Klausur Formale Systeme

Fakultät für Informatik

WS 2012/2013

Prof. Dr. Peter H. Schmitt

14. Februar 2013

Name: ____________________

Vorname: ____________________

Matrikel-Nr.: ____________________

Bitte merken Sie sich die Nummer Ihrer Klausur, zu finden rechts oben in der Ecke!
Unter dieser Nummer wird Ihr Klausurergebnis veröffentlicht.

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (10)	A2 (5)	A3 (5)	A4 (6)	A5 (9)	A6 (10)	A7 (7)	A8 (8)	Σ (60)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung (5+3 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 jede der zwei 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“.
- p und q sind Prädikatssymbole, c und f sind Funktionssymbole, und x und y sind Variablen.
- Es gelten die üblichen Klammereinsparungsregeln.

	<u>keine</u> Formel der PL1	allgemeingültig	erfüllbar, aber nicht allgemeingültig	unerfüllbar
$(\forall x\ f(x, f(x))) \to (\forall x \exists y\ f(x, y))$	X			
$(q \to \mathbf{1}) \vee (\mathbf{1} \to q)$		X		
$(\forall x \forall y (f(x) \doteq f(y) \to x \doteq y)) \to \exists x\ f(x) \doteq x$			X	
$(\exists x \forall y\ p(f(x), y)) \to (\exists x \forall y\ p(x, f(y)))$		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
Zu jeder aussagenlogischen Formel gibt es unendlich viele logisch äquivalente Formeln.	X	
Zu jeder Substitution σ gibt es eine Substitution ρ, so daß die Hintereinanderausführung $\rho \circ \sigma$ die identische Substitution ist.		X
Zu jeder prädikatenlogischen Formel ϕ gibt es eine Struktur $\mathcal{M}$, in der sie falsch ist, d.h. $\mathcal{M} \not\models \phi$.		X
Die LTL-Formel $\big((\Box\Diamond A) \vee (\Box\Diamond B)\big) \leftrightarrow \Box\Diamond(A \vee B)$ ist allgemeingültig.	X	
Die LTL-Formel $(A\ \mathrm{U}\ B) \to (A \vee B)$ ist allgemeingültig	X	
$(a^*b)^\omega(ab^*)^\omega$ ist ein ω-regulärer Ausdruck		X

2 Erfüllbarkeit (5 Punkte)

Definition. Sei S eine Menge von aussagenlogischen Klauseln. Für ein Literal L sei $\bar{L}$ das zu L *komplementäre* Literal, d.h.

$$\bar{L} = \begin{cases} \neg A & \text{falls } L = A \\ A & \text{falls } L = \neg A \end{cases}$$

Ein Literal L heißt *isoliert* in S, wenn $\bar{L}$ in keiner Klausel in S vorkommt. Eine Klausel C in S heißt *isoliert*, wenn sie ein isoliertes Literal enthält.

Zeigen oder widerlegen Sie: Ist S eine unerfüllbare Klauselmenge und C eine isolierte Klausel in S, dann ist auch $S \setminus \{C\}$ unerfüllbar.

Angenommen es gäbe eine erfüllende Interpretation I für $S \setminus \{C\}$. Sei L ein isoliertes Literal in C. Nach Annahme muß es mindestens ein solches geben. Wir setzen:

$$J(P) = \begin{cases} I(P) & \text{falls } P \text{ nicht im Literal } L \text{ vorkommt} \\ W & \text{falls } L = P \\ F & \text{falls } L = \neg P \end{cases}$$

Offensichtlich gilt $J(C) = W$. Aber gilt auch noch $J(D) = W$ für alle anderen Klauseln D in S? Sicher, da $\bar{L}$ in keiner Klausel in S vorkommt, kann schlimmstenfalls der Wechseln von $I(L) = F$ zu $J(L) = W$ erfolgen. Aus $I(D) = W$ folgt also auch $J(D) = W$.

