@ invariant (head == null <==> elem.length == 0)
    @           && elem.length == len
    @           && (head != null ==> elem[0] == head && 0 < len);
    @*/
  //(1)
  /*@ invariant
    @
    @
    @
    @
    @*/
  //(2)
  /*@ invariant
    @
    @
    @
    @
    @*/
}
```

Ergänzen Sie die JML-Klasseninvarianten (1) und (2), so dass sie Folgendes aussagen:

(1) Nacheinanderfolgende Array-Elemente sind auch in der Liste (als Vorgänger bzw. Nachfolger) benachbart.

(2) Für jedes Array-Element außer dem letzen gilt: Der Vorgänger des Nachfolgers des Elements ist das Element selbst.

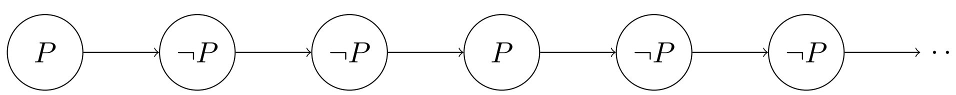
7 Lineare Temporal Logik (LTL) (4+3 = 7 Punkte)

a. Gegeben sei eine Ampel, die ein rotes und ein grünes Licht hat. Die aussagenlogischen Variablen R und G sind genau dann wahr, wenn das rote bzw. das grüne Licht der Ampel leuchtet. Erläutern Sie die LTL-Formel

$$\Box\Big((R \leftrightarrow \neg G) \wedge (R \rightarrow (R\,\mathbf{U}\,G)) \wedge (G \rightarrow (G\,\mathbf{U}\,R))\Big)$$

in eigenen Worten. Bitte beachten Sie dabei den Unterschied zwischen den Operatoren $\mathbf{U}$ und $\mathbf{U}_w$.

b. Geben Sie eine LTL-Formel an, die genau dann wahr ist, wenn die aussagenlogische Variable P genau in den Zeitpunkten ξ_i wahr ist, für die i teilbar durch 3 ist. Der erste Zeitpunkt ist ξ_0.



Hinweis: Verwenden Sie den Next-Operator $\mathbf{X}$.