Karlsruher Institut für Technologie

Name:	
Vorname:	
Matrikel-Nr.:	

Klausur Formale Systeme

Fakultät für Informatik

WS 2023/24

Prof. Dr. Bernhard Beckert

28. Februar 2024

Die Bearbeitungszeit beträgt 60 Minuten.

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

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung ((2 + 1 + 1 + 2) + 4 = 10 Punkte)

a. Geben Sie kurze Antworten zu folgenden Fragen bzw. Aufgaben:

i. Gegeben sei die Signatur $\Sigma = (\{p, q\}, \{c, d\}, \alpha)$ mit Stelligkeiten $\alpha(p) = \alpha(q) = 1$ und $\alpha(c) = \alpha(d) = 0$.
Geben Sie ein Herbrand-Modell (D, I) über Σ an für die Formel $(\forall x\, p(x)) \wedge \neg q(c)$.

$D =$ _____

$I(c) =$ _____

$I(d) =$ _____

$I(p)$ (_____) = _____

$I(p)$ (_____) = _____

$I(q)$ (_____) = _____

$I(q)$ (_____) = _____

ii. Nennen Sie jeweils eine Theorie in Prädikatenlogik erster Stufe, deren Erfüllbarkeitsproblem entscheidbar bzw. nicht entscheidbar ist:

Entscheidbar: _____

Nicht entscheidbar: _____

iii. Sei F eine aussagenlogische Tautologie über den Variablen A, B, C. Wie viele unterschiedliche reduzierte Shannongraphen, die F darstellen, gibt es?

_____

iv. Gegeben sei eine aussagenlogische Klauselmenge mit m Klauseln über n Variablen, von denen k Klauseln Einerklauseln sind. Wie viele Fallunterscheidungen benötigt DPLL maximal, um die Unerfüllbarkeit zu beweisen? Geben Sie die genaue Schranke an (keine O-Notation) und begründen Sie diese.

_____

_____

_____

_____

Fortsetzung 1 Zur Einstimmung

b. Geben Sie korrekte und vollständige Tableaukalkülregeln für den Shannon-Operator sh an. Denken Sie daran, die Regeln für beide Vorzeichen anzugeben.

Hinweis: Per Definition von sh gilt $\mathrm{sh}(\mathbf{W}, P, Q) \equiv Q$ und $\mathrm{sh}(\mathbf{F}, P, Q) \equiv P$.

2 Unsat Core (2 + 2 + 3 + 2 = 9 Punkte)

Das Konzept des *Unsat Core* ist wie folgt definiert:

> Sei U eine unerfüllbare Formelmenge. Eine Teilmenge $C \subseteq U$ ist ein *Unsat Core* (von U) genau dann, wenn:
>
> 1. C ist unerfüllbar,
> 2. jede echte Teilmenge $C' \subsetneq C$ ist erfüllbar.

Hinweis: Diese Definition ist unabhängig davon, in welcher Logik man sich befindet, so lange das Konzept der Erfüllbarkeit definiert ist.

a. Geben Sie einen Unsat Core der folgenden Menge prädikatenlogischer Formeln an und begründen Sie ihre Antwort.

$$\{ \quad \forall x\,(p(x) \wedge q(x)), \quad p(a) \vee q(a), \quad \neg p(b) \quad \}$$

(p ist ein einstelliges Prädikatensymbol und a, b sind Konstanten.)

b. Geben Sie ein aussagenlogisches Beispiel an, das zeigt, dass eine unerfüllbare Menge U zwei verschiedene Unsat Cores $C_1 \neq C_2$ haben kann mit $C_1 \cap C_2 \neq \emptyset$.

c. Ist entscheidbar, ob eine gegebene endliche Formelmenge C ein Unsat Core ist? Wie hängt das damit zusammen, ob das Erfüllbarkeitsproblem in der jeweiligen Logik entscheidbar ist?

d. Im Zusammenhang mit automatischem Beweisen ist Erklärbarkeit des Ergebnisses ein wichtiges Konzept. Begründen Sie kurz, warum das Konzept des Unsat Core für die Erklärbarkeit der Ergebnisse, die ein Beweissystem liefert, wichtig ist.