3 Markierungsalgorithmus für Hornformeln (5 Punkte)

Überprüfen Sie folgende Hornformeln auf Erfüllbarkeit. Benutzen Sie den in der Vorlesung vorgestellten Markierungsalgorithmus. **Unterstreichen** Sie dazu die zu markierenden Literale in der Formel **und** geben Sie unter Schritt n an, **welche(s) Literal(e) im n-ten Schritt markiert** wurde(n). Geben Sie **zudem** ein **Modell** an **oder** benennen Sie die **Hornformel**, aufgrund dessen der Algorithmus mit unerfüllbar abbricht!

a. $(P_1 \land P_2 \to \mathbf{0}) \land P_1 \land (P_1 \land P_3 \to P_2) \land P_3$

Schritt 1:	Schritt 2:	Schritt 3:
P_1, P_3	P_2	

Ergebnis:

Konfliktklausel: $\boxed{P_1} \land \boxed{P_2} \to \mathbf{0} \implies$ unerfüllbar.

b. $(P_1 \to \mathbf{0}) \land (P_2 \land P_3 \to P_4) \land (P_1 \to P_2)$

Schritt 1:	Schritt 2:	Schritt 3:	Schritt 4:

Ergebnis:

Diese Formel enthält keine Fakten $\implies$ Erfüllbar, z.B. durch die Belegung $I_F \equiv F$, die jede Variable zu F auswertet.

c. $(P_3 \land P_4 \to P_5) \land P_3 \land (P_2 \to P_1) \land P_2$

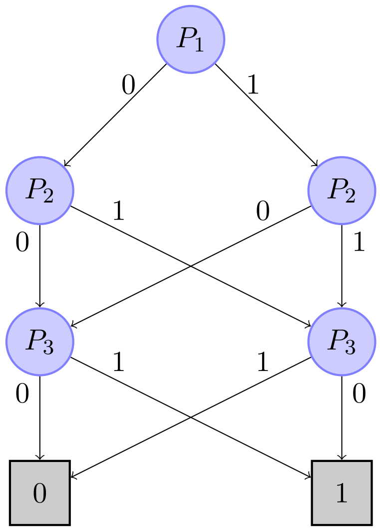
Schritt 1:	Schritt 2:	Schritt 3:	Schritt 4:	Schritt 5:
P_2, P_3	P_1			

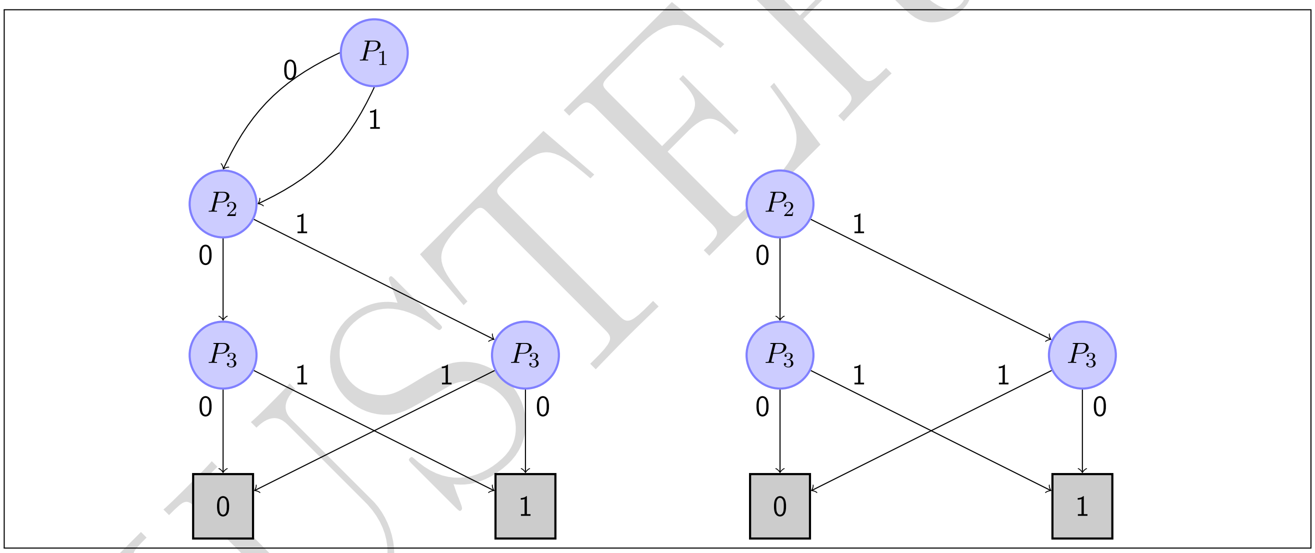
Ergebnis:

Die konstante Interpretation $I_W \equiv W$, die jede Variable zu W auswertet, ist ein Modell der Formel.

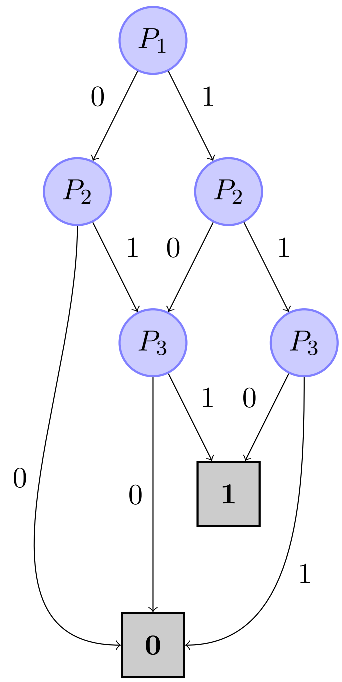
4 Shannongraphen (4 + 2 Punkte)

a. Konstruieren Sie zu dem folgenden Shannnon-Graphen den reduzierten Shannon-Graphen (mit der gleichen Variablen-Ordnung $P_1 < P_2 < P_3$). Geben Sie alle Zwischenschritte an.





b. Geben Sie zu dem folgenden Shannongraphen eine äquivalente aussagenlogische Formel in disjunktiver Normalform an.



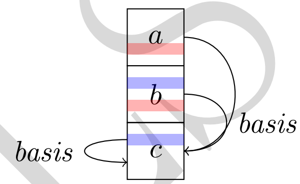
Lösung: $(\neg P_1 \wedge P_2 \wedge P_3) \vee (P_1 \wedge \neg P_2 \wedge P_3) \vee (P_1 \wedge P_2 \wedge \neg P_3)$

5 Formalisieren in Prädikatenlogik (3 + 1,5 + 2 + 2,5 Punkte)

Gegeben sei die prädikatenlogische Signatur, welche genau das einstellige Funktionssymbol $basis(\cdot)$, die einstelligen Prädikatensymbole $rot(\cdot)$ und $blau(\cdot)$, und das zweistellige Prädikatensymbol $auf(\cdot,\cdot)$ enthält.

Ein Universum D bestehen aus endlich vielen Blöcken, die zu Türmen gestapelt werden können. $auf(x,y)$ beschreibt, dass Block x direkt auf Block y liegt. Dabei darf höchstens ein Block direkt auf einem anderen liegen. Die Funktion *basis* ordnet allen Blöcken eines Turmes den Basisblock des Turmes zu, also den untersten Block des Turmes. Die Blöcke sind **farbig gestreift**: $rot(x)$ beschreibt, dass Block x rote Streifen hat, $blau(x)$, dass Block x blaue Streifen hat.

Z.B. kann $U = \{a,b,c\}$ mit $I(rot) = \{a,b\}$, $I(blau) = \{b,c\}$, $I(auf) = \{(a,b),(b,c)\}$ und $I(basis)(x) = c$ für alle $x \in U$ folgendermaßen veranschaulicht werden:



a. Geben Sie eine Formel der Prädikatenlogik erster Stufe an, die genau dann wahr ist, wenn es einen Turm gibt, der genau aus zwei Blöcken besteht.

$\exists x \exists y \big(auf(x,y) \land y \doteq basis(y) \land \neg \exists v\ auf(v,x)\big)$ (1)

b. Geben Sie eine Formel der Prädikatenlogik erster Stufe an, die genau dann wahr ist, wenn es genau einen Turm gibt.

