Klausur Formale Systeme

Fakultät für Informatik

WS 2009/2010

Prof. Dr. Bernhard Beckert

08. April 2010

Name: Mustermann
Vorname: Peter
Matrikel-Nr.: 0000000

Klausur-ID: 0000

A1 (15)	A2 (9)	A3 (7)	A4 (10)	A5 (11)	A6 (9)	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+2 = 15 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.

Prädikatenlogik 1. Stufe	keine Formel der PL1	erfüllbar (aber nicht allgemeing.)	allgemeingültig (und erfüllbar)	unerfüllbar
$\exists x \exists c\,(p(c) \to p(x))$	X			
$\big(p(c) \to \exists x\, p(x)\big) \leftrightarrow \exists x\,(p(c) \to p(x))$			X	
$\big((\exists x\, p(x)) \to p(c)\big) \leftrightarrow \exists x\,(p(x) \to p(c))$		X		
$\forall x\, p(x) \wedge \exists y\,(p(y) \to \mathbf{0})$				X

***p* ist ein Prädikatensymbol,**
***c* ist ein Konstantensymbol,**
***x*,*y* sind Variablen.**

Modallogik	erfüllbar (aber nicht allgemeing.)	allgemeingültig (und erfüllbar)	unerfüllbar
$\Diamond A \to \Box A$	X		
$\Box\mathbf{0} \wedge \Diamond A$			X
$(\Box A \wedge \Diamond\mathbf{1}) \to \Diamond A$		X	

***A* ist eine aussagenlogische Variable.**

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 von Hornformeln. Falls M kein Faktum enthält, dann ist M erfüllbar. Dabei ist ein Faktum eine Hornformel, die nur aus einem positiven Literal besteht und kein negatives Literal enthält.	X	
Das Weglassen von Regeln aus einem logischen Kalkül erhält seine Korrektheit.	X	
Sei $(D, \succ)$ ein beliebiges noethersche Reduktionssystem und sei $s \in D$ beliebig. Dann gilt: Es kann nur endlich viele verschiedene Elemente $s' \in D$ geben, so dass $s \succ s'$ gilt.		X
Es gibt Formeln ϕ der Prädikatenlogik erster Stufe, für die gilt: Alle Modelle von ϕ haben unendliche Universen.	X	
Seien s und t Terme. Wenn s mit t unifizierbar ist, und s' ein echter Unterterm von s, dann ist s' niemals mit t unifizierbar.		X
Sei `m()` eine Methode, und seien (P, Q) und (P, Q') zwei OCL-Vor-/Nachbedingungspaare, so dass `m()` sowohl (P, Q) als auch (P, Q') erfüllt. Dann erfüllt `m()` auch $(P, (Q \texttt{ and } Q'))$.	X	

c. Gegeben sei die prädikatenlogische Signatur Σ, die als einzige Symbole

- das Konstantensymbol c und
- das einstellige Prädikatensymbol p enthält.

Geben Sie eine **erfüllbare** Formel über dieser Signatur Σ an, die kein Herbrandmodell über Σ besitzt.

