Karlsruher Institut für Technologie

Name:	____________
Vorname:	____________
Matrikel-Nr.:	____________

Klausur Formale Systeme

Fakultät für Informatik

WS 2022/23

Prof. Dr. Bernhard Beckert

28. Februar 2023

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (11)	A2 (9)	A3 (8)	A4 (7)	A5 (9)	A6 (9)	A7 (7)	Σ (60)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung ((1 + 1 + 2 + 1 + 1) + 5 = 11 Punkte)

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

i. Nennen Sie zwei Teilklassen des SAT-Problems, für die die Erfüllbarkeit in polynomieller Zeit entscheidbar ist.

ii. Wann ist eine Formel in Negationsnormalform?

iii. Geben Sie zwei Regeln des Tableaukalküls an. Demonstrieren Sie außerdem die Anwendung einer Regel des Kalküls auf die Formel $1\,(\forall x\ \exists y\ p(x,\ y))$ mit der aus der Vorlesung bekannten Notation, indem Sie die verwendete Regel und die resultierende(n) Formel(n) notieren.

iv. Aus der Vorlesung wissen Sie, dass die Peano-Arithmetik nicht identisch mit der Arithmetik natürlicher Zahlen ist, sie aber *approximiert*. Wie lautet die Teilmengenbeziehung zwischen der Menge P der gültigen Formeln der Peano-Arithmetik und der Menge A der gültigen Formeln der Arithmetik der natürlichen Zahlen? Gilt also $P \subset A$ oder $A \subset P$?
Begründen Sie.

v. Wann heißt eine Logik kompakt? Nennen Sie eine kompakte Logik.

Fortsetzung 1 Zur Einstimmung

b. Geben Sie den reduzierten Shannongraphen für

$$(A \vee B) \wedge (B \to D) \wedge C$$

mit der Variablenordnung

$$A < B < C < D$$

an.

2 Semantische Einschränkung von Termen ((2 + 2 + 2) + 3 = 9 Punkte)

Sei $\mathcal{M}$ eine Menge prädikatenlogischer Interpretationen über einer Signatur Σ, so dass alle Interpretationen $(D, I) \in \mathcal{M}$ dasselbe Universum D haben und sich nur in I unterscheiden.

Wir sagen, dass eine geschlossene prädikatenlogische Formel $F \in \mathit{Form}_\Sigma$ einen Grundterm $t \in \mathit{Term}^0_\Sigma$ *semantisch einschränkt*, wenn es ein $d \in D$ gibt, so dass:

1. es gibt eine Interpretation $(D, I) \in \mathcal{M}$ mit: $\quad val_{D,I}(t) = d$,
2. aber für jede Interpretation $(D, I) \in \mathcal{M}$ mit $(D, I) \models F$ gilt: $\quad val_{D,I}(t) \neq d$.

a. Es gelte nun

- $D = \mathbb{Z}$,
- $\Sigma = \{0, 1, +, \mathrm{mod}, >\} \cup \{c\}$, wobei c eine Konstante ist,
- $\mathcal{M}$ enthalte alle Interpretationen mit Universum $D = \mathbb{Z}$, in denen $0, 1, +, \mathrm{mod}, >$ so interpretiert werden, wie in der Arithmetik ganzer Zahlen üblich. Die Interpretationen in $\mathcal{M}$ unterscheiden sich also nur in der Interpretation von c.

Der Term t sei

$$t \quad := \quad c \;\mathrm{mod}\; (1 + 1)$$

Welche der folgenden Formeln schränken obigen Term t semantisch ein? Begründen Sie jeweils Ihre Antwort.

i. $c \doteq 1$

---

ii. $c > 0$

---

iii. $\exists x (c \doteq x + x)$

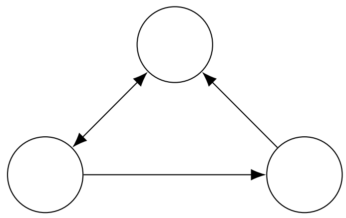
---

b. Gegeben seien ein Term t und zwei Formeln F und G, so dass t weder von F noch von G semantisch eingeschränkt wird. Ist es dann möglich, dass dennoch die Formel $F \wedge G$ den Term t einschränkt? Begründen Sie Ihre Antwort.

