Klausur Formale Systeme

Fakultät für Informatik

WS 2012/2013

Prof. Dr. Peter H. Schmitt

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

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung (4+4 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".
- a, b und r sind Prädikatensymbole, c, f und g sind Funktionssymbole, und x und y sind Variablen.
- Es gelten die üblichen Klammereinsparungsregeln.

	keine Formel der PL1	allgemeingültig	erfüllbar, aber nicht allgemeingültig	unerfüllbar
$\big(\exists x \forall y\ r(g(x), y)\big) \to \big(\exists x\ r(x, g(x))\big)$		X		
$\forall x\ \big(f(x) \land g(x) \to (f \land g)(x)\big)$	X			
$a \to (b \lor (\neg a \to b))$		X		
$\forall x \forall y\ (r(x, y) \land \neg r(c, c))$				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
Jede unerfüllbare aussagenlogische Klauselmenge enthält wenigstens eine Klausel, die nur negative Literale enthält.	X	
Ist $\Box A$ eine modale Tautologie, dann auch $\Box\Box A$.	X	
$\Diamond A \to \Box A$ ist eine modale Tautologie.		X
Jedes lokal konfluente Termersetzungssysstem ist Nöthersch.		X

2 Äquivalenzformeln (5+2 Punkte)

Wie aus der Vorlesung bekannt bestehen (aussagenlogische) Äquivalenzformeln ausschließlich aus aussagenlogischen Atomen, den Konstanten **0** und **1** und der Äquivalenzrelation $\leftrightarrow$.

Definition. Eine Äquivalenzformel A ist in Normalform genau dann, wenn

- A die Formel **1** ist oder
- wenn **1** in A nicht vorkommt und jedes aussagenlogische Atom sowie die **0** maximal einmal in A vorkommen.

a. Zeigen Sie: Jede Äquivalenzformel lässt sich in eine aussagenlogisch äquivalente Äquivalenzformel in Normalform überführen.

Hinweis: Elementare Eigenschaften der boolschen Operatoren können ohne Beweis verwendet werden.

Sei A eine Äquivalenzformel, die noch nicht in Normalform ist. Durch Verwendung der Assoziativität und Kommutativität von $\leftrightarrow$ kann man aus A eine aussagenlogisch äquivalente Formel konstruieren, die mindestens eine der Formeln $x \leftrightarrow x$, $\mathbf{0} \leftrightarrow \mathbf{0}$ $\mathbf{1} \leftrightarrow \mathbf{1}$ als Unterformel enthält. Aufgrund der Äquivalenzen $x \leftrightarrow x \equiv \mathbf{1}$, $\mathbf{0} \leftrightarrow \mathbf{0} \equiv \mathbf{1}$ kann man durch Ersetzung der linken Seiten durch die rechte zu einer äquivalente Formel A' kommen, die zwei Vorkommen von x bzw. **0** weniger enthält. Diese Operation wiederholt man bis jede Variable und die Konstante **0** höchstens einmal in A' vorkommen.
Durch äquivalente Ersetzung von $\mathbf{1} \leftrightarrow \mathbf{1}$ durch **1** kann man die Anzahl der Vorkommen von **1** reduzieren bis entweder **1** überhaupt nicht mehr vorkommt oder die ganze Formel nur noch aus **1** besteht.

b. Zeigen oder widerlegen Sie: Jede aussagenlogische Formel lässt sich als Äquivalenzformel darstellen.

Betrachten wir aussagenlogische Formeln über der Signatur $\Sigma = \{a_1, a_2\}$. Nach Teilaufgabe **a.** sind alle Äquivalenzformel über Σ äquivalent zu einer der folgenden 8 Formeln: **1**, **0**, a_1, a_2, $\mathbf{0} \leftrightarrow a_1$, $\mathbf{0} \leftrightarrow a_2$, $a_1 \leftrightarrow a_2$ oder $\mathbf{0} \leftrightarrow a_1 \leftrightarrow a_2$. Insgesamt gibt es aber $2^4 = 16$ verschiedene binäre Operatoren. Also lässt sich nicht jede aussagenlogische Formel als Äquivalenzformel darstellen. (Wie sich leicht überprüfen lässt ist z. B. $a_1 \wedge a_2$ zu keiner der 8 Formeln äquivalent.)