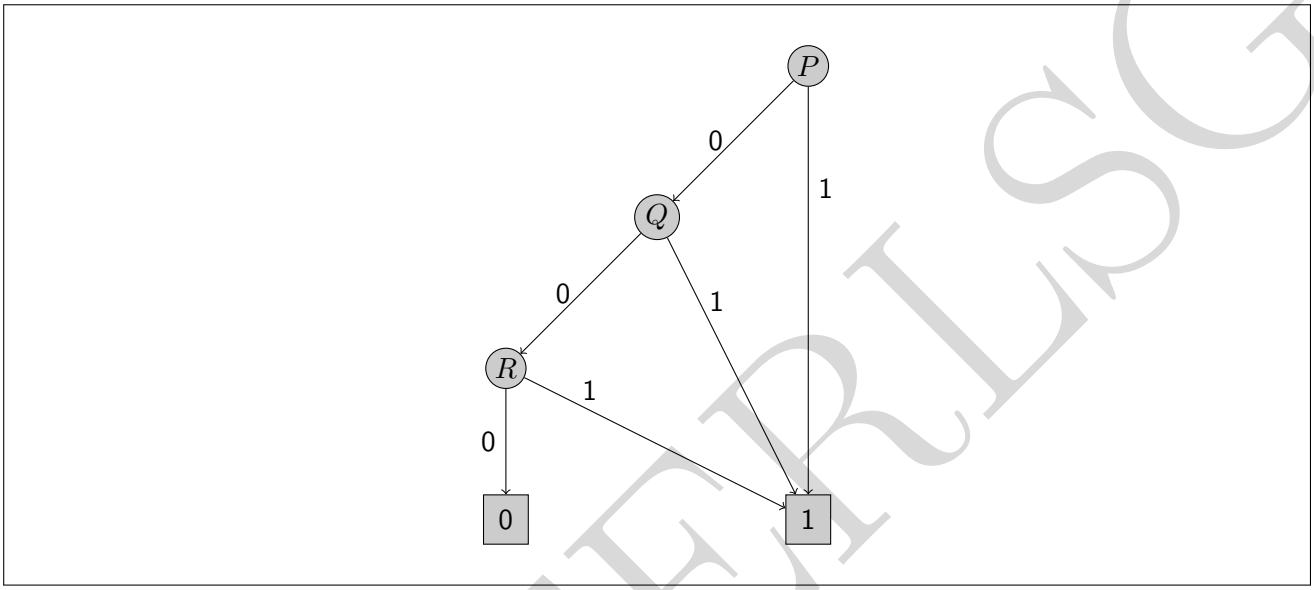
$\exists x\, p(x)$

2 Shannongraphen (5+2+2 Punkte)

a. Zeichnen Sie einen vollständig reduzierten Shannongraphen für die Formel

$$\neg P \to (\neg Q \to R) \ .$$

Verwenden Sie dabei die Variablenordnung $P < Q < R$.



b. Die Variablenordnung sei im folgenden beliebig aber fest. Für eine beliebige Formel A der Aussagenlogik bezeichne

- $shannon(A)$ den vollständig reduzierten Shannongraphen für A,
- $\#(shannon(A))$ die Anzahl der Knoten in diesem Graphen.

Begründen Sie kurz, warum für jede Formel A gilt:

$$\#(shannon(A)) \ = \ \#(shannon(\neg A))$$

Vertausche den Knoten 0 mit dem Knoten 1. Der so entstandene Graph sei mit $shannon'(A)$ bezeichnet. Dann gilt:

- $shannon'(A)$ ist genauso wie $shannon(A)$ vollständig reduziert, und
- $shannon'(A) = shannon(\neg A)$.

c. Seien A und B beliebige Formeln der Aussagenlogik. Gilt die Aussage

$$\#(shannon(A \wedge B)) \ = \ \#(shannon(A)) + \#(shannon(B)) \ ?$$

Begründen Sie Ihre Antwort kurz.

Die Aussage ist falsch, wie folgendes Gegenbeispiel zeigt.
Sei A beliebig und $B = \neg A$. Dann gilt unabhänig von der Wahl von A:

$$\#(shannon(A \wedge \neg A)) = 1 \ \text{ und } \ \#(shannon(A)) + \#(shannon(B)) = 2 * \#(shannon(A)) > 1$$

3 Formalisieren in Prädikatenlogik (7 Punkte)

Formalisieren Sie die vier folgenden Aussagen in Prädikatenlogik erster Stufe mit Gleichheit. Benutzen Sie jeweils die angegebenen Prädikatsymbole. Sie können davon ausgehen, dass alle Formeln mit der Menge der Spieler als Universum interpretiert werden.

Vor einem Länderspiel erklärt Jogi Löw der Presse die Taktik und die Stimmung in seiner Mannschaft:

- Jeder Stürmer wird eingesetzt.
  Prädikate: $st(\cdot)$, $e(\cdot)$

$\boxed{\forall x \left(st(x) \rightarrow e(x)\right)}$

- Alle eingesetzten Spieler haben nichts gegeneinander.
  Prädikate: $e(\cdot)$, $g(\cdot,\cdot)$

$\boxed{\forall x \forall y \left((e(x) \wedge e(y)) \rightarrow \neg g(x,y)\right)}$

- Jeder Spieler hat etwas gegen irgendeinen anderen Spieler.
  Prädikate: $g(\cdot,\cdot)$

$\boxed{\forall x \exists y \left(\neg(x \doteq y) \wedge g(x,y)\right)}$

Ein Journalist folgert:

- Jeder Stürmer hat etwas gegen irgendeinen Nicht-Stürmer.
  Prädikate: $st(\cdot)$, $g(\cdot,\cdot)$

$\boxed{\forall x \left(st(x) \rightarrow \exists y \left(\neg st(y) \wedge g(x,y)\right)\right)}$

