Klausur Formale Systeme

Fakultät für Informatik

WS 2013/2014

Prof. Dr. Peter H. Schmitt

17. Februar 2014

Name: _____

Vorname: _____

Matrikel-Nr.: _____

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (10)	A2 (5)	A3 (6)	A4 (5)	A5 (13)	A6 (7)	A7 (9)	A8 (5)	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“.
- a, b, c, p, q und r sind Prädikatensymbole, g ist ein Funktionssymbol, x und y sind Variablen.
- Es gelten die üblichen Klammereinsparungsregeln.

	keine Formel der PL1	allgemeingültig	erfüllbar, aber nicht allgemeingültig	unerfüllbar
$\big(a \to (b \to c)\big) \to \big((a \to b) \to c\big)$			X	
$\forall x \exists y (g(x) \doteq y)$		X		
$\exists x\ \neg\big(r(g(x), x) \to \forall y\ r(y, g(y))\big)$			X	
$(\exists x\, p(x)) \wedge (\forall y\, \neg p(y))$				X
$\forall x (p(x) \leftrightarrow q(x)) \to (p \leftrightarrow q)$	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
Sei F' eine Teilformel einer PL1-Formel F. Wenn F' erfüllbar ist, so ist auch F erfüllbar.		X
Für jeden Büchiautomaten $\mathcal{A}$ gibt es eine LTL-Formel F, so dass $L^\omega(\mathcal{A}) = \{\xi \in V^\omega \mid \xi \models F\}$ gilt.		X
Sei P eine aussagenlogische Variable und M eine Menge von aussagenlogischen Formeln. Wenn $M \models P$ gilt, so muss P in einer Formel in M auftreten.		X
Wenn die modallogische Formel $\neg A$ eine modale Tautologie ist, dann ist auch $\neg\Diamond A$ eine modale Tautologie.	X	
Für jede AL-Formel gibt es eine äquivalente AL-Formel in KNF mit höchstens drei Literalen pro Klausel.		X

2 Beweisaufgabe (5 Punkte)

Es ist zu zeigen:

> Der Tableaukalkül für Prädikatenlogik (PL1) wie in der Vorlesung vorgestellt, wird unvollständig, wenn man fordert, dass auf jedem Ast eines Tableaus auf jeden Knoten höchstens einmal eine Regel angewendet werden darf.

Geben Sie dazu eine allgemeingültige PL1-Formel φ an, so dass es kein geschlossenes Tableau mit Wurzelknoten 0φ gibt, das die zusätzliche Forderung erfüllt.

Es genügte, eine Formel mit der gesuchten Eigenschaft anzugeben. Die Erklärungen sind nur zur weiteren Erläuterung.
In der Formel

$$\forall x(p(x) \to p(f(x))) \land p(a) \to p(f(f(a)))$$

muss der Quantor zweimal per γ-Regel instanziiert werden, weil zunächst von $p(a)$ auf $p(f(a))$ und daraus dann auf $p(f(f(a)))$ geschlossen werden muss.
Alternativ kann auch die Formel aus Aufgabe 5 herangezogen wergen:

$$\big(\forall x \forall y(p(x,y) \to p(y,x)) \land \forall x \forall y \forall z(p(f(x,y),z) \to p(x,f(y,z))) \land p(a,f(b,c))\big) \to p(f(a,b),c)$$

Aus dem Resolutionsbeweis in Aufgabe 5 geht hervor, dass die beiden ersten Klauseln jeweils 2x mit verschiedener Unifikation als Resolutionspartner benutzt werden. Übertragen auf einen Tableaubeweis heißt das, dass die Quantoren der ersten beiden Voraussetzungen jeweils 2x verschieden instanziiert werden müssen.

3 DPLL (6 Punkte)

Gegeben sei die folgende Menge von aussagenlogischen Klauseln:

$$M := \{\{\neg A, \neg B, \neg C, D\},\ \{\neg A, B\},\ \{\neg B, C\},\ \{\neg A, \neg B, \neg C, \neg D\},\ \{A, \neg B\},\ \{A, B, C\},\ \{B, \neg C\}\}$$

Zeigen Sie mit Hilfe des Davis-Putnam-Loveland-Algorithmus, dass M unerfüllbar ist.

Da M keine Unit-Klauseln enthält, beginnen wir mit einer Fallunterscheidung. Als Variable dafür bietet sich z.B. A an, da dadurch Unit-Klauseln entstehen. Die Ergebnisse der Unit-Propagation sind dabei schrittweise von links nach rechts abgebildet.

