Klausur Formale Systeme

Fakultät für Informatik

WS 2018/2019

Prof. Dr. Bernhard Beckert

01. März 2019

Name: _____

Vorname: _____

Matrikel-Nr.: _____

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (18)	A2 (9)	A3 (6)	A4 (8)	A5 (8)	A6 (7)	A7 (9)	Σ (65)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung (5+10+3 = 18 Punkte)

a. Kreuzen Sie in der folgenden Tabelle alles Zutreffende an.

Für jede korrekte Antwort gibt es einen Punkt, für falsche Antworten gibt es keinen Punkt (für falsche Antworten wird nichts abgezogen).

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, q sind Prädikatensymbole, f ist ein Funktionssymbol und x, y sind Variablen.
- Es gelten die üblichen Klammereinsparungsregeln.

	keine Formel der PL1	allgemeingültig	erfüllbar, aber nicht allgemeingültig	unerfüllbar
$\forall x\, p(f(x)) \to \forall y\, p(y)$			X	
$\exists x \exists y\, (x \doteq y \wedge p(x) \wedge \neg p(y))$				X
$\exists x\, f(p(x)) \doteq p(f(x))$	X			
$(\exists x \forall y\, (p(x) \to q(y))) \;\leftrightarrow\; (\forall y \exists x\, (p(x) \to q(y)))$		X		
$\forall x\, \big((p(x) \to q(x)) \vee (q(x) \to p(x))\big)$		X		

b. Kreuzen Sie in der folgenden Tabelle das Zutreffende an.

Für jede korrekte Antwort gibt es zwei Punkte, für falsche Antworten gibt es keinen Punkt (für falsche Antworten wird nichts abgezogen).

	Richtig	Falsch
Die Formel $\Box P \to P$ charakterisiert die Klasse der transitiven Kripke-Rahmen.		X
In der Aussagenlogik gilt: $p \wedge q$ ist genau dann unerfüllbar, wenn $p \to \neg q$ allgemeingültig ist.	X	
Die Erfüllbarkeit aussagelogischer Horn-Formeln ist in Polynomialzeit entscheidbar.	X	
Seien x eine Variable, c eine Konstante und f, g Funktionssymbole. Dann sind die Terme x und $f(c, g(x))$ unifizierbar.		X
Eine *unendliche* Menge M prädikatenlogischer Formeln (PL1) ist genau dann unerfüllbar, wenn es eine *endliche* unerfüllbare Teilmenge $E \subset M$ gibt.	X	

Fortsetzung 1 Zur Einstimmung

c. Überprüfen Sie folgende Horn-Formel auf Erfüllbarkeit. Benutzen Sie dafür den in der Vorlesung vorgestellten Markierungsalgorithmus.

Geben Sie dabei für jede Iteration des Algorithmus an, **welche(s) Atom(e) in der jeweiligen Iteration markiert** wurde(n).

Wenn die Formel erfüllbar ist, geben Sie **zudem** ein **Modell** an!

$$P_1 \wedge P_3 \wedge (P_3 \to P_2) \wedge (P_1 \wedge P_2 \to P_4) \wedge (P_4 \wedge P_5 \to \mathbf{0})$$

Iteration 1 — Markiere alle Fakten (P_1 und P_3):

$$\boxed{P_1} \wedge \boxed{P_3} \wedge (\boxed{P_3} \to P_2) \wedge (\boxed{P_1} \wedge P_2 \to P_4) \wedge (P_4 \wedge P_5 \to \mathbf{0})$$

Iteration 2 — Markiere P_2:

$$\boxed{P_1} \wedge \boxed{P_3} \wedge (\boxed{P_3} \to \boxed{P_2}) \wedge (\boxed{P_1} \wedge \boxed{P_2} \to P_4) \wedge (P_4 \wedge P_5 \to \mathbf{0})$$

Iteration 3 — Markiere P_4:

$$\boxed{P_1} \wedge \boxed{P_3} \wedge (\boxed{P_3} \to \boxed{P_2}) \wedge (\boxed{P_1} \wedge \boxed{P_2} \to \boxed{P_4}) \wedge (\boxed{P_4} \wedge P_5 \to \mathbf{0})$$

Iteration 4 — Es kann nichts mehr markiert werden, die Formel is also **erfüllbar**.

Modell:
$I(P_1) = W$,
$I(P_2) = W$,
$I(P_3) = W$,
$I(P_4) = W$,
$I(P_5) = F$.

