



Klausur Formale Systeme
Fakultät für Informatik
SS 2016

Prof. Dr. Bernhard Beckert

05. August 2016

Vorname: **Vorname**
Name: **Familiennam**
Matrikel-Nr.: **Matr.-Nr.**
Platz-Nr.: **Hörsaal** **Sitzplatz**
Code: **Nonce**

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (13)	A2 (5)	A3 (8)	A4 (8)	A5 (11)	A6 (7)	A7 (8)	Σ (60)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung

(5+5+3 = 13 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, q, r und s sind Prädikatensymbole, f ist ein Funktionssymbol, c ist ein Konstantensymbol und x, y sind Variablen.
- Es gelten die üblichen Klammereinsparungsregeln.

	keine Formel der PL1	allgemeingültig	erfüllbar, aber nicht allgemeingültig	unerfüllbar
$(\exists x p(x)) \leftrightarrow (p(c) \wedge (\exists x p(x)))$			X	
$\forall x \forall y (p(x) \vee \neg p(y))$			X	
$\exists f \forall x (f(x) \doteq x)$	X			
$\neg q \wedge (\neg r \rightarrow (\neg s \wedge q)) \wedge (r \rightarrow (s \wedge q))$				X
$((\exists x \neg p(x)) \rightarrow \mathbf{0}) \rightarrow p(f(c))$		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
Die Zahl der Knoten eines reduzierten Shannon-Graphen zu einer Formel F ist unabhängig von der gewählten Variablenordnung.		X
Sei F eine Formel der PL1. Aus der Vollständigkeit des Tableauekalküls folgt, dass ein geschlossenes Tableau für F oder eines für $\neg F$ existiert.		X
Der Unifikationsalgorithmus nach Robinson terminiert immer.	X	
Für Formeln der PL1, die weder Variablen noch Quantoren enthalten (d. h., alle Terme sind Grundterme), ist das Erfüllbarkeitsproblem entscheidbar.	X	
Für jeden Büchi-Automaten \mathcal{A} über dem Alphabet $\mathbb{P}(\Sigma)$ gibt es eine LTL-Formel F über der Signatur Σ , so dass die von \mathcal{A} akzeptierten ω -Strukturen genau die Modelle von F sind.		X

Fortsetzung 1 Zur Einstimmung

c. Nicht jedes Reduktionssystem, das *lokal konfluent* ist, ist auch *konfluent*.

Geben Sie dafür ein Beispiel! Das heißt: Geben Sie ein Reduktionssystem (D, \succ) an, das

- zwar *lokal konfluent*,
- jedoch *nicht konfluent*

ist.

$D = \{a, b, c, d\}$

$a \succ b, b \succ a, a \succ c, b \succ d$

Dieses Reduktionssystem ist lokal konfluent. Es ist aber nicht konfluent, da zum einen $a \rightarrow c, a \rightarrow d$ und zum anderen c, d irreduzible Elemente sind, und es also kein $s \in D$ mit $c \rightarrow s$ und $d \rightarrow s$ gibt.

2 Peano-Axiome und Arithmetik

(2+3 Punkte)

Seien (wie in der Vorlesung)

- PA
die (Menge der) Peano-Axiome,
- $Cn(PA) = \{\phi \in Fml_{\Sigma_{PA}} \mid PA \models \phi\}$
die Menge der prädikatenlogischen Formeln, die aus den Peano-Axiomen logisch folgen,
- $\mathcal{N} = (\mathbb{N}, +, *, 0, 1)$
die prädikatenlogische Struktur, deren Universum die Menge der natürlichen Zahlen ist und in der $+, *, 0, 1$ wie in der Arithmetik üblich interpretiert sind,
- $Th(\mathcal{N}) = \{\phi \in Fml_{\Sigma_{PA}} \mid \mathcal{N} \models \phi\}$
die Menge der in \mathcal{N} wahren Formeln.

a. Welche der Mengen $Cn(PA)$ und $Th(\mathcal{N})$ ist *rekursiv aufzählbar*?