- $A \leftarrow 0$

$\underline{A \leftarrow 0}$	$\underline{B \leftarrow 0}$	$\underline{C \leftarrow 1}$
$\neg B, C$		
$\neg B$		
B, C	C	
$B, \neg C$	$\neg C$	$\square$

- $A \leftarrow 1$

$\underline{A \leftarrow 1}$	$\underline{B \leftarrow 1}$	$\underline{C \leftarrow 1}$	$\underline{D \leftarrow 1}$
$\neg B, \neg C, D$	$\neg C, D$	D	
B			
$\neg B, C$	C		
$\neg B, \neg C, \neg D$	$\neg C, \neg D$	$\neg D$	$\square$

Da in beiden Fällen die leere Klausel abgeleitet worden ist, ist M unerfüllbar.

4 Unifikation (2+1+2 Punkte)

Seien

- f ein zweistelliges Funktionssymbol,
- g ein einstelliges Funktionssymbol,
- c, d nullstellige Funktionssymbole,
- x, y, z Variablen.

Bestimmen Sie für die folgenden Mengen von Termen einen allgemeinsten Unifikator μ (als *eine* Substitution und nicht als Verkettung von Substitutionen). Falls es keinen allgemeinsten Unifikator gibt, begründen Sie, warum es keinen gibt!

a. $\{\ f(x, g(f(x, f(x, c)))),\ f(x, g(f(x, f(z, c)))),\ f(z, g(f(g(y), f(y, c))))\ \}$

1. $\{\ f(z, g(f(z, f(z, c)))),\ f(z, g(f(g(y), f(y, c))))\ \}$ $\mu = \{x/z\}$
2. $\{\ f(g(y), g(f(g(y), f(g(y), c)))),\ f(g(y), g(f(g(y), f(y, c))))\ \}$ $\mu = \{z/g(y)\} \circ \{x/z\} = \{x/g(y), z/g(y)\}$
3. Occur check: $y \in Var(g(y)) \Rightarrow$ nicht unifizierbar.

b. $\{\ f(c, f(x, y)),\ f(x, f(y, d))\ \}$

1. $\{\ f(c, f(c, y)),\ f(c, f(y, d))\ \}$ $\mu = \{x/c\}$
2. $\{\ f(c, f(c, c)),\ f(c, f(c, d))\ \}$ $\mu = \{y/c\} \circ \{x/c\} = \{y/c, x/c\}$
3. Nicht unifizierbar, die Menge enthält mehr als einen Term und die Differenz der Menge enthält keine Variable.

c. $\{\ f(x, f(z, x)),\ f(y, f(g(c), y))\ f(y, f(z, f(z, z)))\ \}$

1. $\{\ f(x, f(z, x)),\ f(x, f(g(c), x))\ f(x, f(z, f(z, z)))\ \}$ $\mu = \{y/x\}$
2. $\{\ f(x, f(g(c), x)),\ f(x, f(g(c), f(g(c), g(c))))\ \}$ $\mu = \{z/g(c)\} \circ \{y/x\} = \{z/g(c), y/x\}$
3. $\{\ f(f(g(c), g(c)), f(g(c), f(g(c), g(c))))\ \}$ $\mu = \{x/f(g(c), g(c))\} \circ \{z/g(c), y/x\}$ $= \{x/f(g(c), g(c)), z/g(c), y/f(g(c), g(c))\}$

   $\Rightarrow \mu = \{x/f(g(c), g(c)), z/g(c), y/f(g(c), g(c))\}$ ist ein allgemeinster Unifikator.

5 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 einstelliges Prädikatensymbol, f ist ein zweistelliges Funktionssymbol, a, b, c 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.

$$\Big\{\underbrace{\{\neg p(x,y), p(y,x)\}}_{(1)} \;,\; \underbrace{\{\neg p(f(x,y),z), p(x,f(y,z))\}}_{(2)} \;,\; \underbrace{\{p(a,f(b,x))\}}_{(3)} \;,\; \underbrace{\{\neg p(c,f(a,x))\}}_{(4)}\Big\}$$

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

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

Gegeben ist die folgende Java-Klassendefinition:

```
class RemoveDup {
    public static int[] removeDup(int[] a) {
        ...
    }
}
```