3 Unifikation (1+1+3 Punkte)

Seien

- f ein zweistelliges Funktionssymbol,
- g ein einstelliges Funktionssymbol,
- c, d nullstellige Funktionssymbole,
- x, y, z Variablen.

Bestimmen Sie für die folgenden Mengen von Termen einen allgemeinsten Unifikator μ (als *eine* Substitution und nicht als Verkettung von Substitutionen). Falls es keinen allgemeinsten Unifikator gibt, begründen Sie, warum es keinen gibt!

a. $\{\ f(c, x),$
$f(d, c)\ \}$

Nicht unifizierbar, die Menge enthält mehr als einen Term und die Differenz der Menge enthält keine Variable.

b. $\{\ x,$
$y,$
$z\ \}$

1. $\{y, z\}$ $\qquad \mu = \{x/y\}$
2. $\{z\}$ $\qquad \mu = \{y/z\} \circ \{x/y\} = \{x/z, y/z\}$

$\Rightarrow \mu = \{y/z, x/z\}$ ist ein allgemeinster Unifikator

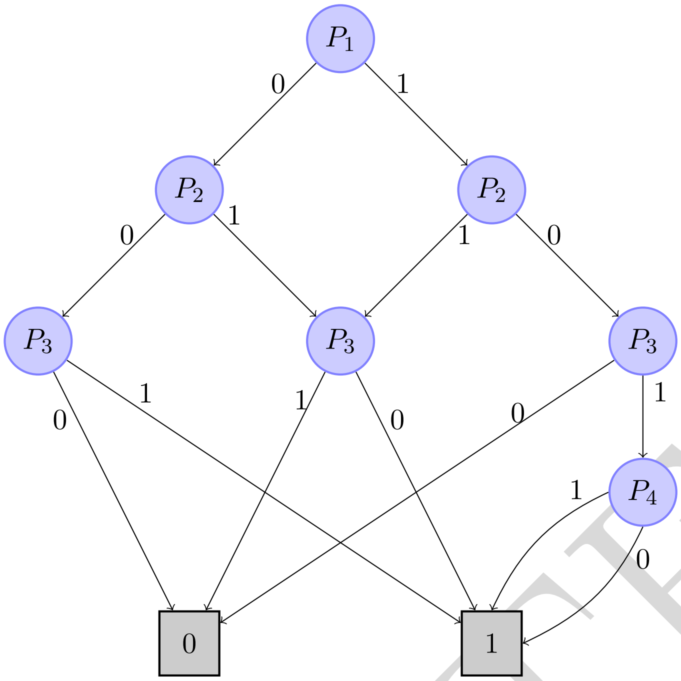
c. $\{\ f(g(z), g(f(g(c), f(z, c)))),$
$f(x, g(f(z, y)))\ \}$