$\forall x \forall y \big(basis(x) = basis(y)\big)$ (2)

c. Geben Sie eine Formel der Prädikatenlogik erster Stufe an, die genau dann wahr ist, wenn jeder Block rot- oder blau- oder rot-blau-gestreift ist und es einen rot-blau-gestreiften Block gibt.

$\forall x \big(rot(x) \lor blau(x)\big) \land \exists x \big(rot(x) \land blau(x)\big)$ (3)

d. Geben Sie eine Formel der Prädikatenlogik erster Stufe an, die genau dann wahr ist, wenn keine direkt aufeinander liegenden Blöcke Streifen in der gleichen Farbe haben.

$\forall x \forall y \big(auf(x,y) \to (\neg rot(x) \lor \neg rot(y)) \land (\neg blau(x) \lor \neg blau(y))\big)$ (4)

6 Tableau (10 Punkte)

Es sei eine PL1-Signatur gegeben, die die einstelligen Prädikatensymbole r und s, das zweistellige Prädikatensymbol p, das einstellige Funktionssymbol f und die Konstantensymbole c und d enthält.

Vervollständigen und schließen Sie den folgenden Tableau-Beweis. Notieren Sie dabei:

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

$$1\ \forall x.\forall y.(p(x,y) \to p(f(y), f(x))) \quad (1)$$

$$1\ (p(f(f(c)), f(f(d))) \to (s(d) \wedge r(f(c)))) \quad (2)$$

$$0\ p(c,d) \to r(f(c)) \quad (3)$$

$1\ p(c,d)$ $_{4[\alpha(3)]}$

$0\ r(f(c))$ $_{5[\alpha(3)]}$

$1\ \forall y.(p(X_1,y) \to p(f(y), f(X_1)))$ $_{6[\gamma(1)]}$

$1\ (p(X_1,X_2) \to p(f(X_2), f(X_1)))$ $_{7[\gamma(6)]}$

- $0\ p(X_1,X_2)$ $_{8[\beta(7)]}$
  - $*_{4;\,8}$
- $1\ p(f(X_2), f(X_1))$ $_{9[\beta(7)]}$
  - $1\ \forall y.(p(X_3,y) \to p(f(y), f(X_3)))$ $_{10[\gamma(1)]}$
  - $1\ (p(X_3,X_4) \to p(f(X_4), f(X_3)))$ $_{11[\gamma(10)]}$
    - $0\ p(X_3,X_4)$ $_{12[\beta(11)]}$
      - $*_{9;12}$
    - $1\ p(f(X_4), f(X_3))$ $_{13[\beta(11)]}$
      - $0\ p(f(f(c)), f(f(d)))$ $_{14[\beta(2)]}$
        - $*_{13;14}$
      - $1\ (s(d) \wedge r(f(c)))$ $_{15[\beta(2)]}$
        - $1\ s(d)$ $_{16[\alpha(15)]}$
        - $1\ r(f(c))$ $_{17[\alpha(15)]}$
        - $*_{5;17}$

Die schließende Substitution ist dabei $\sigma = \{X_1/c, X_2/d, X_3/f(d), X_4/f(c)\}$.

7 Spezifikation mit der Java Modeling Language (4+3 Punkte)

a. Geben Sie für den folgenden Methodenvertrag für die Methode `void st(int d)` die Bedeutung in natürlicher Sprache an:

```
Class L {
   /*@non_null*/ int[] a;

   /*@ public normal_behaviour
     @  requires !(\exists int i; 0 <= i && i < a.length; a[i] == d);
     @  ensures  a.length == \old(a.length) + 1;
     @  ensures  a[\old(a.length)] == d;
     @  ensures  (\forall int i; 0 <= i && i < \old(a.length);
     @               a[i] == \old(a[i]));
     @  assignable a, a[*];
     @*/
   public void st(int d) {
     ...
   }
}
```

Wenn die Methode mit einem Wert `d` aufgerufen wird, der bisher nicht innerhalb des Arrays `this.a` auftritt, so gilt:

- Es wird keine Ausnahme auftreten.
- Nach Ausführung der Methode ist das array a um eine Stelle länger als vor der Ausführung.
- Nach der Ausführung steht der Wert `d` an der letzten Stelle im Array.
- Nach der Ausführung stehen in den Stellen bis auf die letzte Stelle ganau die Werte, die vor Ausführung der Methode an dieser Stelle gelesen wurden.
- Es werden nur die Speicherstelle `this.a` und alle Einträge darin verändert. (Neue Objekte können übrigens immer erstellt werden.)

