Klausur Formale Systeme

Fakultät für Informatik

WS 2010/2011

Prof. Dr. Bernhard Beckert

17. Februar 2011

Name: **Name**

Matrikel-Nr.: **Matrikelnummer**

Gruppe: **Gruppe**

Platz: **Platz**

Klausur-ID: **Id**

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (13)	A2 (4)	A3 (7)	A4 (12)	A5 (7)	A6 (11)	A7 (6)	Σ (60)

Bewertungstabelle bitte frei lassen!

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

Bonus: ______ **Gesamtpunkte:**

1 Zur Einstimmung (4+3+6 = 13 Punkte)

a. Bitte kreuzen Sie in den folgenden Tabellen die für die Formeln in der jeweiligen Logik zutreffende Eigenschaft an. Für korrekte Antworten erhalten Sie einen Punkt, für falsche Antworten wird ein Punkt abgezogen. Dabei werden jedoch nie weniger als 0 Punkte pro Tabelle vergeben.

PL1 (Prädikatenlogik 1. Stufe)	keine Formel der PL1	erfüllbar (aber nicht allgemeing.)	allgemeingültig (und erfüllbar)	unerfüllbar
$\forall x \exists x\ p(x)$		X		
$p(a \vee a) \leftrightarrow p(\mathbf{1})$	X			
$\forall x \exists y(p(x) \to q(y)) \ \wedge\ \neg\big((\exists x\ p(x)) \to (\exists y\ q(y))\big)$				X
$\forall x \forall y(p(x,y) \leftrightarrow \neg p(y,x))$				X

p, q sind Prädikatensymbole, a ist ein Konstantensymbol, die übrigen Bezeichner sind Variablen.

LTL (Lineare Temporale Logik)	erfüllbar (aber nicht allgemeing.)	allgemeingültig (und erfüllbar)	unerfüllbar
$\Diamond B \wedge \Box\neg(A\ \mathbf{U}\ B)$			X
$\Box((A \wedge \neg C \wedge \Diamond C) \to (B\ \mathbf{U}\ C))$	X		
$A\ \mathbf{V}\ \mathbf{1}$		X	

A, B, C sind aussagenlogische Variablen.

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.

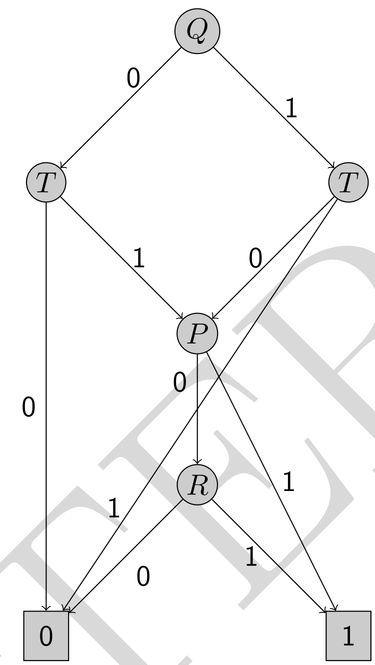
	Richtig	Falsch
Sei M eine Menge prädikatenlogischer Formeln. Gibt es eine Formel ϕ, so dass $M \models \phi$ und $M \models \neg\phi$ gilt, so hat M kein Modell.	X	
Aus der Unentscheidbarkeit der Prädikatenlogik erster Stufe folgt: Es gibt eine allgemeingültige Formel F, deren Allgemeingültigkeit sich mit keinem Kalkül beweisen lässt.		X
Zu jedem Büchi-Automaten $\mathcal{A}$ gibt es eine LTL-Formel $B_{\mathcal{A}}$ mit $L^\omega(\mathcal{A}) = \{\xi \in V^\omega \mid \xi \models B_{\mathcal{A}}\}$		X
Es gibt eine prädikatenlogische Formel F, deren Skolemnormalform äquivalent zu F ist.	X	
Die transitive Hülle einer wohlfundierten Relation ist wieder wohlfundiert.	X	
Wenn $A \to \Box\Diamond A$ in einer Kripke-Struktur (S, R, I) falsch ist, dann kann R nicht symmetrisch sein.	X	

2 Shannon-Formeln (4 Punkte)

Gegeben sei folgende aussagenlogische Formel:

$$\neg(Q \leftrightarrow T) \land (P \lor R)$$

Zeichnen Sie den vollständig reduzierten Shannongraphen für diese Formel.
Verwenden Sie dazu die Variablenordnung: $Q < T < P < R$.