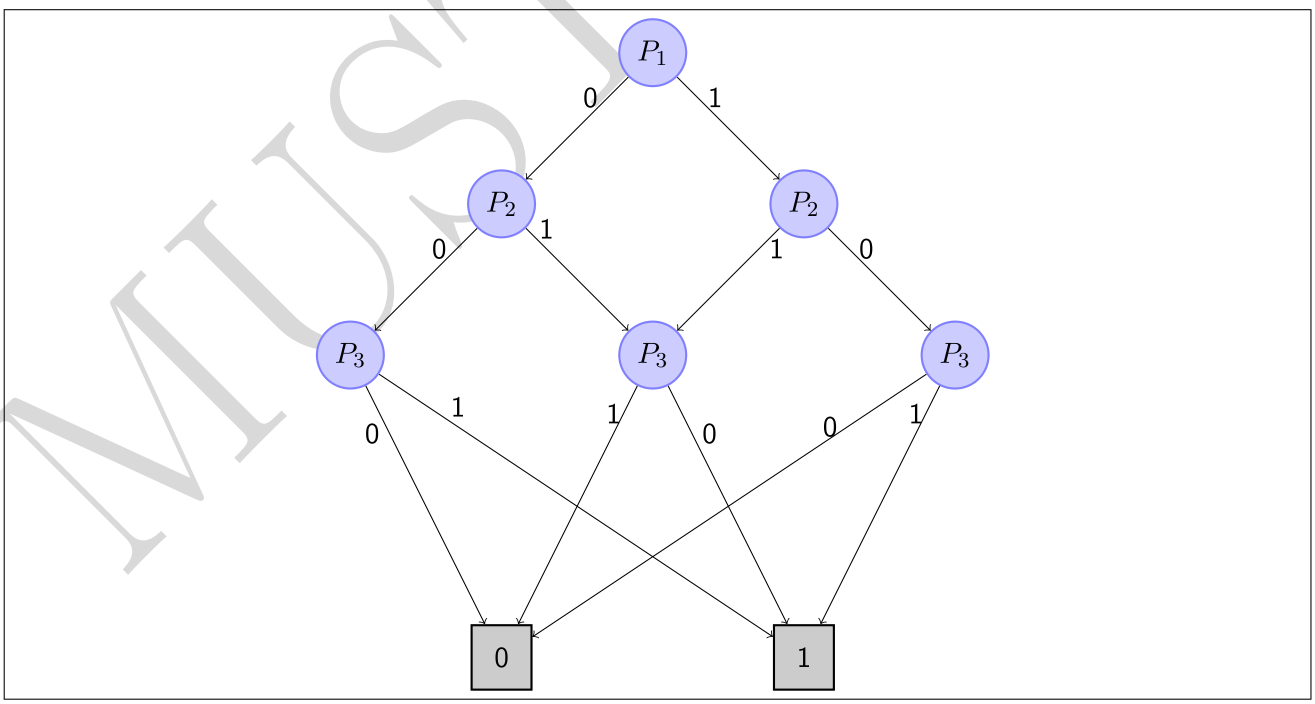
1. $f(g(z), g(f(g(c), f(z, c)))),$
   $f(g(z), g(f(z, y)))$ $\qquad \mu = \{x/g(z)\}$
2. $f(g(g(c)), g(f(g(c), f(g(c), c)))),$
   $f(g(g(c)), g(f(g(c), y)))$ $\qquad \mu = \{z/g(c)\} \circ \{x/g(z)\} = \{x/g(g(c)), z/g(c)\}$
3. $f(g(g(c)), g(f(g(c), f(g(c), c)))),$
   $f(g(g(c)), g(f(g(c), f(g(c), c))))$ $\qquad \mu = \{y/f(g(c), c)\} \circ \{x/g(g(c)), z/g(c)\}$
   $= \{x/g(g(c)), z/g(c), y/f(g(c), c)\}$

$\Rightarrow \mu = \{x/g(g(c)), z/g(c), y/f(g(c), c)\}$ ist ein allgemeinster Unifikator.

