Klausur Formale Systeme

Universität Karlsruhe
Fakultät für Informatik

WS 2008/2009

Prof. Dr. Bernhard Beckert

19. Februar 2009

Name:	
Vorname:	
Matrikel-Nr.:	

Bitte kleben Sie hier Ihren Platzaufkleber auf!

Bitte geben Sie auf jedem benutzten Blatt rechts oben Ihren Namen und Ihre Matrikel-Nummer an!

A1 (15)	A2 (6)	A3 (5)	A4 (8)	A5 (10)	A6 (9)	A7 (7)	Σ (60)

Bewertungstabelle bitte frei lassen!

Zum Bestehen der Klausur sind 20 der erreichbaren 60 Punkte hinreichend.

Bonus: ______ **Gesamtpunkte:**

1 Zur Einstimmung (4+5+3+3 = 15 Punkte)

Kreuzen Sie in den folgenden Tabellen alles Zutreffende an.

Für jede falsche Antwort wird ein halber Punkt abgezogen!
(Dabei werden jedoch keinesfalls weniger als 0 Punkte für jede der vier 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".
- In Teilaufgabe a. kann eine Formel mehr als eine der genannten Eigenschaften haben. In Teilaufgabe b. und c. *genau* eine.
- p, q, r, s und t sind Prädikatssymbole, f ist ein Funktionssymbol, und x und y sind Variablen.
- Es gelten die üblichen Klammereinsparungsregeln.

a.

	keine Formel der PL1	erfüllbar	allgemein-gültig	uner-füllbar
$\big(\forall x \exists y (p(x) \to q(y))\big) \leftrightarrow \big((\exists x\, p(x)) \to (\exists y\, q(y))\big)$		X	X	
$\big((r \wedge s) \to t\big) \to s$		X		
$\forall x (p(x) \to q(p(x)))$	X			
$\exists x \big(p(x) \to p(f(x))\big)$		X	X	

b.

	Richtig	Falsch
Die modallogische Formel $\Diamond \mathbf{1}$ charakterisiert die endlosen Kripkerahmen.	X	
Zu jedem Büchi-Automaten $\mathcal{A}$ gibt es einen *deterministischen* Büchi-Automaten $\mathcal{A}'$, so dass $L^\omega(\mathcal{A}) = L^\omega(\mathcal{A}')$ gilt.		X
Für jede geschlossene prädikatenlogische Formel G gilt: Es gibt ein Modell für G oder für das Negat von G oder für beide.	X	
Das Erfüllbarkeitsproblem der Prädikatenlogik ist entscheidbar.		X
Für alle PL1-Formeln G, G_1, G_2 gilt: Wenn G sowohl mit G_1 als auch mit G_2 unifizierbar ist, dann sind auch G_1 und G_2 miteinander unifizierbar.		X

1 Zur Einstimmung *(Fortsetzung)*

c.

	Ja	Nein
Für alle aussagenlogischen Klauselmengen M gilt: Falls jede Klausel in M mindestens ein positives Literal enthält, dann ist M erfüllbar.	X	
Für alle aussagenlogischen Klauselmengen M gilt: Falls jede aussagenlogische Variable, die in M vorkommt, sowohl negiert als auch nicht negiert in M auftritt, dann ist M unerfüllbar.		X
Jede 2KNF-Formel (Kromformel) ist auch eine Hornformel.		X

d. Sind folgende modallogische Formeln allgemeingültig, d.h., gelten sie in allen Kripkestrukturen?

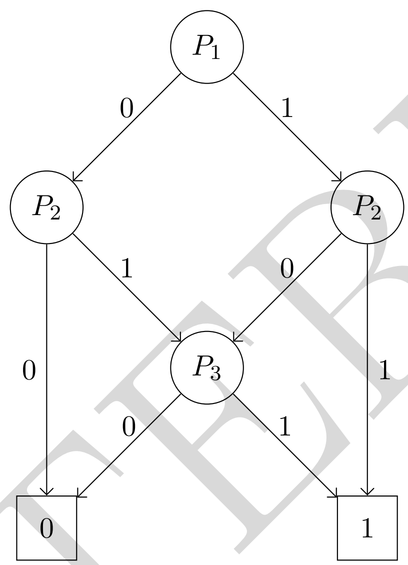
Modallogische Formel	Allgemeingültig	Nicht allgemeingültig
$(\Box A \wedge \Diamond\Diamond A) \rightarrow \Box\Diamond A$		X
$\Box A \vee \Diamond(\neg A \vee B)$	X	
$\Box\big((A \wedge \Diamond B) \rightarrow A\big)$	X	