2 Modallogik (2+3+4 = 9 Punkte)

Für diese Aufgabe betrachten wir die „Hoffen-Modallogik“, in der der Box-Operator $\Box$ als “hoffen” (*engl.: to hope*) interpretiert wird. Das heißt, die Formel

$$\Box_a F$$

bedeutet

Agent a hofft, dass F wahr ist

Im folgenden sei P eine aussagenlogische Variable.

a. Beschreiben sie in natürlicher Sprache die Bedeutung folgender Formel in der Hoffen-Modallogik:

$$\Diamond_a P$$

Die Aussage P ist mit den Hoffnungen des Agenten a konsistent (steht zu diesen nicht in Widerspruch). Alternativ: es ist nicht der Fall, dass Agent a hofft, dass P falsch ist.

b. Beschreiben sie in natürlicher Sprache die Bedeutung folgender Formel in der Hoffen-Modallogik:

$$\Box_a P \rightarrow P$$

Wenn Agent a hofft, dass P wahr ist, dann ist P auch wahr. Bewertung: 1P

Sollte diese Formel ein Axiom der Hoffen-Modallogik sein? Begründen Sie!

Nein. Nur weil, jemand hofft, dass etwas wahr sei, ist es noch lange nicht wahr. Bewertung: 2P (1/2P für “nein” ohne Begründung).

c. Beschreiben sie in natürlicher Sprache die Bedeutung folgender Formel in der Hoffen-Modallogik:

$$(\Box_a P) \rightarrow \Box_a (P \rightarrow \Box_a P)$$

Wenn Agent a hofft, dass P wahr ist, dann hofft Agent a auch, dass er, wenn er sich in einer Welt/Situation befindet, in der P wahr ist, (immer noch) hofft, dass P wahr sei. Bewertung: 2P

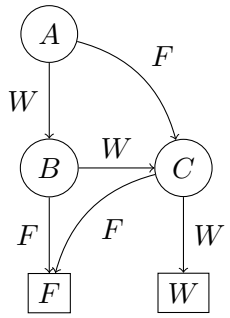
Sollte diese Formel ein Axiom der Hoffen-Modallogik sein? Begründen Sie!

Ja. Das drückt aus, dass man hofft, dass die eigenen Hoffnungen erhalten bleiben, wenn der Wunsch in Erfüllung geht, dass die Hoffnungen also mit der Wirklichkeit kompatibel sind. Bewertung: 2P; 1/2P für “ja” ohne Begründung

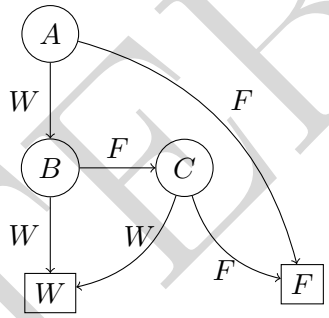
3 Shannongraphen (3+3 = 6 Punkte)

Zeichnen Sie zu jeder der folgenden aussagenlogischen Formeln den reduzierten Shannongraphen. Die Variablenordnung ist dabei $A < B < C$.

a. $(A \to B) \wedge C$



b. $(A \wedge B) \vee (A \wedge C)$



4 Formalisieren in PL1 (2+2+2+2 = 8 Punkte)

Gegeben sei die prädikatenlogische Signatur $\Sigma = (\{\}, \{put, get, empty, null\}, \alpha)$. Sie enthält die Konstanten *empty* und *null*, sowie die mehrstelligen Funktionssymbole $put(\cdot,\cdot,\cdot)$ und $get(\cdot,\cdot)$.

Zur Auswertung der Formeln werden nur solche Interpretationen (D, I) über Σ verwendet, in denen

- das Universum D eine Menge von Maps (assoziative Datenfelder) ist,
- die Funktion $put(m, k, v)$ eine Map zurückliefert, die dadurch entsteht, dass in der Map m der Wert v unter dem Schlüssel k eingetragen wird,
- die Funktion $get(m, k)$ den Wert der Map m unter dem Schlüssel k zurückliefert,
- die Konstante *empty* die leere Map bezeichnet,
- die Konstante *null* bezeichnet, dass unter einem Schlüssel kein Wert eingetragen ist.

Geben Sie jeweils eine Formel der Prädikatenlogik mit Gleichheit über Σ an, die folgende Sachverhalte darstellt:

a. Die leere Map *empty* hat unter keinem Schlüssel einen eingetragenen Wert.

$\boxed{\forall k\ get(empty, k) \doteq null}$

---

b. Entsteht eine Map dadurch, dass ein Wert v unter einem Schlüssel k in eine bereits existierende Map eingetragen wird, so befindet sich in der neu entstandenen Map der Wert v unter dem Schlüssel k.

