Klausur Formale Systeme

Universität Karlsruhe

Fakultät für Informatik

SS 1999/2000

Prof. Dr. P. H. Schmitt

9. Mai 2000

Name: ______________________________

Vorname: ______________________________

Matrikel-Nr.: ______________________________

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

A1 (4)	A2 (4)	A3 (10)	A4 (6)	A5 (10)	A6 (10)	A7 (6)	A8 (10)	Σ (60)

Bewertungstabelle bitte frei lassen !!!

Zum Bestehen der Klausur benötigen Sie 20 der erreichbaren 60 Punkte.

Gesamtpunkte:

1 Zur Einstimmung (2 + 2 Punkte)

Kreuzen Sie in den folgenden Tabellen alles Zutreffende an.

Für jede falsche Antwort wird ein halber Punkt abgezogen!
(Dabei werden insgesamt jedoch keinesfalls weniger als 0 Punkte für diese Aufgabe vergeben.)

Hinweise:

- „PL1" steht für „Prädikatenlogik erster Ordnung"; auf diese beziehen sich auch die Begriffe „erfüllbar", „allgemeingültig" und „unerfüllbar".
- In Teilaufgabe a. kann eine Formel mehr als eine der genannten Eigenschaften haben.
- c is eine Konstante.

a.

	<u>keine</u> Formel der PL1	erfüllbar	allgemein-gültig	uner-füllbar
$\exists n \exists x (f^n(x) \to f(x))$	$\times$			
$p(c) \to \forall x (p(x))$		$\times$		
$p(c) \to \exists x (p(x))$		$\times$	$\times$	
$\exists x (p(x) \leftrightarrow (q(x) \leftrightarrow (p(x) \leftrightarrow \neg q(x))))$				$\times$

b.

	richtig	falsch
Für jede PL1-Formel ϕ gilt: ϕ ist allgemeingültig oder das Negat von ϕ ist erfüllbar.	$\times$	
Wenn eine PL1-Formel erfüllbar ist, dann ist jede ihrer Teilformeln erfüllbar.		$\times$
Es gibt eine PL1-Interpretation, in der alle PL1-Formeln wahr sind.		$\times$
Für alle Terme s, t_1, t_2 gilt: Wenn s sowohl mit t_1 als auch mit t_2 unifizierbar ist, dann sind auch t_1 und t_2 miteinander unifizierbar.		$\times$

2 Semantik der Prädikatenlogik (2 + 2 Punkte)

Per Definition ist das Universum D prädikatenlogischer Interpretationen (D, I) nicht leer.

a. Geben Sie eine unerfüllbare Formel F an, die erfüllbar wäre, wenn man die Definition ändern und das leere Universum zulassen würde.

Lösung:

$\forall x(p \land \neg p)$

b. Geben Sie eine unerfüllbare Formel G an, die auch dann unerfüllbar bliebe, wenn man die Definition ändern und das leere Universum zulassen würde.

Lösung:

$\exists x(p \land \neg p)$

3 Tableaukalkül (10 Punkte)

Zeigen Sie mit Hilfe des Tableaukalküls, daß die Formel

$$p(a) \ \wedge\ \neg q(d) \ \wedge\ \forall x[((p(b) \wedge p(c)) \rightarrow p(x)) \ \rightarrow\ \neg(q(x) \rightarrow (q(b) \vee q(c)))]$$

unerfüllbar ist. Vervollständigen Sie dazu den folgenden Tableaubeweis:

Hinweis: a, b, c und d sind Konstanten.

(1;–) $\mathbf{1}\ p(a) \ \wedge\ \neg q(d) \ \wedge\ \forall x[((p(b) \wedge p(c)) \rightarrow p(x)) \ \rightarrow\ \neg(q(x) \rightarrow (q(b) \vee q(c)))]$

(2;1) $\mathbf{1}\ p(a)$

(3;1) $\mathbf{1}\ \neg q(d) \ \wedge\ \forall x[((p(b) \wedge p(c)) \rightarrow p(x)) \ \rightarrow\ \neg(q(x) \rightarrow (q(b) \vee q(c)))]$

(4;3) $\mathbf{1}\ \neg q(d)$

(5;3) $\mathbf{1}\ \forall x[((p(b) \wedge p(c)) \rightarrow p(x)) \ \rightarrow\ \neg(q(x) \rightarrow (q(b) \vee q(c)))]$