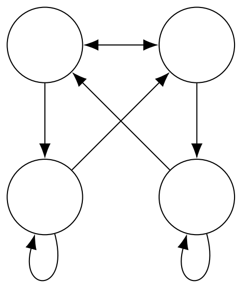
3 Modallogik ((2 + 2) + (1 + 3) = 8 Punkte)

a. Geben Sie für jede Welt in den folgenden Kripkestrukturen eine Belegung der aussagenlogischen Variablen A an, so dass die jeweils gegebene Formel in jeder Welt wahr ist.

i. $(\neg A \rightarrow \Diamond A) \;\wedge\; (A \rightarrow \Diamond\Diamond\neg A)$



ii. $(\Diamond\Box A) \;\wedge\; (\Diamond\Box\neg A)$



b. Zeichnen Sie zu jeder der folgenden modallogischen Formeln jeweils eine Kripkestruktur, so dass die Formel in **jeder** Welt der Kripkestruktur wahr ist.

Hinweis: Die Erreichbarkeitsbeziehung und die Wahrheitswerte der in der jeweiligen Formel vorkommenden aussagenlogischen Variablen müssen für jede Welt erkennbar sein.

i. $(\Box A) \wedge (\neg\Diamond A)$

ii. $(\neg B \rightarrow \Diamond B) \;\wedge\; (B \rightarrow \Diamond\neg B) \;\wedge\; (\neg A \rightarrow \Diamond\Diamond A) \;\wedge\; (A \rightarrow \Diamond\Diamond\neg A)$

4 Formalisieren in Prädikatenlogik (PL1) (1 + 2 + 2 + 2 = 7 Punkte)

Wir modellieren Method Frames. Das sind Bedingungen, die beschreiben, auf welche Zustandsvariablen einer Java-Klasse eine Java-Methode zugreifen darf. Während `assignable`-Klauseln in JML für eine konkrete Methode die konkreten zugreifbaren Variablen auflisten, betrachten wir hier in dieser Aufgabe allgemeinere Method-Frame-*Regeln*.

Gegeben sei dazu die prädikatenlogische Signatur $\Sigma = (\{m1, m2\}, \{met(\cdot), var(\cdot), acc(\cdot,\cdot)\}, \alpha)$. Sie enthält

- die Konstantensymbole $m1$ und $m2$,
- die einstelligen Prädikatensymbole var und met
- das zweistellige Prädikatensymbol acc

Zur Auswertung der Formeln werden nur solche Interpretationen (D, I) über Σ verwendet, in denen die Symbole wie erwartet interpretiert werden. Das heißt:

$I(m1)$	die Methode M1
$I(m2)$	die Methode M2
$I(met(x)) = W$	gdw. x ist eine Methode
$I(var(x)) = W$	gdw. x ist eine Zustandsvariable
$I(acc(x, y)) = W$	gdw x ist eine Methode, y ist eine Zustandsvariable und x darf auf y zugreifen

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

a. Es gibt mindestens eine Methode und mindestens eine Zustandsvariable.

b. Auf jede Zustandsvariable darf entweder M1 oder M2 zugreifen. Es gibt keine Zustandsvariablen, auf die beide Methoden zugreifen dürfen.

c. Manche Methoden dürfen auf keine Zustandsvariable zugreifen.

d. Es gibt eine Variable, auf die nur eine einzige Methode zugreifen darf.

5 Resolutionskalkül (5 + 3 + 1 = 9 Punkte)

Gegeben sei die Signatur Σ mit Konstanten- bzw. Funktionssymbolen $F_\Sigma = \{a, b, f, g\}$ und Prädikatsymbolen $P_\Sigma = \{p, q, r\}$ wobei $\alpha_\Sigma(a) = \alpha_\Sigma(b) = \alpha_\Sigma(r) = 0$, $\alpha_\Sigma(f) = \alpha_\Sigma(p) = 1$ und $\alpha_\Sigma(g) = \alpha_\Sigma(q) = 2$.

a. Betrachten Sie die folgende Formel über der Signatur Σ:

$$\forall x \left(\left(\forall y \left(\left(\forall z\, q(x, f(z)) \right) \to q(x, f(y)) \right) \right) \land \left(\exists y\, p(g(x, y)) \lor \neg r \right) \right)$$

Transformieren Sie die Formel in Skolem-Normalform.