Keine

Nur $Cn(PA)$

Nur $Th(\mathcal{N})$

Beide

$Cn(PA)$ ist aufzählbar (mit Hilfe eines beliebigen korrekten und vollständigen Kalküls für die Prädikatenlogik). $Th(\mathcal{N})$ ist nicht aufzählbar (Gödelscher Unvollständigkeitssatz).

b. Welche Bedeutung hat der Unterschied zwischen $Cn(PA)$ und $Th(\mathcal{N})$ für die Praxis?
Begründen Sie (kurz) ihre Antwort!

Die Bedeutung der Lücke zwischen $Cn(PA)$ und $Th(\mathcal{N})$ für die Praxis ist *gering*.

Es gibt wenig relevante/verständliche Beispiele für ϕ mit $\phi \in Th(\mathcal{N})$ und $\phi \notin Cn(PA)$.

Nichtstandardmodelle von PA (Modelle, die nicht isomorph zu \mathcal{N} sind) sind alle „seltsam“. Dies illustriert der Satz von Tennenbaum: In allen Nichtstandardmodellen von PA sind Addition und Multiplikation nicht-berechenbare Funktionen.

3 DPLL

(8 Punkte)

Gegeben sei die folgende Menge M aussagenlogischer Klauseln:

$$\{\{\neg A, \neg C, D\}, \{\neg D, \neg B, A\}, \{\neg D, E, \neg B\}, \{\neg C, \neg B, A\}, \{\neg D, \neg B, \neg E\}, \{E, \neg A, D\}, \{C\}\}$$

Zeigen Sie mit Hilfe des Davis-Putnam-Loveland-Algorithmus, dass M **erfüllbar** ist. Geben Sie die Zwischenschritte und das sich ergebende **Modell** an.

Wichtig:

1. Wenn Sie Fallunterscheidungen machen müssen, nehmen Sie diese in **alphabetischer Reihenfolge** vor (also A vor B , usw.).
2. Betrachten Sie **zuerst** immer komplett den Fall mit **positiver Polarität**, bevor Sie mit dem negativen Fall anfangen.
3. Es reicht, nur **ein** Modell zu finden, Sie müssen also nicht den gesamten Belegungsraum durchsuchen!

<u>Originalklauseln</u>	<u>Unit C</u>	<u>A ← 1</u>	<u>Unit D</u>
$\{\neg A, \neg C, D\}$	$\{\neg A, D\}$	$\{D\}$	–
$\{\neg D, \neg B, A\}$	$\{\neg D, \neg B, A\}$	–	–
$\{\neg D, E, \neg B\}$	$\{\neg D, E, \neg B\}$	$\{\neg D, E, \neg B\}$	$\{E, \neg B\}$
$\{\neg C, \neg B, A\}$	$\{\neg B, A\}$	–	–
$\{\neg D, \neg B, \neg E\}$	$\{\neg D, \neg B, \neg E\}$	$\{\neg D, \neg B, \neg E\}$	$\{\neg B, \neg E\}$
$\{E, \neg A, D\}$	$\{E, \neg A, D\}$	$\{E, D\}$	–
$\{C\}$	–	–	–
<u>B ← 1</u>	<u>Unit E</u>	<u>Backtracking B</u>	
–	–		
–	–		
$\{E\}$	–		
–	–		
$\{\neg E\}$	□		
–	–		
–	–		
<u>B ← 0</u>	<u>Menge erfüllbar</u>		
–			
–			
–			
–			
–			
–			
–			
Modell: $A \leftarrow 1, B \leftarrow 0, C \leftarrow 1, D \leftarrow 1, E \leftarrow 1$			

4 Formalisieren in PL1

(8 Punkte)

Formalisieren Sie folgende Aussagen in Prädikatenlogik erster Stufe mit Gleichheit. Benutzen Sie dafür die jeweils angegebenen interpretierten Symbole.