2 Shannongraphen (6 Punkte)

Gegeben sei die aussagenlogische Signatur $\Sigma = \{P_1, P_2, P_3\}$ mit der Variablenordnung $P_1 < P_2 < P_3$, und die aussagenlogische Formel G habe die folgende Eigenschaft:

> G wird *genau* in denjenigen Interpretation $I : \Sigma \to \{W, F\}$ zu wahr ausgewertet, die mindestens zwei Variablen aus Σ mit W (wahr) belegen.

Geben Sie den reduzierten Shannongraphen für G an.



3 Unifikation und Substitution (4+1 = 5 Punkte)

Gegeben sei eine Signatur, die das Funktionssymbol f, das Konstantensymbol c und das Prädikatensymbol p enthält; x_1, x_2 und x_3 sind Variablen.

a. Geben Sie für die folgenden beiden Formelpaare jeweils an, ob sie unifizierbar sind.
Wenn ja, geben Sie zusätzlich an:

– einen allgemeinsten Unifikator,
– die Ergebnisformel

i. $p(x_1, x_2, x_3), \quad p(f(x_2), f(x_3), c)$

Unifizierbar.
Allgemeinster Unifikation: $\{x_1/f(f(c)), x_2/f(c), x_3/c\}$
Ergebnis: $p(f(f(c)), f(c), c)$

ii. $p(x_1, x_2, x_3), \quad p(f(x_2), f(x_3), x_1)$

Nicht unifizierbar.
Der Robinsonalgorithmus lässt als Zwischenergebnis die Substitution $\{x_1/f(f(x_3)), x_2/f(x_3)\}$ und das Termpaar $p(f(f(x_3)), f(x_3), x_3), p(f(f(x_3)), f(x_3), f(f(x_3)))$ entstehen. Die Teilterme x_3 und $f(f(x_3))$ sind aber nicht unifizierbar (Occur Clash).

b. Geben Sie ein Beispiel für eine prädikatenlogische Formel φ und eine Substitution σ an, so dass σ *nicht* kollisionsfrei für φ ist.

$\varphi = \boxed{\forall x\, p(y)}$

$\sigma = \boxed{\{y/x\}}$

4 Semantik der Prädikatenlogik (6+2 = 8 Punkte)

Gegeben sei folgende prädikatenlogische Formel G:

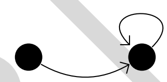
$$
\begin{aligned}
& \forall x \exists y\, r(x,y) \\
\wedge\ & \forall x \exists y\, \neg r(x,y) \\
\wedge\ & \forall x \forall y \forall z \big((r(x,y) \wedge r(y,z)) \rightarrow r(x,z) \big)
\end{aligned}
$$

a. Die Formel G ist erfüllbar.
Geben Sie von allen Modellen von G ein solches Modell (D, I) an, bei dem die Anzahl der Elemente des Universums D **minimal** ist.

Hinweis: Sie können dazu einen Graphen mit Knotenmenge D und Kantenmenge $I(r)$ skizzieren.

Die minimalen Modelle von G haben zwei Elemente. Sie lassen sich wie folgt skizzieren:

oder



b. Wir betrachten nun die Formel G':

$$G \ \wedge\ \forall x\, \neg r(x,x)$$

Geben Sie eine Interpretation (D', I') mit $D' = \mathbb{Z}$ an, die ein Modell von G' ist.

