Karlsruher Institut für Technologie

Name: _____

Vorname: _____

Matrikel-Nr.: _____

Klausurnummer:

_____

Klausur Formale Systeme

Fakultät für Informatik

SS 2024

Prof. Dr. Bernhard Beckert

03. September 2024

Die Bearbeitungszeit beträgt 60 Minuten.

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

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung (1 + 3 + 3 + 2 = 9 Punkte)

a. Was sagt der Kompaktheitssatz über eine unerfüllbare Menge M von Formeln aus?

b. Welche der folgenden Sachverhalte lassen sich als prädikatenlogische Formel erster Stufe mit Gleichheit formalisieren? Geben Sie jeweils eine prädikatenlogische Formel an oder geben Sie eine kurze mathematische Begründung an, warum die Aussage nicht formalisiert werden kann:

i. Die Formel charakterisiert genau die Modelle, deren Universum genau ein Element hat.

ii. Die Formel charakterisiert genau die Modelle, deren Universum mindestens zwei Elemente hat.

iii. Die Formel charakterisiert genau die Modelle, deren Universum endlich viele Elemente hat.

Fortsetzung 1 Zur Einstimmung

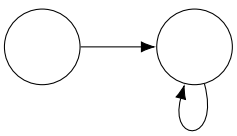
c. Gegeben sei die folgende aussagenlogische Formel:

$$(\neg P \vee \neg Q \vee \neg R \vee \neg S) \ \wedge \ (\neg Q \vee P) \ \wedge \ (\neg P \vee R) \ \wedge \ (\neg P \vee S) \ \wedge \ Q$$

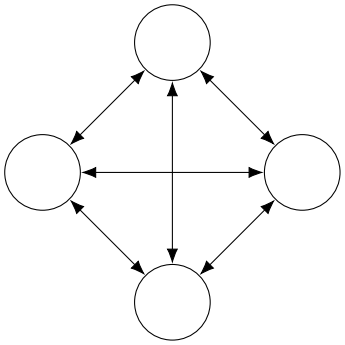
Sie sollen eine erfüllende Belegung für die Formel finden bzw. deren Unerfüllbarkeit zeigen. Nennen Sie jenes Verfahren aus der Vorlesung, das die beste Worst-Case-Komplexität hat und auf diese Formel anwendbar ist. Nutzen Sie dieses Verfahren, um die (Un-)Erfüllbarkeit der Formel zu zeigen. Notieren Sie geeignete Zwischenschritte des Verfahrens, um die Funktionsweise des Algorithmus verständlich zu machen.

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

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



ii. $\Diamond A \rightarrow \Box\neg A$



2 Varianten der Linear Temporal Logic (3 + 2 + 4 = 9 Punkte)

In der *Linear Temporal Logic* (LTL), wie sie in der Vorlesung eingeführt wurde, gibt es die temporallogischen Operatoren $\Box$, $\Diamond$, $\mathbf{U}$ (*until*), X (*next*). Wir führen nun zusätzliche Operatoren ein:

$\Diamond_{\leq n} \quad \Diamond_{\geq n} \qquad$ (für $n \in \mathbb{N}_0$)

Deren Semantik ist:

- $\Diamond_{\leq n}\phi$: ϕ wird spätestens im n-ten Zustand ab jetzt wahr
- $\Diamond_{\geq n}\phi$: ϕ wird in einem Zustand wahr, der n oder mehr Schritte in der Zukunft liegt.

a. Geben Sie eine zu

$$\Diamond_{\geq 3}\phi$$

äquivalente Formel an, die nur die Standardoperatoren $\Box, \Diamond, \mathbf{U}, X$ (und $\neg, \wedge, \vee$) verwendet.

b. Wir definieren zusätzlich den Operator $\Box_{\geq n}$, dessen Semantik durch folgende Äquivalenz festgelegt ist:

$$\Box_{\geq n}\phi \quad \leftrightarrow \quad \neg\Diamond_{\geq n}\neg\phi$$

Geben Sie die Bedeutung von $\Box_{\geq n}\phi$ in natürlicher Sprache wieder.

c. Betrachten wir nun eine Variante von LTL, in der $\Diamond_{\leq n}$ der **einzige** temporallogische Operator ist. Kann man in dieser Logik eine Formel angeben, die zu $\Box\phi$ äquivalent ist? ($\Box$ ist der Standard-LTL-Operator.)