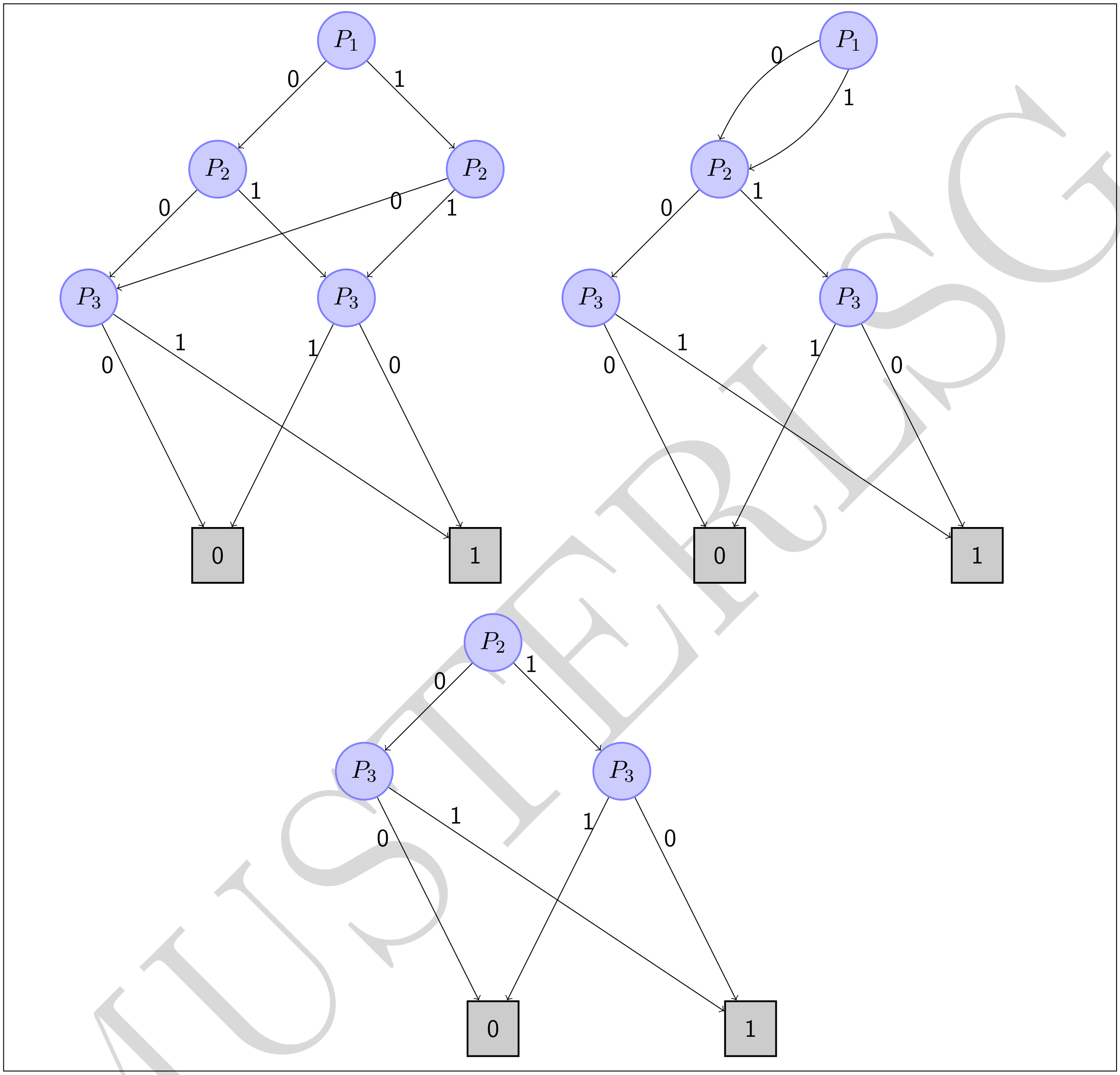
4 Shannongraphen (6 Punkte)

Konstruieren Sie zu dem folgenden Shannnongraphen den reduzierten Shannongraphen (mit der gleichen Variablen-Ordnung $P_1 < P_2 < P_3 < P_4$). Verwenden Sie das Verfahren aus der Vorlesung. Geben Sie alle Zwischenschritte an, d. h. geben Sie nach jedem Reduktionsschritt den daraus resultierenden Graphen an. Es gibt für jeden korrekten Zwischenschritt Punkte, d. h. insbesondere, die volle Punktzahl kann nur erreicht werden, wenn alle Zwischenschritte korrekt angegeben sind.