$I'(r) = <$, d.h. $(a,b) \in I'(r) \iff a < b$

5 Resolutionskalkül (10 Punkte)

Gegeben sei eine Signatur Σ, die die Konstante a, das einstellige Funktionssymbol s und das dreistellige Prädikatensymbol p enthält.
Ferner sei folgende Formel G über Σ gegeben:

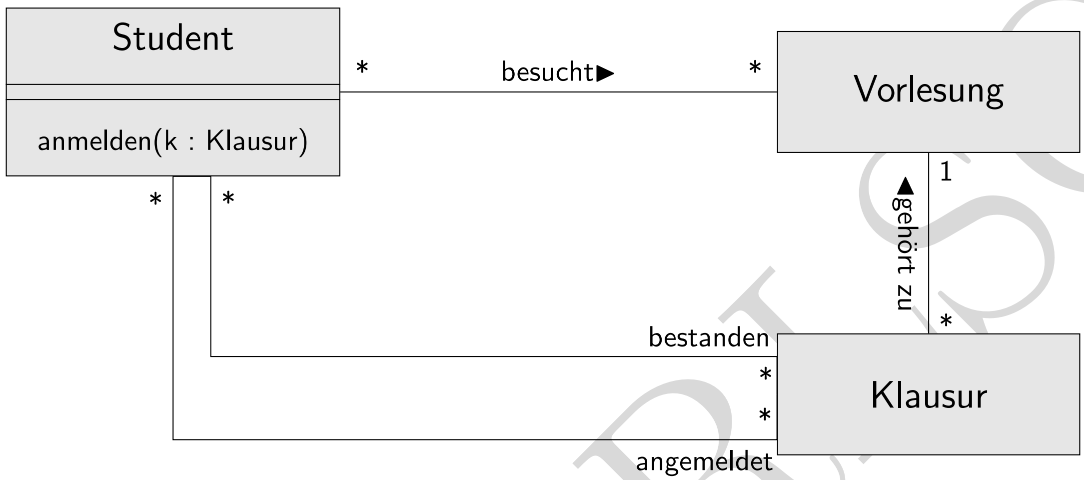
$$\begin{array}{ll} & \forall x\, p(a,x,x) \\ \wedge & \forall u \forall v \forall w\, (p(u,v,w) \rightarrow p(s(u),v,s(w)) \\ \wedge & \neg p(s(s(a)),s(a),s(s(s(a)))) \end{array}$$

Zeigen Sie mit Hilfe des Resolutionskalküles, dass G unerfüllbar ist.

Notieren Sie Ihren Beweis so, dass bei jeder neu entstehenden Klausel klar erkennbar ist, aus welchen Elternklauseln sie entsteht.

[1]	$\{p(a,x,x)\}$	
[2]	$\{\neg p(u,v,w), p(s(u),v,s(w))\}$	
[3]	$\{\neg p(s(s(a)),s(a),s(s(s(a))))\}$	
[4](2, 3)	$\{\neg p(s(a),s(a),s(s(a)))\}$	$\{u/s(a), v/s(a), w/s(s(a))\}$
[5](4, 2)	$\{\neg p(a,s(a),s(a)\}$	$\{u/a, v/s(a), w/s(a)\}$
[6](5, 1)	$\square$	$\{x/s(a)\}$

UML-Diagramm zu Aufgabe 6



Übersicht über wichtige OCL-Operationen

Folgende Operationen sind auf alle Gesamtheiten (Mengen, Multimengen und Listen) anwendbar.

Operation	**Ergebnis** (bei Anwendung auf M)
`size()`	die Anzahl der Elemente in M.
`including(o)`	die Gesamtheit, die M erweitert um o entspricht.
`collect(v|exp)`	die Gesamtheit, die entsteht, wenn der Audruck *exp* für jedes Element in M ausgewertet wird.
`intersection(N)`	der Durchschnitt von M und N.
`union(N)`	die Vereinigung von M und N.
`includes(o)`	wahr genau dann, wenn o ein Element in M ist.
`includesAll(N)`	wahr genau dann, wenn jedes Element n der Gesamtheit N auch ein Element in M ist.
`isEmpty()`	wahr genau dann, wenn M kein Element enthält.
`exists(v|b)`	wahr genau dann, wenn es ein Element v in M gibt, so dass dafür der boolesche Ausdruck b zu wahr auswertet.
`forAll(v|b)`	wahr genau dann, wenn für jedes Element v in M der boolesche Ausdruck b zu wahr auswertet.
`select(v|b)`	die Gesamtheit der Elemente v von M, die b erfüllen.