3 Entscheidungsverfahren für uninterpretierte Funktionssymbole (5 + 2 = 7 Punkte)

a. Gegeben ist die folgende prädikatenlogische Formel:

$$\underset{(A)}{a \doteq f(b)} \wedge \underset{(B)}{b \doteq f(a)} \wedge \underset{(C)}{a \doteq g(a, f(a))} \wedge \underset{(D)}{\neg a \doteq g(a, b)} \wedge \underset{(E)}{\neg a \doteq g(f(b), b)} \ . \tag{1}$$

Darin sind a, b Konstantensymbole und $f(\cdot), g(\cdot, \cdot)$ sind Funktionssymbole.

Geben Sie den vollständigen Kongruenzgraphen durch die Ausführung des *Shostak*-Algorithmus auf die Formel (1) an.

Durchgezogene Kanten stehen dabei für Gleichheiten und gestrichelte Kanten für Ungleichheiten. Geben Sie an jeder Kante die Begründung an (Name der Gleichheiten).

b. Lesen Sie aus dem fertigen Graphen ab, ob (1) erfüllbar ist oder nicht. Begründen Sie und geben Sie wenn möglich ein erfüllendes Modell an.

4 Formalisieren in Prädikatenlogik (PL1)

(1 + 2 + 2 + 2 = 7 Punkte)

Gegeben sei die prädikatenlogische Signatur $\Sigma = (\{implies\}, \{statement, person, logician, believes\}, \alpha)$. Sie enthält die einstelligen Prädikatensymbole $statement(\cdot)$, $person(\cdot)$ und $logician(\cdot)$, das zweistellige Prädikatensymbol $believes(\cdot, \cdot)$ sowie das zweistellige Funktionssymbol $implies(\cdot, \cdot)$.

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

- das Universum D eine Menge von Aussagen, Personen und Logikerinnen ist,
- das Prädikat $statement(x)$ genau dann wahr ist, wenn x eine Aussage ist,
- das Prädikat $person(x)$ genau dann wahr ist, wenn x eine Person ist,
- das Prädikat $logician(x)$ genau dann wahr ist, wenn x eine Logikerin ist,
- das Prädikat $believes(x, y)$ genau dann wahr ist, wenn Logikerin x glaubt, dass Aussage y wahr ist, und
- die Funktion $implies(x, y)$ die Aussagen x und y auf die Aussage "$x \longrightarrow y$" (y folgt aus x, bzw. unter Voraussetzung x gilt die Folgerung y) abbildet.

Hinweis: Die Aussage "$x \longrightarrow y$" bezeichnen wir forthin als *Implikation*.

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

a. Jede Logikerin ist eine Person.

b. Es gibt (mindestens) eine Aussage,

- die alle Logikerinnen glauben und
- die als Folgerung mit beliebiger Voraussetzung dazu führt, dass die Implikation von allen Logikerinnen geglaubt wird.

c. Wenn eine Logikerin glaubt, dass eine Aussage wahr ist, und sie außerdem glaubt, dass diese erste Aussage eine zweite Aussage impliziert, dann glaubt sie auch diese zweite Aussage.

d. Es gibt (mindestens) eine Folgerung, deren Implikation keine Logikerin glaubt, wenn sie die Voraussetzung glaubt.