(6;4) $\mathbf{0}\ q(d)$

<u>Lösung:</u>

$\vdots$

(7;5) $\mathbf{1}\ ((p(b) \wedge p(c)) \rightarrow p(x_1)) \ \rightarrow\ \neg(q(x_1) \rightarrow (q(b) \vee q(c)))$

Linker Ast:

(8;7) $\mathbf{0}\ (p(b) \wedge p(c)) \rightarrow p(x_1)$

(10;8) $\mathbf{1}\ p(b) \wedge p(c)$

(11;8) $\mathbf{0}\ p(x_1)$

(12;10) $\mathbf{1}\ p(b)$

(13;10) $\mathbf{1}\ p(c)$

Rechter Ast:

(9;7) $\mathbf{1}\ \neg(q(x_1) \rightarrow (q(b) \vee q(c)))$

(14;9) $\mathbf{0}\ q(x_1) \rightarrow (q(b) \vee q(c))$

(15;14) $\mathbf{1}\ q(x_1)$

(16;14) $\mathbf{0}\ q(b) \vee q(c)$

(17;16) $\mathbf{0}\ q(b)$

(18;16) $\mathbf{0}\ q(c)$

Nach Anwendung der Substitution $\{x_1/b\}$ (oder auch der Substitution $\{x_1/c\}$) sind alle Äste dieses Tableaus geschlossen.

Es ist nicht möglich, das Tableau ohne Expansion der Formeln in den Knoten 10 und 16 abzuschließen.

4 Resolution (6 Punkte)

Geben Sie einen Resolutionsbeweis für die Unerfüllbarkeit der Menge folgender Klauseln an, bei dem (einschließlich der leeren Klausel) **höchstens vier Resolventen** abgeleitet werden:

(1) $\{ p(c) \}$
(2) $\{ \neg p(x),\ p(f(x)) \}$
(3) $\{ \neg p(f(f(f(f(c))))) \}$

Hinweis: c ist eine Konstante.

Lösung:

(4) $[2,2]$ $\{ \neg p(x),\ p(f(f(x))) \}$
(5) $[4,4]$ $\{ \neg p(x),\ p(f(f(f(f(x))))) \}$
(6) $[1,5]$ $\{ p(f(f(f(f(c))))) \}$
(7) $[3,6]$ $\Box$

5 Davis-Putnam-Verfahren (10 Punkte)

Die (aussagenlogische) Klauselmenge S bestehe aus den folgenden Klauseln:

(1)	$\{ P, Q, T \}$	(5)	$\{ \neg R, \neg S, T \}$
(2)	$\{ \neg P \}$	(6)	$\{ \neg R, P, \neg T \}$
(3)	$\{ \neg Q, R \}$	(7)	$\{ R, \neg T \}$
(4)	$\{ \neg R, S \}$		

Zeigen Sie mit Hilfe des Davis-Putnam-Verfahrens, daß S unerfüllbar ist.

Lösung:

Wähle die Einerklausel $\{\neg P\}$.
Ergebnis der Vereinfachung von S mit $\{\neg P\}$ ist die Menge S', bestehend aus:

$$\begin{array}{lll} \{ Q, T \} & \{ \neg Q, R \} & \{ \neg R, S \} \\ \{ \neg R, \neg S, T \} & \{ \neg R, \neg T \} & \{ R, \neg T \} \end{array}$$

Die Menge S' enthält keine Einerklauseln. Wir widerlegen $S' \cup \{Q\}$ und $S' \cup \{\neg Q\}$.

Zur Widerlegung von $S' \cup \{Q\}$:
Ergebnis der Vereinfachung von $S' \cup \{Q\}$ mit $\{Q\}$ ist:

$$\{ R \} \quad \{ \neg R, S \} \quad \{ \neg R, \neg S, T \} \quad \{ \neg R, \neg T \} \quad \{ R, \neg T \}$$

Ergebnis der Vereinfachung dieser Klauselmenge mit $\{R\}$ ist:

$$\{ S \} \quad \{ \neg S, T \} \quad \{ \neg T \}$$

Ergebnis der Vereinfachung dieser Klauselmenge mit $\{S\}$ ist:

$$\{ T \} \quad \{ \neg T \}$$

Ergebnis der Vereinfachung dieser Klauselmenge mit $\{T\}$ ist:

$$\square$$

Damit ist, da nun die leere Klausel enthalten ist, die Klauselmenge $S' \cup \{Q\}$ widerlegt.

