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
$\exists x \, \exists y \, f(x) \doteq f(y)$		X		
$\forall f \, \exists p \, \forall x \, \forall y \, p(x, y) \leftrightarrow (f(x) \doteq y)$	X			
$(\forall x \, p(x)) \to (\exists x \, p(x))$		X		
$\exists x \exists y (\neg(x \doteq y) \land (p(x, y) \leftrightarrow (x \doteq y)) \land p(x, y))$				X
$(\forall x \, \exists y \, p(x, y)) \to (\exists x \, \forall y \, p(x, y))$			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
Zu jeder Substitution σ gibt es eine Substitution ρ, so dass die Komposition $\rho \circ \sigma$ die identische Substitution ist.		X
Es gibt eine erfüllbare Formel ϕ der Prädikatenlogik erster Stufe, für die gilt: Alle Modelle von ϕ haben unendliche Universen.	X	
Enthält eine aussagenlogische Klauselmenge keine Klausel, die nur aus negativen Literalen besteht, dann ist die Klauselmenge erfüllbar.	X	
Das Weglassen von Regeln aus einem logischen Kalkül erhält seine Vollständigkeit.		X
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.		X

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.

Lösung: Die Formel $A \vee A$ ist erfüllbar, mit dieser Regel kann man aber ableiten, dass sie unerfüllbar ist.

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

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:

- Weniger Regeln
- Standardisierung, Wiederverwendung von Tools/Verfahren
- Vereinfachung der Formelstruktur (z.B. Liste statt Baum)
- Kann die Formel vereinfachen

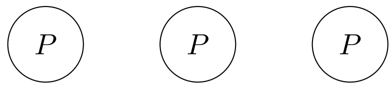
Nachteile:

- Zeitaufwand
- unverständlich, unlesbar
- Formel kann größer werden

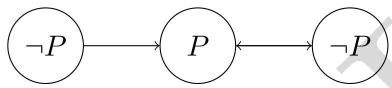
3 Modallogik (2+2+2 = 6 Punkte)

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.

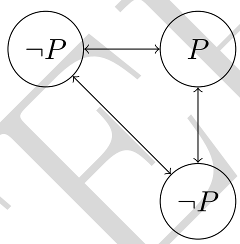
a. $P \wedge \Box\neg P$



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



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



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.

$$\boxed{\forall x\ (\mathit{weiß}(x) \land \neg \mathit{schwarz}(x)) \lor (\neg \mathit{weiß}(x) \land \mathit{schwarz}(x))}$$

b. Je zwei verschiedene Objekte haben nicht denselben Nachfolger

$$\boxed{\forall x\, \forall y\ (\neg x \doteq y \to \neg \mathit{nachfolger}(x) \doteq \mathit{nachfolger}(y))}$$

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

$$\boxed{\forall x\ (\mathit{weiß}(x) \to \mathit{schwarz}(\mathit{nachfolger}(x))) \land (\mathit{schwarz}(x) \to (\mathit{weiß}(\mathit{nachfolger}(x))))}$$

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

$$\boxed{\forall x\ (\neg x \doteq \mathit{nachfolger}(\mathit{nachfolger}(x))) \land (\neg x \doteq \mathit{nachfolger}(x)) \land (\neg x \doteq \mathit{nachfolger}(\mathit{nachfolger}(\mathit{nachfolger}(x))))}$$

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

(5) $\{q(d, d)\}$ aus [1,2] mit $\sigma = \{x_1/d, x_2/d\}$

(6) $\{p(g(x_4), d)\}$ aus [5,3] mit $\sigma = \{x_3/d\}$

(7) $\{\neg p(g(c)), x_5)\}$ aus [5,4] mit $\sigma = \{\}$

(8) $\square$ aus [6,7] mit $\sigma = \{x_4/c, x_5/d\}$

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.

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

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

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

$$1\ \forall y\, \exists z\, p(X_1,z) \land p(z,y) \quad (4[\gamma(1)])$$

$$1\ \exists z\, p(X_1,z) \land p(z,Y_1) \quad (5[\gamma(4)])$$

$$1\ p(X_1, f(X_1,Y_1)) \land p(f(X_1,Y_1),Y_1) \quad (6[\delta(5)])$$

$$1\ p(X_1, f(X_1,Y_1)) \quad (7[\alpha(6)])$$

$$1\ p(f(X_1,Y_1),Y_1) \quad (8[\alpha(6)])$$

$$1\ \forall y\, \forall z\, ((p(X_2,y) \land p(y,z)) \to p(X_2,z)) \quad (9[\gamma(2)])$$

$$1\ \forall z\, ((p(X_2,Y_2) \land p(Y_2,z)) \to p(X_2,z)) \quad (10[\gamma(9)])$$

$$1\ (p(X_2,Y_2) \land p(Y_2,Z_2)) \to p(X_2,Z_2) \quad (11[\gamma(10)])$$

$$0\ p(X_2,Y_2) \land p(Y_2,Z_2) \quad (12[\beta(11)]) \qquad\qquad 1\ p(X_2,Z_2) \quad (17[\beta(11)])$$

$$0\ p(X_2,Y_2) \quad (13[\beta(12)]) \qquad 0\ p(Y_2,Z_2) \quad (15[\beta(12)]) \qquad * \quad (18[3,17])$$

$$* \quad (14[7,13]) \qquad\qquad * \quad (16[8,15])$$

Schließende Substitution: $\sigma = \{X_1/c,\ Y_1/c,\ X_2/c,\ Y_2/f(c,c),\ Z_2/c\}$

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() { ... }
}
```

Wenn die Methode m für ein Array a, in dem ausschließlich nicht-negative Zahlen stehen, aufgerufen wird, dann

- wird keine Ausnahme auftreten,
- ist der Rückgabewert die Summe der Elemente des Arrays a, die vor Ausführung der Methode echt kleiner 100 sind.
- ändert die Methode keine Heap-Stellen.

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.

```
//(1)
/*@ invariant (\forall int i; 0 < i && i < len; elem[i].l == elem[i-1])
  @            && (\forall int i; 0 <= i && i < len-1; elem[i].r == elem[i+1]);
  @*/
//(2)
/*@ invariant (\forall int i; 0 < i && i < len-1; elem[i].r.l == elem[i]);
  @
  @*/
```

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

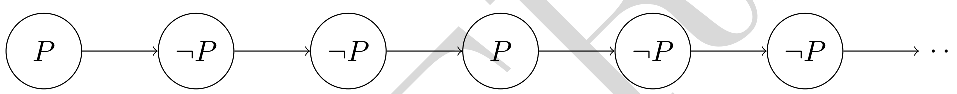
$$\Box\Big((R \leftrightarrow \neg G) \wedge (R \rightarrow (R\,\mathbf{U}\,G)) \wedge (G \rightarrow (G\,\mathbf{U}\,R))\Big)$$

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

Lösung: In jedem Zeitpunkt

- ist die Ampel entweder rot oder grün,
- wenn die Ampel rot ist, dann wird sie irgendwann grün und bis dann bleibt sie rot,
- wenn die Ampel grün ist, dann wird sie irgendwann rot und bis dann bleibt sie grün.

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 $\mathbf{X}$.

$\boxed{P \wedge \Box(P \rightarrow \mathbf{X}\neg P \wedge \mathbf{XX}\neg P \wedge \mathbf{XXX}P)}$