5 Tableaukalkül (10 Punkte)

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

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

Notieren Sie dabei:

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

6 Spezifikation mit der Java Modeling Language

(4 + (1,5 + 3,5) = 9 Punkte)

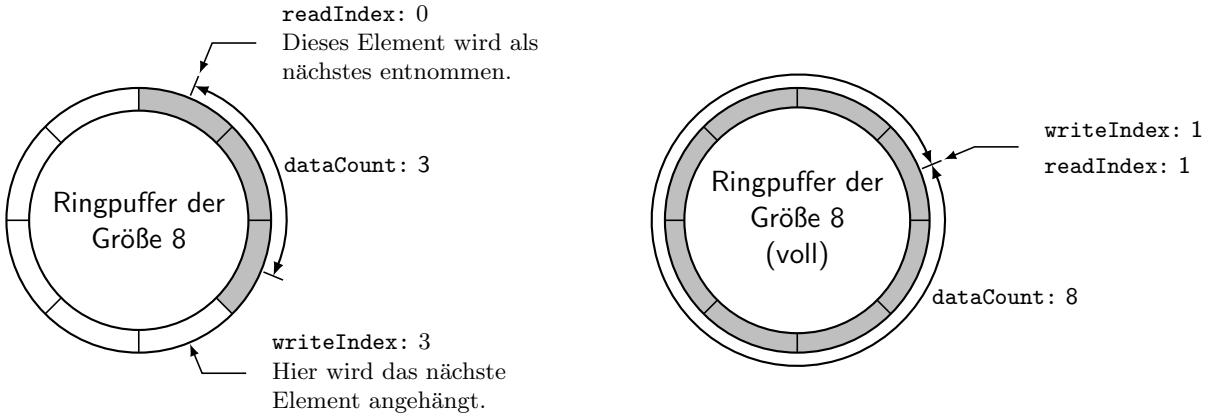
a. Geben Sie in natürlicher Sprache wieder, was der folgende JML-Methodenvertrag für die Methode `m` aussagt.

Hinweis: Die Möglichkeit von Integer Overflows können Sie für diese Aufgabe ignorieren.

```
public final class A {
    /*@ normal_behavior
      @  requires a.length > 0;
      @  requires (\forall int i; 0 <= i && i < a.length; a[i] > 0);
      @  ensures (\forall int i; 0 <= i && i < a.length;
      @               (\exists int k; 0 < k; \result * k == a[i]));
      @  assignable \nothing;
      @*/
    static int m(int[] a) { ... }
}
```

Fortsetzung 6 Spezifikation mit der Java Modeling Language

b. Unten sehen Sie einen Teil der Implementierung eines Ringpuffers. Das ist eine Datenstruktur, die eine FIFO-Warteschlange mit beschränkter Kapazität mithilfe eines Arrays umsetzt, das zyklisch gelesen und beschrieben wird. Zum Anhängen an die Warteschlange gibt es eine Methode `put()`, zum Herausnehmen eine Methode `take()`. Hierbei zeigt `writeIndex` immer auf die Stelle im Array, an die als nächstes geschrieben wird, und `readIndex` auf das Element, das als nächstes gelesen (und entfernt) wird. Die Variable `dataCount` zählt die aktuell im Puffer befindlichen Elemente. Zwei mögliche Zustände eines solchen Ringpuffers sehen Sie hier (beispielhaft für die Größe 8):



i. Damit die Datenstruktur korrekt funktioniert, müssen die Indizes (`readIndex`, `writeIndex`) sowie `dataCount` innerhalb bestimmter Intervalle liegen. Formulieren Sie diese Einschränkungen als Objektinvarianten.

Hinweis: Es ist nicht notwendig, die drei Variablen untereinander in Beziehung zu setzen.

ii. Im Fall, dass der Puffer nicht leer ist, soll die Methode `take()` das älteste Element in der Queue zurückgeben, `dataCount` um eins verringern und außerdem `readIndex` „zyklisch“ weiterbewegen. Vervollständigen Sie den Methodenvertrag mit diesen Nachbedingungen, und geben Sie außerdem eine geeignete `assignable`-Klausel an!

```
class RingBuffer {
  final int[] a;    // data
  int readIndex;    // next index to read from
  int writeIndex;   // next index to write to
  int dataCount;    // needed to distinguish between full and empty ...

  //@ invariant ____________________________________________________________

  //@ invariant ____________________________________________________________

  //@ invariant ____________________________________________________________

  /*@ normal_behavior
    @  requires 0 < dataCount;

    @  ensures ____________________________________________________________

    @ ____________________________________________________________

    @ ____________________________________________________________

    @ ____________________________________________________________

    @ ____________________________________________________________

    @ ____________________________________________________________

    @  assignable ____________________________________________________________


    @*/
  int take() { ... }

  void put(int val) { ... }
}
```