- a. Hunde und Katzen sind Säugetiere.
Prädikate: $hund(\cdot)$, $katze(\cdot)$, $säugetier(\cdot)$.

$$\forall x. ((hund(x) \vee katze(x)) \rightarrow säugetier(x))$$

- b. Nur eine tote Spinne ist eine gute Spinne.
Prädikate: $spinne(\cdot)$, $tot(\cdot)$, $gut(\cdot)$.

$$\forall x. ((spinne(x) \wedge gut(x)) \rightarrow tot(x))$$

- c. Wer eine Katze hat, hat keinen Hund.
Prädikate: $katze(\cdot)$, $hund(\cdot)$, $hat(\cdot, \cdot)$.

$$\forall x. (\exists y. katze(y) \wedge hat(x, y)) \rightarrow (\neg \exists y. hund(y) \wedge hat(x, y))$$

- d. Alle Teilnehmer außer einem haben bestanden.
Prädikate: $teilnehmer(\cdot)$, $bestanden(\cdot)$.

$$\exists x. teilnehmer(x) \wedge \neg bestanden(x) \wedge \forall y. (teilnehmer(y) \wedge \neg y \doteq x \rightarrow bestanden(y))$$

5 Tableaukalkül + Resolution

(4+7 = 11 Punkte)

Im Folgenden sind p, q einstellige Prädikatensymbole, f, g einstellige Funktionssymbole, c ein Konstantensymbol und alle sonstigen Bezeichner Variablensymbole.

- a. Bilden Sie alle möglichen Klauseln, die durch **je einen einzigen** Resolutionsschritt aus den folgenden PL1-Klauseln ableitbar sind.

Notieren Sie die Beweisschritte so, dass bei jeder neu entstehenden Klausel klar erkennbar ist, aus welchen Elternklauseln sie entsteht und welche Substitution verwendet wurde.

$$\{q(c)\}$$

(1)

$$\{\neg q(x), q(f(x))\}$$

(2)

$$\{\neg q(f(x)), q(g(x))\}$$

(3)

Nr.	Eltern	Klausel	Substitution
(4)	(1), (2)	$\{q(f(c))\}$	$\{x/c\}$
(5)	(2), (2')	$\{\neg q(x), q(f(f(x)))\}$	$\{x'/f(x)\}$
(6)	(2'), (3)	$\{\neg q(x), q(g(x))\}$	$\{x'/x\}$
(7)	(2'), (3)	$\{q(f(g(x))), \neg q(f(x))\}$	$\{x'/g(x)\}$

Wobei (2') eine Variante von (2) ist, in der x in x' umbenannt ist: $\{\neg q(x'), q(f(x'))\}$

Fortsetzung 5 Tableauekalkül + Resolution

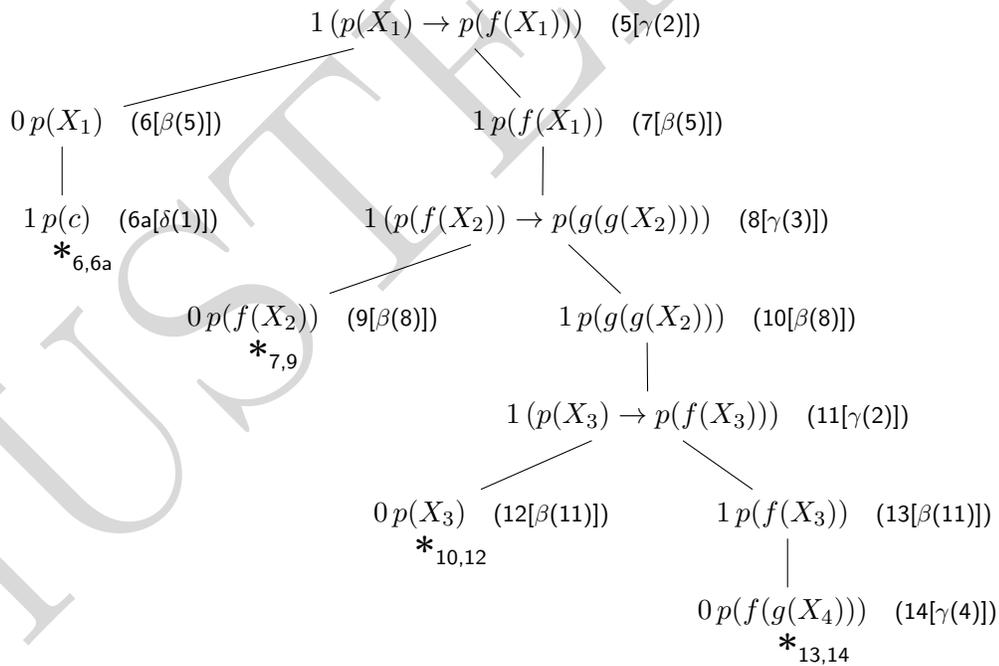
b. Vervollständigen und schließen Sie den folgenden Tableau-Beweis.