$\boxed{\forall m \forall k \forall v\ get(put(m, k, v), k) \doteq v}$

---

c. Außer *empty* gibt es keine andere Map, in der unter keinem Schlüssel ein Wert eingetragen ist.

$\boxed{\forall m\ \big((\forall k\ get(m, k) \doteq null) \rightarrow (m \doteq empty)\big)}$

---

d. Zwei Maps sind genau dann gleich, wenn in beiden Maps unter jedem Schlüssel derselbe Wert eingetragen ist.

$\boxed{\forall m_1 \forall m_2 ((m_1 \doteq m_2) \leftrightarrow \forall k\ (get(m_1, k) \doteq get(m_2, k)))}$

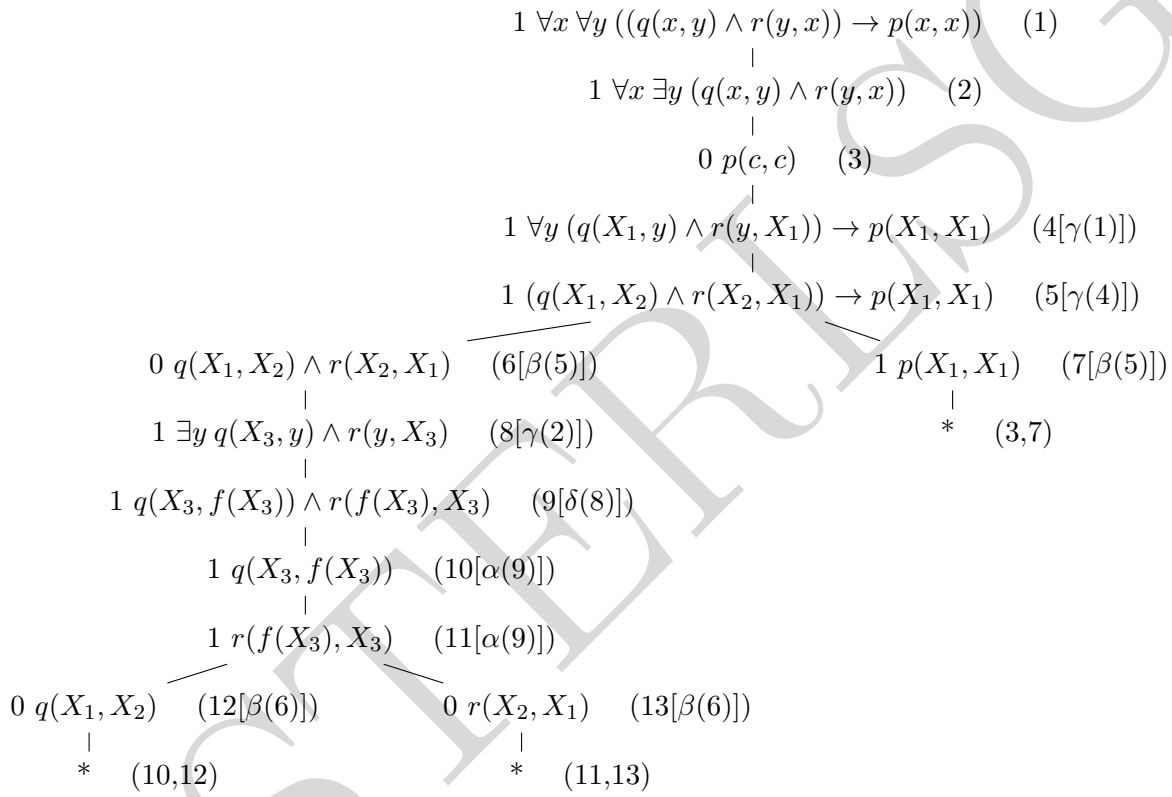
---

5 Tableaukalkül (8 Punkte)

Vervollständigen und schließen Sie den folgenden Tableau-Beweis.

Notieren Sie dabei:

- den Regeltyp $(\alpha, \beta, \gamma, \delta)$ und die Formel, auf die eine Regel angewendet wird,
- bei Abschlüssen die beiden Partner,
- sowie die schließende Substitution.