---

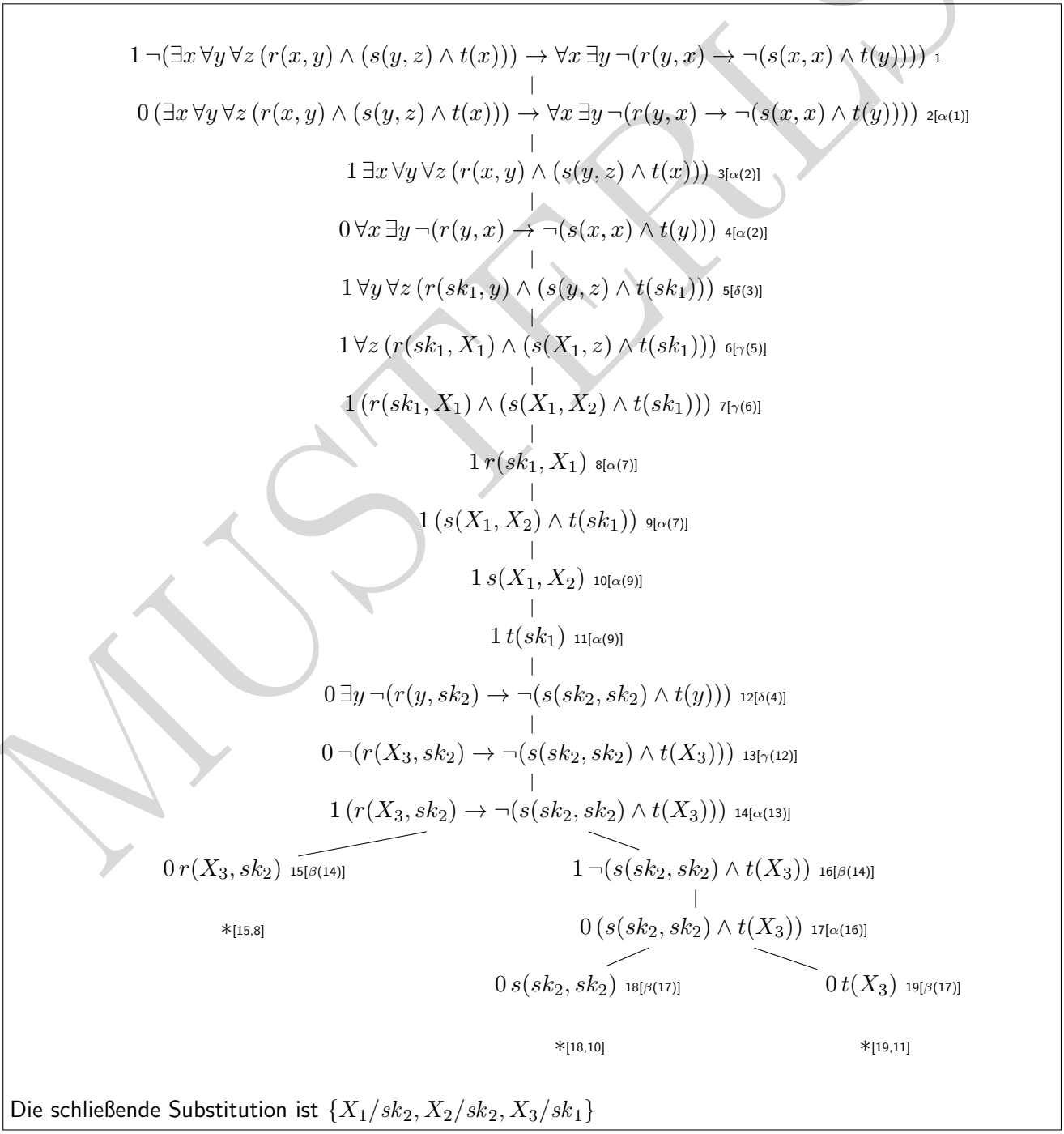
4 Tableaukalkül (10 Punkte)

Beweisen Sie mit Hilfe des Tableaukalküls, dass die folgende prädikatenlogische Formel **unerfüllbar** ist:

$$\neg\big(\big(\exists x\,\forall y\,\forall z\,(r(x,y)\wedge s(y,z)\wedge t(x))\big)\rightarrow\big(\forall x\,\exists y\,\neg(r(y,x)\rightarrow\neg(s(x,x)\wedge t(y)))\big)\big);$$

Notieren Sie dabei:

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



Die schließende Substitution ist $\{X_1/sk_2, X_2/sk_2, X_3/sk_1\}$

5 Formalisieren in Temporallogik (3+4+4 Punkte)

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

a. Die Kaffeemaschine bleibt ausgeschaltet (*off*) bis der Einschaltknopf gedrückt ist (*pressed*). Es ist auch möglich, dass der Einschaltknopf niemals gedrückt wird.