Die Methode `removeDup` soll Duplikate aus einer Liste von Zahlen entfernen. Sie erhält als Argument ein `int`-Array `a` und liefert als Ergebnis ein neu erstelltes Array, in dem dieselben Werte wie `a` enthalten sind, jedoch jeder Wert höchstens einmal. Es ist sichergestellt, dass die Methode die Einträge des Arrays `a` nicht verändert.

a. Geben Sie eine JML-Nachbedingung für die Methode `removeDup` an, die besagt, dass das Ergebnis-Array höchstens so lang ist wie die Eingabe `a`.

```
ensures \result.length <= a.length;
```

ensures

b. Geben Sie eine JML-Nachbedingung für die Methode `removeDup` an, die besagt, dass im Ergebnis-Array kein Wert doppelt vorkommt.

```
ensures (\forall int i; (\forall int j;
                0 <= i && i < j && j < result.length; \result[i] != \result[j]));
```

ensures

c. Geben Sie eine JML-Nachbedingung für die Methode `removeDup` an, die besagt, dass jeder Wert, der im Array **a** vorkommt, auch im Ergebnis auftritt.

```
ensures (\forall int i; 0<=i && i < a.length;
              (\exists int j; 0<=j && j < \result.length; \result[j] == a[i]));
```

ensures

7 Formalisieren in LTL (2+3+4 Punkte)

Formalisieren Sie folgende Aussagen in linearer temporaler Logik (LTL). Benutzen Sie jeweils die in Klammern angegebenen atomaren Aussagen.

a. Ab dem Aufbau einer Netzwerkverbindung (*connect*) erscheint nach einem Fehler (*error*) irgendwann eine Meldung (*message*).

$\Box(connect \rightarrow \Box(error \rightarrow \Diamond message))$

b. Die Datei kann nur gelesen werden (*read*), wenn sie vorher geöffnet worden ist (*open*).

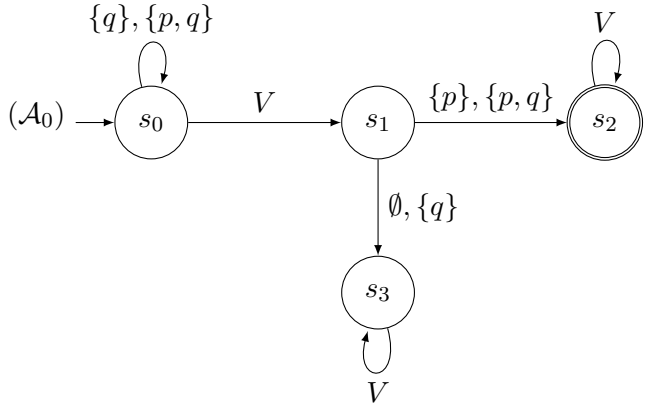
$open\ \mathbf{V}\ \neg read \equiv \neg read\ \mathbf{U_W}\ (open \wedge \neg read) \equiv \neg(\neg open\ \mathbf{U}\ read)$

c. Die Datei kann nur gelesen werden (*read*), wenn sie *unmittelbar* vorher geöffnet worden ist (*open*). Wir nehmen wie üblich an, ein Zeitpunkt t liegt unmittelbar vor Zeitpunkt t' gdw. $t = t' - 1$.

$\neg read \wedge \Box(\mathbf{X} read \rightarrow open)$

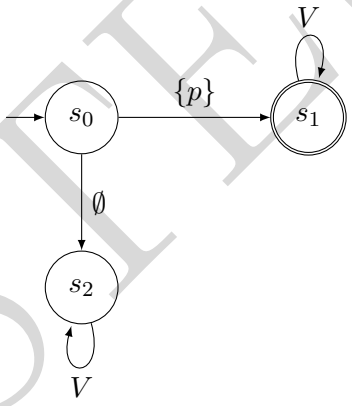
8 LTL und Büchi-Automaten (3+2 Punkte)

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



$A_0 =$ $q\ U\ (X\ p)$

b. Geben Sie einen Büchi-Automaten $\mathcal{A}_1$ über dem Alphabet $V = \{\emptyset, \{p\}\}$ an, der genau $p\ U\ p$ akzeptiert, d.h. so dass $L^\omega(\mathcal{A}_1) = \{\xi \in V^\omega : \xi \models p\ U\ p\}$ gilt.



(Die LTL-Formel $p\ U\ p$ ist äquivalent zu p.)