7 Lineare Temporale Logik (LTL)
((1 + 1) + (1,5 + 1,5) + (1 + 1 + 1) = 8 Punkte)

Im Folgenden soll mithilfe von LTL das Verhalten eines Geräts mit zwei Lampen beschrieben werden. Eine Lampe ist rot, die andere grün. Dazu wird die Signatur $\Sigma = \{R, G\}$ verwendet:

R ist wahr gdw. die rote Lampe leuchtet

G ist wahr gdw. die grüne Lampe leuchtet

a. Übersetzen Sie folgende LTL-Formeln in natürliche Sprache:

i. $(\Diamond R) \rightarrow (\Box G)$

ii. $(R \,\mathbf{U}\, G) \wedge \Box R$

b. Formalisieren Sie die folgenden Aussagen in LTL.

i. Die rote Lampe leuchtet in genau einem Zustand.

ii. Immer wenn die grüne Lampe an ist, ist sie im nächsten Zustand aus und im übernächsten wieder an.

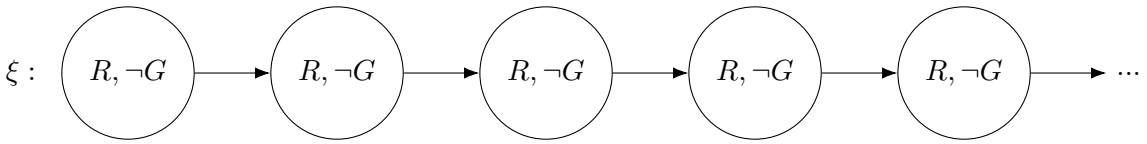
Fortsetzung 7 Lineare Temporale Logik

In der folgenden Teilaufgabe sind jeweils eine LTL-Formel F sowie eine Omega-Struktur ξ gegeben. Die Belegung der Omega-Struktur ändert sich nach dem 5. Zustand nicht mehr.

Geben Sie jeweils an, ob $\xi \models F$ gilt, und begründen Sie kurz.

c. **i.**

$$F \;=\; R \,\mathbf{U}\, G$$

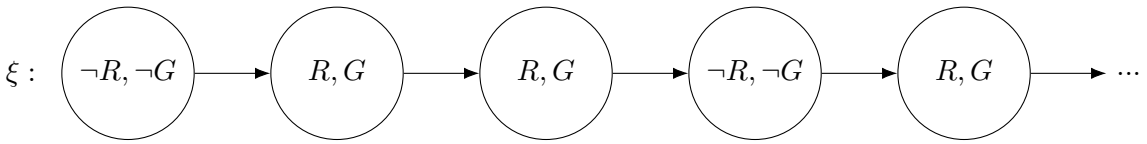


Gilt $\xi \models F$? Begründen Sie kurz.

---

ii.

$$F \;=\; \Box(R \rightarrow \mathbf{XXX}R)$$

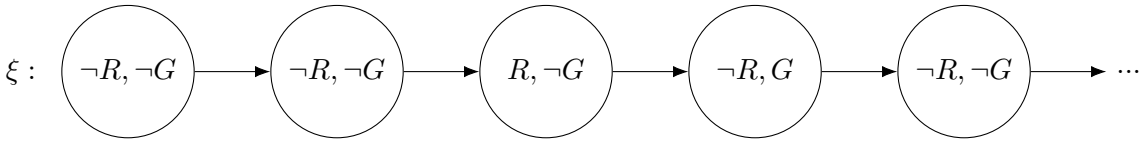


Gilt $\xi \models F$? Begründen Sie kurz.

---

iii.

$$F \;=\; \Diamond R \wedge \Diamond G$$



Gilt $\xi \models F$? Begründen Sie kurz.