Schließende Substitution: $\sigma = \{X_1 \setminus c, X_2 \setminus f(c), X_3 \setminus c\}$

6 Spezifikation mit der Java Modeling Language

(3+4 = 7 Punkte)

a. Geben Sie die Bedeutung der folgenden JML-Klasseninvariante in natürlicher Sprache wieder.

```
public class A {
    public int[] p;
    /*@ public invariant p != null
      @       && (\forall int i; 0 <= i && i < p.length;
      @              (\forall int j; 0 <= j && j < p.length; i != j ==> p[i] != p[j]))
      @       && (\forall int i; 0 <= i && i < p.length; 0 <= p[i] && p[i] < p.length)
      @       && (\forall int i; 0 <= i && i < p.length;
      @              (\exists int j; 0 <= j && j < p.length; p[j] == i);
      @*/
}
```

Das Array `p` ist von `null` verschieden, alle Array-Elemente sind voneinander verschieden und alle Array-Elemente sind mindestens vom Wert `0` und höchstens vom Wert `p.length - 1`. Außerdem ist jede Zahl von `0` bis `p.length - 1` im Array `p` enthalten.

Fortsetzung 6 Spezifikation mit der Java Modeling Language

b. Sei weiterhin eine Methode `m` in der obigen Klasse `A`. Vervollständigen Sie den nachstehenden JML-Methodenvertrag für `m`, so dass er Folgendes besagt:

Wenn `a` von `null` und von `p` verschieden ist, sowie mindestens so viele Elemente wie das Array `p` enthält, terminiert die Methode und wirft keine Exception, ändert keine Speicherstellen, und erfüllt folgende Nachbedingungen:

- Die Methode gibt ein von `null` verschiedenes `int`-Array zurück, das genau so viele Elemente wie das Array `a` enthält.
- An jeder Position `i` von `0` bis zur Länge des Arrays `p` (exklusive) enthält das zurückgegebene Array den Wert, den das Array `a` an der Position enthält, die dem Wert gleich ist, der im Array `p` an Position `i` steht.
- An allen weiteren Positionen (Länge von `p` bis zum Ende von `a`) sind die Werte des zurückgegebenen Arrays identisch zu den Werten des Arrays `a`.

```
public class A {
    public int[] p;
    /*@
      @
      @
      @
      @
      @
      @
      @
      @
      @
      @
      @
      @
      @
      @*/
    public int[] m(int[] a) { ... }
}
```

```
/*@ public normal_behaviour
  @  requires a != null && a != p && p.length <= a.length;
  @  assignable \nothing;
  @  ensures \result != null && \result.length == a.length;
  @  ensures (\forall int i; 0 <= i && i < p.length;
  @                          \result[i] == a[p[i]]);
  @  ensures (\forall int i; p.length <= i && i < a.length;
  @                          \result[i] == a[i]);
  @*/
public int[] m(int[] a) { ... }
```

7 Lineare Temporale Logik (LTL) und Büchiautomaten (2+3+4 = 9 Punkte)

Formalisieren Sie folgende Aussagen in LTL über der Signatur $\Sigma = \{a, e\}$. Dabei verwenden wir folgende Definition des Begriffs e-Intervall:

> **Definition.** Sei eine ω-Struktur ξ gegeben. Ein e-Intervall in ξ ist ein zusammenhängendes Teilstück von Zeitpunkten in ξ, so dass:
>
> - e ist wahr in jedem der Zeitpunkte des Intervalls,
> - e ist falsch in dem Zeitpunkt direkt vor und in dem Zeitpunkt direkt nach dem Intervall.

Hinweis 1 Zum besseren Verständnis: Sind $n, m \in \mathbb{N}$ der erste und letzte Zeitpunkt eines e-Intervalls, dann gilt

1. $\xi_i \models e$ für alle $n \leq i \leq m$,
2. $\xi_{n-1} \not\models e$ und $\xi_{m+1} \not\models e$

Hinweis 2 Zur Vereinfachung gehen wir davon aus, dass ein e-Intervall entsprechend der obigen Definition nicht im Anfangszustand einer ω-Struktur beginnen kann, so dass der in der Definition erwähnte „Zustand direkt vor dem Intervall“ immer existiert.

Hinweis 3 Bedenken Sie, dass es nützlich sein kann, den Next-Operator $\mathbf{X}$ zu verwenden.

a. a kann nur innerhalb eines e-Intervalls wahr sein.

$$\square\big(a \rightarrow e\big)$$ (2 Punkte)

b. a ist in jedem e-Intervall in höchstens einem Zeitpunkt wahr.

$$\square\Big(\big(a \wedge e\big) \rightarrow \mathbf{X}\big(\neg a\ \mathbf{U}\ \neg e\big)\Big)$$ (3 Punkte)

c. a ist in jedem e-Intervall in mindestens einem Zeitpunkt wahr.

$$\square\Big(\big(\neg e \wedge \mathbf{X}e\big) \rightarrow \mathbf{X}\big(e\ \mathbf{U}\ a\big)\Big)$$ (4 Punkte)