Hinweis: Auf eine der γ -Formeln muss zweimal eine Regel angewendet werden.

Notieren Sie bei Ihrem Beweis:

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

$$\begin{array}{l}
 1 \exists x p(x) \quad (1) \\
 | \\
 1 \forall x (p(x) \rightarrow p(f(x))) \quad (2) \\
 | \\
 1 \forall x (p(f(x)) \rightarrow p(g(g(x)))) \quad (3) \\
 | \\
 0 \exists x p(f(g(x))) \quad (4) \\
 |
 \end{array}$$



Schließende Substitution: $\mu = \{X_1/c, X_2/c, X_3/g(g(c)), X_4/g(c)\}$

6 Spezifikation mit der Java Modeling Language

(4+3 = 7 Punkte)

- a. Gegeben sei die Methode `doubleEvenNumbers(int[] a)`, die folgende Funktionalität implementiert:

Die Methode ändert das Array `a`, indem *genau* die Elemente von `a` mit 2 multipliziert werden, deren Werte *gerade* sind.

(Beispielsweise ist bei Eingabe von `a = {3, 2, 4}` das resultierende Array `a = {3, 4, 8}`.)

Formalisieren Sie diese Verhaltensbeschreibung, indem Sie einen präzisen und vollständigen JML-Vertrag für diese Methode angeben:

```
/*@ public normal_behavior
   @ requires true;
   @
   @ ensures
   @
   @
   @
   @
   @
   @
   @
   @ assignable
   @*/
   public void doubleEvenNumbers(int[] a) { ... }
```

```
/*@ public normal_behavior
   @ ensures (\forallall int i; 0 <= i && i < a.length;
   @     ((\old(a[i]) % 2 == 0) ==> a[i] == \old(a[i]) * 2)
   @     && ((\old(a[i]) % 2 != 0) ==> a[i] == \old(a[i])));
   @ assignable a[*];
   @*/
   public void doubleEvenNumbers(int[] a) { ... }
```

- b. Geben Sie die Bedeutung des folgenden Methodenvertrags in natürlicher Sprache wieder:

```
/*@ public normal_behavior
   @ ensures (\exists int j, i; 0 <= i && i < a.length && 0 <= j && j <= i;
   @     a[i] == a[j] && i - j == \result);
   @ ensures (\forallall int j, i; 0 <= i && i < a.length && 0 <= j && j <= i;
   @     a[i] == a[j] ==> i - j <= \result);
   @*/
   public int f(int[] a) { ... }
```

Hinweis: Es gibt genau drei Unterschiede zwischen der ersten und zweiten Nachbedingung. Diese Unterschiede sind durch Unterstreichen und **Fettdruck** hervorgehoben.

Die Methode liefert den Abstand zwischen den am weitesten entfernten Duplikaten im Array zurück, falls mindestens ein Duplikat existiert; andernfalls ist das Ergebnis 0.