Zur Widerlegung von $S' \cup \{\neg Q\}$:
Ergebnis der Vereinfachung von $S' \cup \{\neg Q\}$ mit $\{\neg Q\}$ ist:

$$\{ T \} \quad \{ \neg R, S \} \quad \{ \neg R, \neg S, T \} \quad \{ \neg R, \neg T \} \quad \{ R, \neg T \}$$

Ergebnis der Vereinfachung dieser Klauselmenge mit $\{T\}$ ist:

$$\{ \neg R, S \} \quad \{ \neg R \} \quad \{ R \}$$

Ergebnis der Vereinfachung dieser Klauselmenge mit $\{R\}$ ist:

$$\{ S \} \quad \square$$

Damit ist, da nun die leere Klausel enthalten ist, auch die Klauselmenge $S' \cup \{\neg Q\}$ widerlegt.

6 Normalformberechnung (6 + 4 Punkte)

a. Geben Sie die kurze konjunktive Normalform F_{kknf} der Formel

$$F \;=\; (A \vee B) \rightarrow ((A \vee B) \rightarrow C)$$

an.

Lösung:

Wir verwenden die Abkürzungen Q_1, Q_2, Q_3 wie folgt:

$$\underbrace{(\underbrace{A \vee B}_{Q_3}) \rightarrow (\overbrace{(\underbrace{A \vee B}_{Q_3}) \rightarrow C}^{Q_2})}_{Q_1}$$

Erfüllbarkeitsäquivalent zu F ist damit Formel

$$\begin{array}{l}
Q_1 \quad \wedge \\
(Q_1 \leftrightarrow (Q_3 \rightarrow Q_2)) \quad \wedge \\
(Q_2 \leftrightarrow (Q_3 \rightarrow C)) \quad \wedge \\
(Q_3 \leftrightarrow (A \vee B))
\end{array}$$

Nach Auflösen der Äquivalenzen erhält man:

$$\begin{array}{l}
Q_1 \quad \wedge \\
(\neg Q_1 \vee (Q_3 \rightarrow Q_2)) \quad \wedge \\
(Q_1 \vee \neg(Q_3 \rightarrow Q_2)) \quad \wedge \\
(\neg Q_2 \vee (Q_3 \rightarrow C)) \quad \wedge \\
(Q_2 \vee \neg(Q_3 \rightarrow C)) \quad \wedge \\
(\neg Q_3 \vee (A \vee B)) \quad \wedge \\
(Q_3 \vee \neg(A \vee B))
\end{array}$$

Nach Transformation in KNF erhält man schließlich das Ergebnis F_{kknf}:

$$\begin{array}{l}
Q_1 \quad \wedge \\
(\neg Q_1 \vee \neg Q_3 \vee Q_2) \quad \wedge \\
(Q_1 \vee Q_3) \quad \wedge \\
(Q_1 \vee \neg Q_2) \quad \wedge \\
(\neg Q_2 \vee \neg Q_3 \vee C) \quad \wedge \\
(Q_2 \vee Q_3) \quad \wedge \\
(Q_2 \vee \neg C) \quad \wedge \\
(\neg Q_3 \vee A \vee B) \quad \wedge \\
(Q_3 \vee \neg A) \quad \wedge \\
(Q_3 \vee \neg B)
\end{array}$$

b. Transformieren Sie die Formel

$$G \;=\; \forall w \exists x [p(w,x) \;\vee\; \forall y (q(w,x,y) \wedge \exists z (r(y,z)))]$$

in Skolem-Normalform (mit Matrix in konjunktiver Normalform).

Lösung:

Transformation in Pränex-Normalform ergibt:

$$\forall w \exists x \forall y \exists z [p(w,x) \;\vee\; (q(w,x,y) \wedge r(y,z))]$$

Nach Skolemisierung erhält man die Formel:

$$\forall w \forall y [p(w,f(w)) \;\vee\; (q(w,f(w),y) \wedge r(y,g(w,y)))]$$

Schließlich muß noch die Matrix der Formel in konjunktive Normalform gebracht werden. Das Ergebnis ist die Skolem-Normalform von G:

$$\begin{aligned}\forall w \forall y [&(p(w,f(w)) \;\vee\; q(w,f(w),y)) \wedge \\ &(p(w,f(w)) \;\vee\; r(y,g(w,y)))]\end{aligned}$$

7 Unifikation (1 + 1 + 2 + 2 Punkte)

Geben Sie jeweils an, ob die folgenden Terme unifizierbar sind, und geben Sie gegebenenfalls einen allgemeinsten Unifikator an.