3 Unifikation/Resolution (7 Punkte)

Seien

- p ein einstelliges Prädikatensymbol,
- q ein zweistelliges Prädikatensymbol,
- r ein dreistelliges Prädikatensymbol,
- f ein einstelliges Funktionssymbol,
- a, b, c, d Konstanten,
- w, x, z Variablen.

Geben Sie für jedes der folgenden sechs Klauselpaare genau die Klauseln an, die sich daraus in **einem** Resolutionsschritt ableiten lassen. Tragen Sie diese in die folgende Tabelle ein. Es kann Klauselpaare geben, bei denen mehr als eine Resolvente einzutragen ist.

Wenn ein Paar von Klauseln keine Resolvente hat, so markieren Sie dies entsprechend in der Tabelle durch ein Kreuz (und nicht dadurch, dass Sie das Feld der resultierenden Resolventen frei lassen).

	nicht resolvierbar
1. $\{p(x),\ p(c)\}, \quad \{\neg p(c)\}$	
Resolventen: $\boxed{\{p(x)\} \text{ und } \Box\text{, sowie } \{p(c)\}}$	☐
2. $\{p(f(x)),\ \neg p(f(x))\}, \quad \{p(f(x)),\ \neg p(f(x))\}$	
Resolventen: $\boxed{\{p(f(x)),\ \neg p(f(x))\}}$	☐
3. $\{\neg r(x,x,c),\ p(x)\}, \quad \{r(f(w),f(z),w),\ q(w,z)\}$	
Resolventen: $\boxed{\{p(f(c)), q(c,c)\}}$	☐
4. $\{\neg r(x,x,c),\ p(x)\}, \quad \{r(f(w),f(d),w),\ q(w,z)\}$	
Resolventen:	X
5. $\{p(f(b))\}, \quad \{\neg p(a)\}$	
Resolventen:	X
6. $\{p(x),\ q(f(x),a)\}, \quad \{\neg q(z,w),\ \neg p(f(z))\}$	
Resolventen: $\boxed{\{q(f(f(z)),a),\ \neg q(z,w)\} \text{ und } \{p(x),\ \neg p(f(f(x)))\}}$	☐

4 Tableaukalkül (12 Punkte)

Vervollständigen Sie das folgende Tableau, bis es geschlossen ist. Notieren Sie dabei:

- bei jeder Erweiterung, durch welche Regelanwendung eine Formel auf dem Tableau entstanden ist,
- bei Abschlüssen die beiden Partner,
- die schließende Substitution.

$$0\,\big(\forall x\,\forall y\,(p(a,a) \vee \neg p(x,y))\big) \to \big(\exists w\,p(w,w) \vee \forall z\,\exists x\,\forall y\,(\neg p(x,y) \wedge \neg p(x,z))\big)\ _{1}$$

$$1\,\forall x\,\forall y\,(p(a,a) \vee \neg p(x,y))\ _{2[\alpha(1)]}$$

$$0\,(\exists w\,p(w,w) \vee \forall z\,\exists x\,\forall y\,(\neg p(x,y) \wedge \neg p(x,z)))\ _{3[\alpha(1)]}$$

$$0\,\exists w\,p(w,w)\ _{4[\alpha(3)]}$$

$$0\,\forall z\,\exists x\,\forall y\,(\neg p(x,y) \wedge \neg p(x,z))\ _{5[\alpha(3)]}$$

$$0\,\exists x\,\forall y\,(\neg p(x,y) \wedge \neg p(x,sk_1))\ _{6[\delta(5)]}$$

$$0\,\forall y\,(\neg p(X_1,y) \wedge \neg p(X_1,sk_1))\ _{7[\gamma(6)]}$$

$$0\,(\neg p(X_1,sk_2(X_1)) \wedge \neg p(X_1,sk_1))\ _{8[\delta(7)]}$$

Linker Zweig:

$$0\,\neg p(X_1,sk_2(X_1))\ _{9[\beta(8)]}$$

$$1\,p(X_1,sk_2(X_1))\ _{11[\alpha(9)]}$$

$$1\,\forall y\,(p(a,a) \vee \neg p(X_3,y))\ _{14[\gamma(2)]}$$

$$1\,(p(a,a) \vee \neg p(X_3,X_4))\ _{15[\gamma(14)]}$$

Linker Unterzweig:

$$1\,p(a,a)\ _{16[\beta(15)]}$$

$$0\,p(X_5,X_5)\ _{18[\gamma(4)]}$$

$$\boxed{\star[16,18]}$$

Rechter Unterzweig:

$$1\,\neg p(X_3,X_4)\ _{17[\beta(15)]}$$

$$0\,p(X_3,X_4)\ _{19[\alpha(17)]}$$

$$\boxed{\star[11,19]}$$

Rechter Zweig:

$$0\,\neg p(X_1,sk_1)\ _{10[\beta(8)]}$$

$$1\,p(X_1,sk_1)\ _{12[\alpha(10)]}$$

$$0\,p(X_2,X_2)\ _{13[\gamma(4)]}$$

$$\boxed{\star[12,10]}$$

Die schließende Substitution ist: $\{X_1/sk_1, X_2/sk_1, X_3/sk_1, X_4/sk_2(sk_1), X_5/a\}$.

5 Formalisieren in Prädikatenlogik 2. Stufe (2+2+3 = 7 Punkte)

a. Formulieren Sie die folgenden Aussagen in Prädikatenlogik 2. Stufe. Formalisieren Sie dabei „Eigenschaften" als Mengen (nämlich die Menge der Elemente, die die jeweilige Eigenschaft besitzen).

- Für jede Eigenschaft gibt es mindestens zwei Elemente, für die sie gilt.

$$\boxed{\forall X \exists y \exists z (X(y) \wedge X(z) \wedge \neg(y \doteq z))}$$

- Jedes Element hat eine Eigenschaft, die es mit mindestens einem anderen Element teilt.

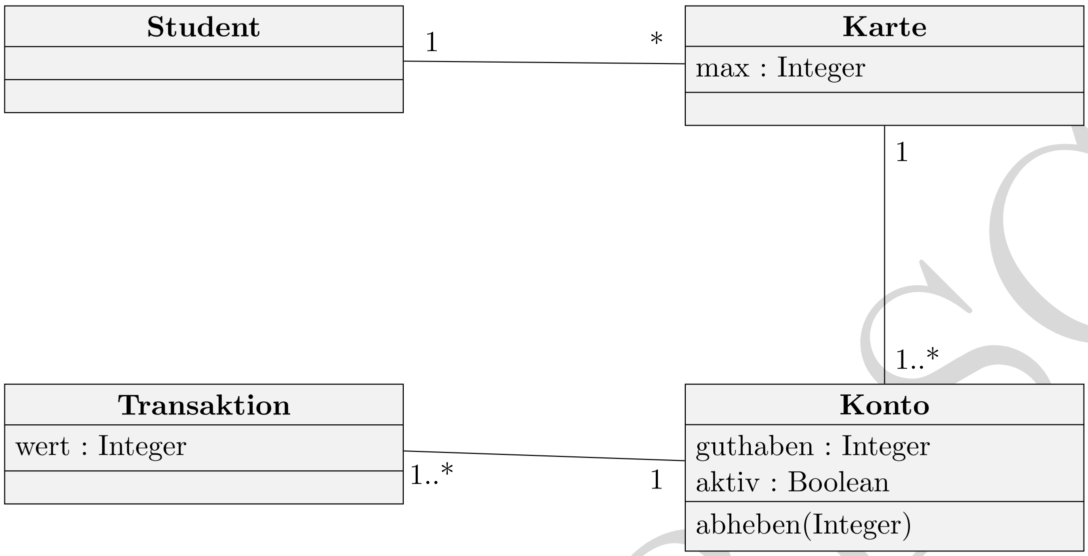
$$\boxed{\forall x \exists Y \exists z (Y(x) \wedge Y(z) \wedge \neg(x \doteq z))}$$

b. Gegeben sei die prädikatenlogische Signatur $\Sigma = (\{\}, \{E\}, \alpha)$ mit $\alpha(E) = 2$. Jedes Modell (V, I) zu dieser Signatur kann als gerichteter Graph $(V, I(E))$ aufgefasst werden. Das Universum V ist dabei die Menge der Knoten und das Prädikat $I(E)$ die Menge der Kanten.