MUSTERLÖSUNG

7 Büchi-Automaten und LTL

(4+3+1 = 8 Punkte)

- a. Gegeben sei die Signatur $\Sigma = \{p, q\}$ und das zugehörige Alphabet $V = \mathbb{P}(\Sigma) = \{\emptyset, \{p\}, \{q\}, \{p, q\}\}$. Geben Sie einen Büchi-Automaten \mathcal{A} über dem Alphabet V an, der genau die LTL-Formel

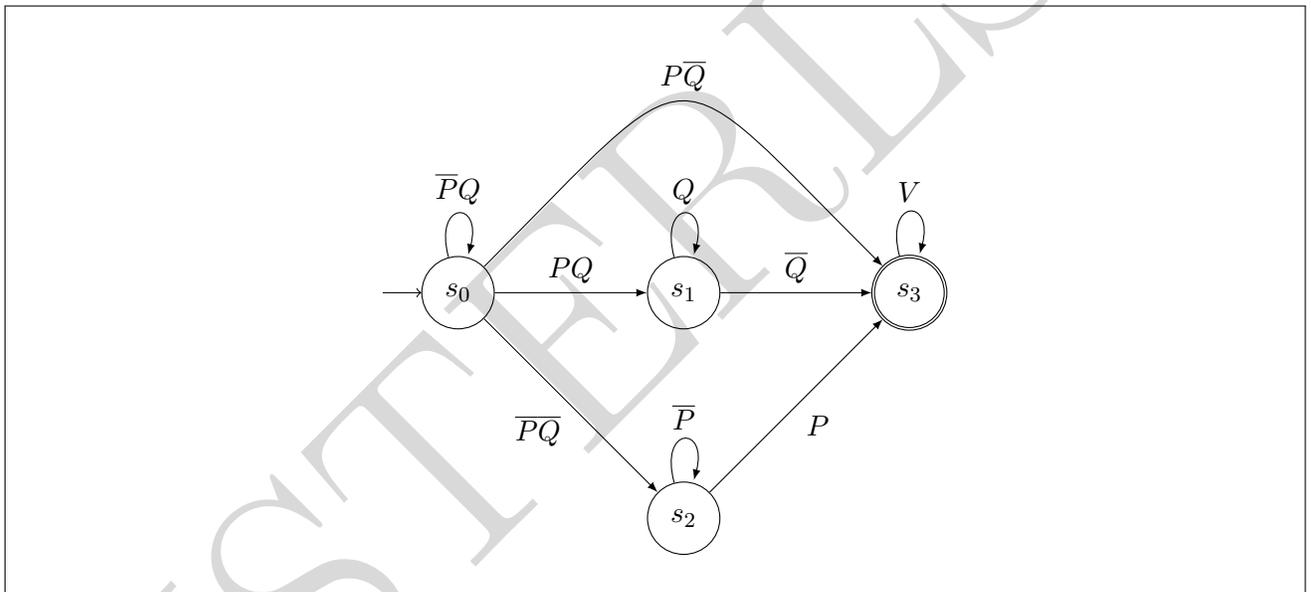
$$(\diamond p) \wedge (\diamond \neg q)$$

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

Es können dabei die folgenden Abkürzungen verwendet werden:

$$\begin{aligned} P &= \{\{p\}, \{p, q\}\}, & \bar{P} &= \{\emptyset, \{q\}\}, & Q &= \{\{q\}, \{p, q\}\}, & \bar{Q} &= \{\emptyset, \{p\}\} \\ PQ &= \{\{p, q\}\}, & \bar{P}\bar{Q} &= \{\{q\}\}, & P\bar{Q} &= \{\{p\}\} & \bar{P}Q &= \{\emptyset\} \end{aligned}$$

\mathcal{A} :



- b. Geben Sie zu folgenden LTL-Formeln φ_1 und φ_2 jeweils eine äquivalente LTL-Formel an, die als temporale Modaloperatoren nur \square und \diamond verwendet (d. h., \mathbf{U} bzw. \mathbf{U}_w kommt nicht vor):

$$\varphi_1 \equiv \diamond(a \mathbf{U} b) \quad \leftrightarrow \quad \boxed{\diamond b}$$

$$\varphi_2 \equiv \diamond(a \mathbf{U}_w b) \quad \leftrightarrow \quad \boxed{\diamond(\square a \vee b)}$$

- c. Formalisieren Sie den folgenden natürlichsprachigen Sachverhalt über der atomaren Aussage A in LTL:

Es gibt keine direkt aufeinanderfolgenden Zeitpunkte, in denen A wahr ist.

$$\boxed{\square(A \rightarrow X\neg A)}$$