(*Bemerkung:* Als implizite Vorbedingung wird zusätzlich angenommen, dass a nicht `null` ist.) Es wird ein Wert in eine Menge eingeführt. Diese Menge wird als Array repräsentiert. Ggf. kann ein neues Array erzeugt werden.

Fortsetzung Aufgabe 7

b. Die Methode `int fac(int n)` liefert für eine positive ganze Zahl n den den Wert der Fakultät $n!$ zurück.

Geben Sie eine hinreichend starke JML-Schleifeninvariante (bei `loop_invariant`) und eine JML-Variante (bei `decreases`) an.

```
/*@ public normal_behaviour
  @   requires n > 0;
  @   ensures \result == (\product int i; 1 <= i && i <= n; i);
  @   assignable \nothing;
  @*/
int fac(int n) {
  int r = 1;

  /*@ loop_invariant
    @
    @
    @
    @
    @
    @
    @
    @ decreases
    @
    @
    @
    @ assignable r, k;
    @*/
  for(int k = 1; k <= n; k++) {
    r = r * k;
  }

  return r;
}
```

Hinweis:
Bitte nehmen Sie an, dass der Datentyp `int` den ganzen Zahlen $\mathbb{Z}$ entspricht. Es findet kein Überlauf statt.

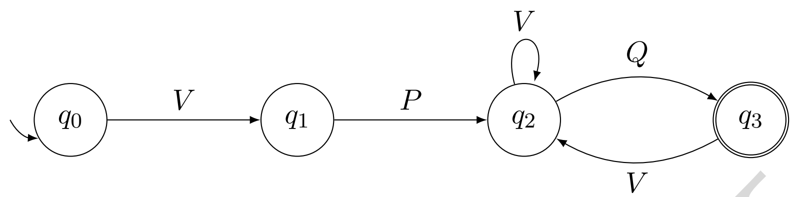
```
@ loop_invariant
@     1 <= k && k <= n+1 && r == (\product int i; 1 <= i && i < k; i);
@
@ decreases n - k + 1;
```

8 LTL und Büchi-Automaten (3+5 Punkte)

Die folgenden LTL-Formeln haben die Signatur $\Sigma = \{p, q\}$, die Büchi-Automaten das Alphabet $V = \{\emptyset, \{p\}, \{q\}, \{p, q\}\}$. Zur Hilfe definieren wir $Q = \{\{q\}, \{p, q\}\}, P = \{\{p\}, \{p, q\}\}, \overline{Q} = \{\emptyset, \{p\}\}$ und $\overline{P} = \{\emptyset, \{q\}\}$.

a. Geben Sie eine LTL-Formel A_0 an, welche genau in den *omega-Strukturen* wahr ist, die der folgende Büchi-Automat $\mathcal{A}_0$ akzeptiert (d.h., so dass $L^\omega(\mathcal{A}_0) = \{\xi \in V^\omega : \xi \models A_0\}$ gilt).

($\mathcal{A}_0$)



$A_0 = \boxed{Xp \wedge \Box\Diamond q}$

b. Geben Sie einen Büchi-Automaten $\mathcal{A}_1$ an, der genau A_1 akzeptiert (d.h., so dass $L^\omega(\mathcal{A}_1) = \{\xi \in V^\omega : \xi \models A_1\}$ gilt). (Sie dürfen die LTL-Formel vorher vereinfachen.)

$A_1 = \Diamond\Box\neg p \to \Box\Diamond q$

($\mathcal{A}_1$)