Zusatzblatt für Shannongraphen

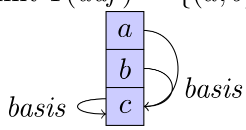


5 Formalisieren in Prädikatenlogik (1 + 1,5 + 3 + 2,5 + 2 Punkte)

Gegeben sei die prädikatenlogische Signatur, welche die einstelligen Funktionssymbole $basis(\cdot)$ und $top(\cdot)$ und das zweistellige Prädikatensymbol $auf(\cdot,\cdot)$ enthält.

Wir betrachten Blockwelten. Ein Universum D besteht also 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.

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



So wie *basis* auf den untersten Block eines Turmes verweist, soll *top* auf den obersten Block verweisen.

a. Geben Sie eine Formel der Prädikatenlogik erster Stufe an, die genau dann wahr ist, wenn jeder Block und der oberste Block seines Turmes die gleiche Basis haben.

$\forall x\big(basis(x) \doteq basis(top(x))\big)$ (1)

b. Geben Sie eine Formel der Prädikatenlogik erster Stufe an, die genau dann wahr ist, wenn keine Blöcke auf obersten Blöcken liegen.

$\forall x \forall y \neg auf(x, top(y))$ (2)

c. Geben Sie eine Formel der Prädikatenlogik erster Stufe an, die genau dann wahr ist, wenn jeder Turm mindestens drei Blöcke enthält.

$\forall x \exists y\big(auf(y, basis(x)) \land \neg top(x) \doteq y\big)$
Alternativ: $\forall x \exists y\big(\neg y \doteq basis(y) \land \neg y \doteq top(y) \land basis(x) \doteq basis(y)\big)$ (3)

d. Wie sehen die Blockwelten aus, wenn folgende Formel wahr ist?

$$\forall x\big(top(basis(x)) \doteq basis(top(x))\big) \quad (4)$$

Alle Türme enthalten genau einen Block.

e. Wie sehen die Blockwelten aus, wenn folgende Formel wahr ist?

$$\forall x\big(auf(top(x), basis(x))\big) \quad (5)$$

Alle Türme enthalten genau zwei Blöcke.

6 Resolution (10 Punkte)

Beweisen Sie mit Hilfe des Resolutionskalküls für die PL1, dass die unten stehende Menge prädikatenlogischer Klauseln unerfüllbar ist. Darin sind q, r einstellige und p ein zweistelliges Prädikatensymbol. f ist ein einstelliges Funktionssymbol und c, d sind Konstantensymbole.

Machen Sie bei jedem Resolutionsschritt erkenntlich, aus **welchen Ausgangsklauseln** die Resolvente entsteht und **welche Substitution** Sie dazu verwendet haben.

$$\Big\{\underbrace{\{\neg p(y_1, x_1), q(f(y_1))\}}_{(1)}, \underbrace{\{p(x_2, f(y_2)), r(f(d))\}}_{(2)}, \underbrace{\{q(d), \neg r(x_3)\}}_{(3)}, \underbrace{\{\neg q(x_4), \neg r(f(x_4))\}}_{(4)}, \underbrace{\{r(f(f(c))), \neg q(f(c))\}}_{(5)}\Big\}$$

(6)	$\{\neg q(f(c))\}$	$(4), (5)$	$\sigma = \{x_4/c\}$
(7)	$\{\neg p(c, x_1)\}$	$(6), (1)$	$\sigma = \{x_1/c\}$
(8)	$\{r(f(d))\}$	$(7), (2)$	$\sigma = \{x_2/c, x_1/f(y_2)\}$
(9)	$\{q(d)\}$	$(8), (3)$	$\sigma = \{x_3/f(d)\}$
(10)	$\{\neg q(d)\}$	$(8), (4)$	$\sigma = \{x_4/d\}$
(11)	$\square$	$(9), (10)$	$\sigma = \{\}$