- Geben Sie eine Formel ϕ der Prädikatenlogik 2. Stufe (PL2) an, so dass $val_{V,I,\beta}(\phi) = W$ gdw. es im Graphen $(V, I(E))$ eine Clique gibt. (Eine Clique ist eine Teilmenge der Knoten eines Graphen, wobei je zwei beliebige, verschiedene Knoten dieser Teilmenge miteinander verbunden sind.)

$$\boxed{\exists M \forall x \forall y \ \ (M(x) \wedge M(y) \wedge \neg x \doteq y \rightarrow E(x, y))}$$

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.
`iterate(v ; init | exp)`	Für jedes Element v in M wird *exp* ausgewertet unter Verwendung des in *init* initialisierten Akkumulators.

6 Object Constraint Language (1+2+3+5 = 11 Punkte)

Auf der linken Seite ist ein UML-Klassendiagramm abgebildet. Es modelliert den Zusammenhang zwischen Studenten, Karten, Konten und Transaktionen: Ein Student besitzt eine Menge von Karten, die jeweils mit beliebig vielen Konten verknüpft sein können. Jedes Konto gehört dabei zu genau einer Karte. Jedem Konto sind eindeutig eine Menge von Transaktionen zugeordnet, die alle Ein- bzw. Auszahlvorgänge des Kontos festhalten.

Vervollständigen Sie die folgenden OCL-Constraints gemäß der angegebenen natürlichsprachlichen Bedeutung.

a. Das Guthaben jedes inaktiven Kontos muss 0 betragen.

```
context Konto
inv: not aktiv implies guthaben = 0
```

b. Jeder Student besitzt maximal ein Konto mit negativem Guthaben.

```
context Student
inv: karte.konto->select(wert < 0)->asSet()->size() < 2
```

c. Der Wert `max` einer Karte entspricht dem höchsten Guthaben aller Konten der Karte. Verwenden Sie hierbei die Funktion `iterate`, um dieses höchste Guthaben zu bestimmen. Sie können davon ausgehen, dass alle Guthaben der Konten nicht negativ sind.

```
context Konto
inv:
max = self.konto->iterate(k:Konto ; m:Integer=0 |
      if k.guthaben > m then k.guthaben else m endif)
```

d. Die Methode `abheben(betrag : Integer)` ist wie folgt spezifiziert:

Wenn zu Beginn der Methode gilt:

- Das Guthaben des Kontos ist größer als der abzuhebende Betrag.

Dann gilt nach Ausführung der Methode:

- Das Guthaben des Kontos ist um `betrag` verringert.
- Es existiert nach Ausführung dieser Methode eine Transaktion des Kontos mit entsprechendem Wert.
- Die Menge der Transaktionen des Kontos ist um eins größer als zu Beginn der Ausführung der Methode.

Dass die „erzeugte“ Transaktion nicht bereits zu Beginn vorhanden war, müssen Sie hier nicht spezifizieren.

```
context Konto::abheben(betrag: Integer)
pre:
```

```
self.guthaben > betrag
```

```
post:
```

```
self.guthaben = self.guthaben@pre - betrag
and self.transaktion->exists(wert = betrag)
and self.transaktion->size() = self.transaktion@pre->size() + 1
```

7 Beweis Aussagenlogik (6 Punkte)

Gegeben seien die folgenden zwei Tableauregeln eines aussagenlogischen Tableaukalküls. Dabei sind A und B jeweils aussagenlogische Formeln. Aus der Vorlesung ist Ihnen die (korrekte) Regel R1 bereits bekannt. Die Regel R2 ist im Gegensatz zu R1 *nicht* korrekt.

R1:

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

R2:

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

Zeigen Sie durch Angabe eines Gegenbeispiels, dass die Regel R2 nicht korrekt ist. Geben Sie dazu eine **nicht unerfüllbare**, aussagenlogische Formel F an, zu der sich unter Verwendung der Regel R2 trotzdem ein geschlossenes Tableau mit der Startmarkierung **1 F** konstruieren lässt. Geben Sie das entsprechende Tableau an.