$off\ \mathbf{U_W}\ pressed \equiv (off\ \mathbf{U}\ pressed) \vee \Box off$

b. Immer wenn die Kaffeemaschine eingeschaltet ist (*on*), und ein mechanisches Problem mit der Maschine auftritt (*problem*), erscheint irgendwann danach eine Fehlermeldung (*error*).

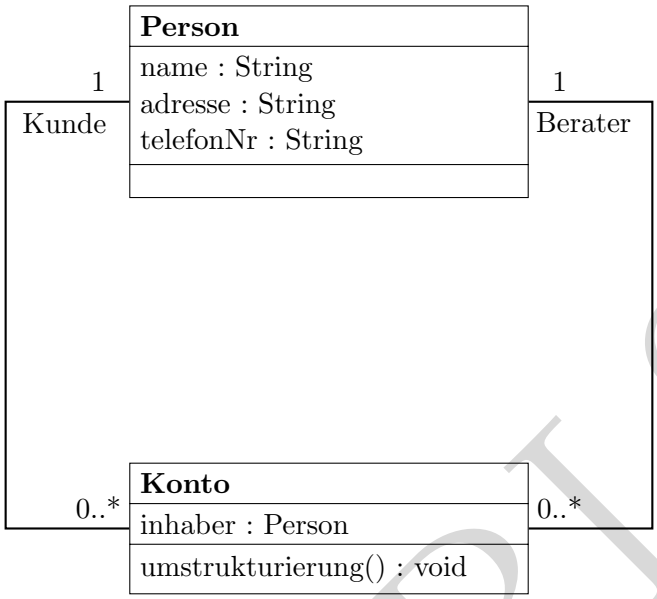
$\Box((on \wedge problem) \rightarrow \mathbf{X}\Diamond error)$
Eine Lösung ohne den **X**-Operator gilt auch als richtig.

c. Zwischen jedem Drücken des Einschaltknopfes (*pressed*) und dem Erscheinen einer Fehlermeldung (*error*) muss ein mechanisches Problem (*problem*) vorgelegen haben.

Hinweis: Betrachten Sie das Negat der Aussage.

$\neg\Diamond(pressed \wedge \mathbf{X}(\neg problem\ \mathbf{U}\ error)) \equiv$
$\Box(pressed \rightarrow \mathbf{X}\neg(\neg problem\ \mathbf{U}\ error)) \equiv$
$\Box(pressed \rightarrow \mathbf{X}(problem\ \mathbf{V}\ \neg error))$
Eine Lösung ohne den **X**-Operator gilt auch als richtig.

UML-Diagramm zu Aufgabe ??



Dieses UML-Diagramm modelliert eine Bank. Die Kunden haben Konten bei der Bank, während die Bank zu jedem Konto einen Bankangestellten als Berater stellt. Die Berater können selbst ebenfalls Konten bei der Bank haben.

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.
`count(o)`	die Anzahl der Vorkommen von o in M.
`sum()`	die Summe der Elemente einer Menge von Zahlen.
`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.
`select(v|b)`	die Gesamtheit der Elemente v von M, die b erfüllen.
`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.

Der folgende Operator ist auf Klassennamen anwendbar

`allInstances()`	der Ausdruck `Class.allInstances()` liefert alle Objekte in der Klasse `Class`.

6 OCL (3+3+3 Punkte)

Vervollständigen Sie mit Bezug auf das links abgebildete UML-Klassendiagramm die folgenden OCL-Constraints gemäß der jeweils angegebenen natürlichsprachlichen Bedeutung.

a. Wenn eine Person mit einem Konto assoziiert ist, dann referenziert das Attribut `inhaber` des Kontos genau diese Person.

```
context Person
inv:
```

```
self.konto->forAll(k| k.inhaber=self)
```

b. Der Berater zu einem Konto darf nicht gleichzeitig der Kunde sein, dem das Konto gehört.

```
context Konto
inv:
```

```
kunde<>berater
```

c. Nach einer Umstrukturierung der Bank hat sich die Zuordnung der Berater zu Konten geändert. Ebenfalls haben sich die Telefonnummern der Berater geändert. Dennoch bleibt zu jedem Konto die Telefonnummer des zugeordneten Beraters die gleiche.

Zur Verdeutlichung: Wenn man mit dem für ein Konto zuständigen Berater sprechen will, kann man also dieselbe Telefonnummer anrufen, die man zu diesem Zweck vor der Umstrukturierung angerufen hätte.

Wie lautet die entsprechende Nachbedingung der Methode `umstrukturierung()`?

Hinweis: Benutzen Sie `@pre`.

```
context Konto::umstrukturierung() : void
post:
```

```
berater.telefonNr=berater@pre.telefonNr@pre
```