7 Spezifikation mit der Java Modeling Language (5 Punkte)

a. Gegeben sei eine Methode `m` mit folgender Signatur:

```
public int m(int[] a, int[] b);
```

Geben Sie für den folgenden Methodenvertrag in natürlicher Sprache eine äquivalente JML-Spezifikation für `m` an:

Wenn die Methode `m` mit Arrays gleicher Länge aufgerufen wird, so gilt:

- Es wird keine Ausnahme auftreten.
- Ist das Ergebnis der Methode ungleich 0, so gibt es einen Index `i` innerhalb der Array-Grenzen so, dass die Differenz `a[i]-b[i]` ungleich 0 ist. Das Ergebnis der Methode entspricht in diesem Fall der Differenz `a[i]-b[i]` für den *kleinsten* Index `i` mit dieser Eigenschaft.
- Ist das Ergebnis der Methode 0, so stimmen die Einträge der Arrays an allen Stellen überein.
- Es werden keine vom Programm bereits verwendeten Speicherstellen verändert. (Neue Objekte können allerdings erstellt werden.)

```
/*@
  @ public normal_behaviour
  @   requires a.length == b.length;
  @   ensures  \result != 0 ==>
  @     (\exists int i; 0<=i && i < a.length;
  @          a[i] != b[i]
  @      && (\forall int j; 0<=j && j < i; a[j] == b[j])
  @      && \result == a[i] - b[i]);
  @   ensures  \result == 0 ==>
  @     (\forall int i; 0<=i && i < a.length; a[i] == b[i]);
  @   assignable \nothing;

  @*/
public int m(int[] a, int[] b);
```

8 LTL und Büchi-Automaten (1+5+3 Punkte)

Seien p und q aussagenlogische Variablen. Dann ist die Semantik der LTL-Formel $p\,\mathbf{f}\,q$ ("p forbids q") folgendermaßen definiert:

$$\xi \models p\,\mathbf{f}\,q :\iff \text{Für jedes } n \in \mathbb{N}, \text{ für das } \xi_n \models p \text{ gilt, gilt } \xi_{n+1} \models \neg q.$$

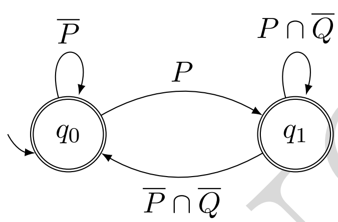
a. Geben Sie eine zu $p\,\mathbf{f}\,q$ äquivalente LTL-Formel an, die $\mathbf{f}$ nicht verwendet.

$\Box(p \to X\neg q)$

b. Geben Sie einen Büchi-Automaten $\mathcal{A}_\mathbf{f}$ über dem Alphabet $V = \{\emptyset, \{p\}, \{q\}, \{p,q\}\}$ an, der genau $p\,\mathbf{f}\,q$ akzeptiert, d.h. so dass $L^\omega(\mathcal{A}_\mathbf{f}) = \{\xi \in V^\omega : \xi \models p\,\mathbf{f}\,q\}$ gilt.

Sie können folgende vordefinierte Mengen mitbenutzen:

$V = \{\emptyset, \{p\}, \{q\}, \{p,q\}\}$ $\quad Q = \{\{q\}, \{p,q\}\}$
$P = \{\{p\}, \{p,q\}\}$ $\quad \overline{Q} = \{\emptyset, \{p\}\}$
$\overline{P} = \{\emptyset, \{q\}\}$



c. Geben Sie eine Formel mit ausschließlich aussagenlogischen Operatoren und dem Operator $\mathbf{f}$ an, die ausdrückt, dass genau in den geraden Zeitpunkten p gilt.

$p \wedge (p\,\mathbf{f}\,p) \wedge (\neg p\,\mathbf{f}\,\neg p)$