Hinweis: Geben Sie in Ihrer Antwort geeignete Zwischenschritte an. Nicht alle Zwischenschritte sind notwendig, jedoch sollte der Lösungsweg klar erkennbar sein.

b. Betrachten Sie die folgende Klauselmenge C über der Signatur Σ und mit Variablen w_i, x_i, y_i, z_i $(1 \leq i \leq 5)$:

I $C_1 = \{\ q(x_1, f(x_1)),\quad p(x_1)\ \}$
II $C_2 = \{\ p(x_2),\quad r\ \}$
III $C_3 = \{\ \neg p(g(w_3, x_3)),\quad q(y_3, g(y_3, z_3))\ \}$
IV $C_4 = \{\ \neg q(x_4, f(g(y_4, z_4)))\ \}$
V $C_5 = \{\ \neg q(x_5, g(x_5, y_5))\ \}$

Zeigen Sie mit Hilfe des Resolutionskalküls für Prädikatenlogik, dass diese Klauselmenge unerfüllbar ist. Geben Sie dabei für jeden Resolutionsschritt die beteiligten Eltern-Klauseln, die Resolvente und den Unifikator an.
Hinweis: Es werden nicht alle Zeilen benötigt um $\Box$ abzuleiten.

VI

VII

VIII

IX

X

XI

XII

Fortsetzung 5 Resolutionskalkül

c. Geben Sie eine echte Teilmenge $C' \subset C$ an, die bereits unerfüllbar ist und begründen Sie warum C' unerfüllbar ist. Sie müssen keinen weiteren Resolutionsbeweis führen, ein Satz reicht als Begründung aus.

6 Spezifikation mit der Java Modeling Language

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

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

```
public final class A {
  /*@ normal_behavior
    @  requires x > 1;
    @  ensures  (\exists int r; 0 <= r && r <= x/2; r * r == x)
    @     ==> (0 <= \result && \result <= x/2 && \result * \result == x);
    @  ensures !(\exists int r; 0 <= r && r <= x/2; r * r == x)
    @     ==> \result == -1;
    @  assignable \nothing;
    @*/
  static int m(int x) { ... }
}
```

ii. Warum ist die Bereichseinschränkung für `\result` (unterstrichener Term oben) wichtig?

Fortsetzung 6 Spezifikation mit der Java Modeling Language

b. Unten sehen Sie die Implementierung und einen teilweise angegebenen Vertrag der Methode `indexOf`, die den kleinstmöglichen Index (beginnend bei `start`) des gegebenen Zeichens im übergebenen `char[]` zurückgibt, und -1, wenn das Zeichen nicht vorkommt. Der Code enthält eine Schleife. Geben Sie für diese eine möglichst starke Schleifenspezifikation an.

```
/*@ normal_behavior
  @  requires 0 <= start && start < str.length;
  @  ensures ...;
  @  assignable ...;
  @*/
static int indexOf(final char[] str, final char c, final int start) {
    int res = -1;

    /*@ loop_invariant ____________________

      @ ____________________

      @ ____________________

      @ ____________________

      @ ____________________

      @ ____________________

      @ ____________________

      @ ____________________

      @ decreases ____________________

      @ assignable ____________________

      @*/
    for (int i = start; i < str.length; i++) {
        if (str[i] == c && res == -1) {
            res = i;
        }
    }
    return res;
}
```

7 Lineare Temporale Logik (LTL) und Büchiautomaten (1 + 1 + 1) + 4 = 7 Punkte)

a. Formalisieren Sie folgenden Sachverhalt zur exklusiven Zugriffskontrolle mit zwei Prozessen P_1, P_2 in LTL. Verwenden Sie dabei folgende Signatur:

T_i	Prozess P_i befindet sich in der Anmeldephase
C_i	Prozess P_i befindet sich in einer kritischen Region

i. Die Prozesse P_1 und P_2 sind niemals gleichzeitig in der kritischen Region.

---

ii. Nur direkt anschließend an eine vorhergehende Anmeldephase darf man sich in der kritischen Region befinden (Formel für Prozess P_1 reicht).

---

iii. Jeder Prozess muss die kritische Region irgendwann auch wieder verlassen.

---

b. Geben Sie einen nicht-deterministischen Büchi-Automaten an, dessen akzeptierte Sprache den Modellen (ω-Wörtern) der LTL-Formel

$$\neg\big((\Diamond a) \leftrightarrow (\Box a)\big)$$

über der Signatur $\Sigma = \{a\}$ entspricht.

Für das Vokabular $V = \mathbb{P}(\Sigma)$ (Potenzmenge von Σ) werden die folgenden, aus der Vorlesung bekannten, Abkürzungen definiert:

$$A = \{M \in V \mid a \in M\} \subset V \qquad \bar{A} = \{M \in V \mid a \notin M\} \subset V$$

Notizen/Schmierpapier — Sollen Ihre Notizen bewertet werden, ist eine klare Zuordnung notwendig (Verweis in der ursprünglichen Aufgabe, sowie klare Aufgabennummer in den Notizen).