6 Object Constraint Language (3+3+3 = 9 Punkte)

Auf der linken Seite ist ein UML-Klassendiagramm abgebildet. Es modelliert den Zusammenhang zwischen Vorlesungen, Studenten und Klausuren: Ein Student besucht Vorlesungen; jeder Vorlesung sind Klausuren zugeordnet; und jeder Student kann zu Klausuren angemeldet sein und kann Klausuren bestanden haben.

a. Geben Sie eine OCL-Invariante für die Klasse Student an, die folgenden Sachverhalt formalisiert:

> Man kann nur dann zu einer Klausur einer Vorlesung angemeldet sein, wenn man für dieselbe Vorlesung nicht bereits eine Klausur bestanden hat.

Die folgenden beiden Invarianten sind alternative Formulierungen des Sachverhalts:

```
context Student
inv: bestanden.vorlesung->intersection(angemeldet.vorlesung)->isEmpty()
inv: angemeldet.vorlesung->forall(v | not bestanden->includes(v))
```

b. Geben Sie die Bedeutung der folgenden OCL-Invarianten in natürlicher Sprache wieder.

```
context Student
   inv: angemeldet.vorlesung->forAll(v | self.vorlesung->includes(v))
```

Ein Student kann nur für Klausuren angemeldet sein, zu denen er auch die zugehörige Vorlesung besucht

c. Die Klasse Student enthält die Methode anmelden(k : Klausur). Geben Sie für diese einen Methodenvertrag an, der besagt:

> Ein Student darf sich für die Klausur k nur anmelden, wenn er für die zu k gehörige Vorlesung noch keine Klausur bestanden hat.
> Nach der ausgeführten Anmeldung ist er für k angemeldet, und der Anmeldestatus für alle anderen Klausuren bleibt, wie er vor dem Vorgang war.

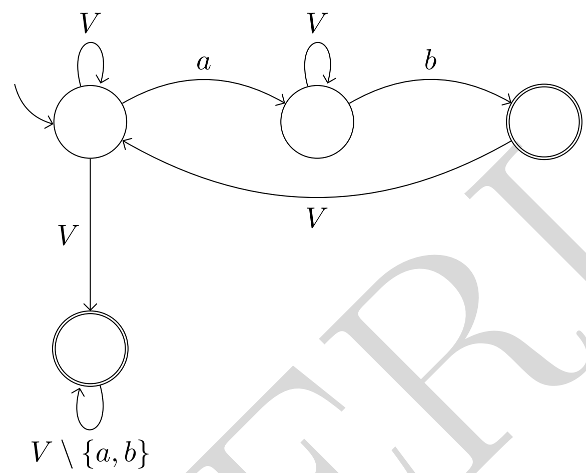
```
context Student::anmelden(k : Klausur)
pre: not bestanden.vorlesung->includes(k.vorlesung)
post: angemeldet = angemeldet@pre->including(k)
```

7 Büchi-Automaten / LTL (5+2 = 7 Punkte)

a. Gegeben sei ein endliches Alphabet V, das wenigstens die beiden Symbole a und b enthält.

Geben Sie einen Büchi-Automaten $\mathcal{B}$ an, so dass die von $\mathcal{B}$ akzeptierte Sprache $L^\omega(\mathcal{B})$ genau diejenigen Wörter $w \in V^\omega$ enthält, für die gilt:

a kommt unendlich oft in w vor, **genau dann** wenn b unendlich oft in w vorkommt.



b. Gegeben sei die Signatur $\Sigma = \{p, q\}$.

Geben Sie einen Büchiautomaten über $V = \mathbb{P}(\Sigma)$ an, der eine ω-Struktur genau dann akzeptiert, wenn sie die folgende LTL-Formel erfüllt:

$\mathbf{X}(p\,\mathbf{U}\,q)$

Verwenden Sie die übliche Mengenschreibweise aus der Vorlesung.