Hinweis: a und b sind Konstanten.

a. $p(f(a, y), z)$ und
$p(g(a, x), z)$

Lösung:

Nicht unifizierbar.

b. $p(a, g(b, x))$ und
$p(y, g(z, f(x)))$

Lösung:

Nicht unifizierbar.

c. $p(x, g(b, y))$ und
$p(a, g(z, f(x)))$

Lösung:

Allgemeinster Unifikator: $\{ x/a,\ y/f(a),\ z/b \}$

d. $p(x, y)$ und
$p(f(y), f(x))$

Lösung:

Nicht unifizierbar.

Diese Seite ist absichtlich frei gelassen.

Diese Seite ist absichtlich frei gelassen.

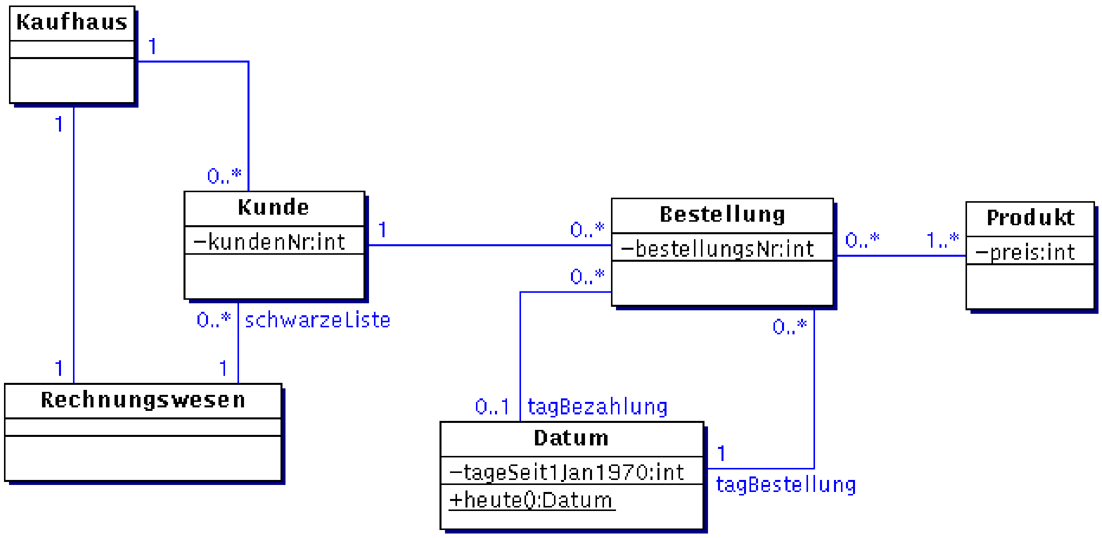

Abbildung zu Aufgabe 8

8 Object Constraint Language ((2 + 2) + (2 + 4) Punkte)

Das links (auf der Rückseite von Blatt 10) dargestellte UML-Klassendiagramm sei gegeben.

a. Geben Sie natürlichsprachliche Übersetzungen der folgenden OCL-Constraints an:

i. Produkt

```
preis > 0  and  preis <= 1000
```

Lösung:

Für jedes Produkt gilt, daß sein Preis größer als Null ist und kleiner oder gleich 1000.

ii. Kunde

```
rechnungswesen = kaufhaus.rechnungswesen
```

Lösung:

Das einem Kunden zugeordnete Rechnungswesen ist das gleiche, wie das Rechnungswesen, das dem Kaufhaus zugeordnet ist, zu dem der Kunde gehört.

b. Geben Sie OCL-Constraints an, die die folgenden Sachverhalte modellieren:

i. Für jedes Kaufhaus gilt: Es gibt keine verschiedenen Kunden mit gleicher Kunden-Nummer.

Lösung:

Kaufhaus

```
kunde->forAll(k1 | kunde->forAll(k2 |
    k1 <> k2 implies k1.kundenNr <> k2.kundenNr))
```

ii. Ein Kunde ist auf der schwarzen Liste des Rechnungswesens, falls er eine Bestellung vor mehr als 30 Tagen aufgegeben hat, die bis heute noch nicht bezahlt ist.

Lösung:

Kunde

```
(bestellung->exists( tagBezahlung->isEmpty() and
                     tagBestellung.tageSeit1Jan1970 + 30 <
                         tagBestellung.heute().tageSeit1Jan1970 )
)  implies
rechnungswesen.schwarzeListe->includes(self)
```