Begründen Sie Ihre Antwort.

3 Shannon-Graphen (2 + 6 = 8 Punkte)

a. Geben Sie eine aussagenlogische Formel F über der Signatur $\Sigma = \{A, B, C, Z\}$ an, die einen 1-Bit-Volladdierer modelliert. Die Formel soll die aussagenlogischen Variablen A, B, C, Z enthalten und genau dann wahr sein, wenn:

Z ist die Summe, also das niederwertige Bit der Addition $A + B$, und
C ist der Übertrag, also das höherwertige Bit der Addition $A + B$.

Hinweis: Nutzen Sie den Äquivalenz-Operator ($\leftrightarrow$).

b. Geben Sie einen *reduzierten* Shannongraphen für F an. Nutzen Sie die Variablenreihenfolge A, B, C, Z.

4 Formalisieren in Prädikatenlogik (PL1)

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

Im Folgenden soll mithilfe von Prädikatenlogik ein Graph mit Knoten und Kanten formalisiert werden, wobei Knoten Beschriftungen ("Labels") haben können.

Hierzu sei die prädikatenlogische Signatur $\Sigma = (\{vert, edge, path\}, \{root, child, label\}, \alpha)$ gegeben. Sie enthält das einstellige Prädikatensymbol $vert(\cdot)$, die zweistelligen Prädikatensymbole $edge(\cdot, \cdot)$ und $path(\cdot, \cdot)$, sowie das einstellige Funktionssymbol $label(\cdot)$ und die beiden Konstanten *root* und *child*.

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

- das Universum D eine Menge von Knoten und Labels ist,
- die Konstanten *root* und *child* Labels sind,
- das Prädikat $vert(x)$ genau dann wahr ist, wenn x ein Knoten ist,
- Prädikat $edge(x, y)$ genau dann wahr ist, wenn eine Kante von Knoten x nach Knoten y führt,
- Prädikat $path(x, y)$ genau dann wahr ist, wenn ein Pfad (möglicherweise über mehrere Kanten) von Knoten x zu Knoten y führt, und
- die Funktion $label(x)$ einen Knoten x auf ein Label abbildet.

Einen Knoten, den die Funktion $label(\cdot)$ auf das Label *root* bzw. *child* abbildet, nennen wir im Folgenden Wurzel- bzw. Kindknoten.

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

a. Von jedem Knoten startet eine Kante.

b. Jeder Pfad führt genau dann von einem Knoten a zu einem Knoten b, wenn a und b der selbe Knoten sind, eine Kante von a nach b führt, oder ein Pfad von a zu einem Knoten führt, von dem eine Kante nach b führt.

c. Jeder Pfad und jede Kante, die von einem Kindknoten x ausgehen, führen zu x selbst.

d. Von jedem Wurzelknoten führen Pfade zu mindestens zwei verschiedenen Kindknoten.

5 Sequenzenkalkül (2 + 2 + 6 = 10 Punkte)

Hinweis: Auf Blatt 12 finden Sie die Regeln des Sequenzenkalküls. Sie dürfen diese Seite von der Klausur abtrennen!

a. Der folgende Sequenzenbeweis ist falsch. Kreisen Sie die falsche Regelanwendung ein und erklären Sie das Problem.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p(c) \Rightarrow p(c)}\,(\text{axiom})}{\exists x.p(x) \Rightarrow p(c)}\,(\text{ex-left})}{\exists x.p(x) \Rightarrow \forall y.p(y)}\,(\text{all-right})}{\Rightarrow (\exists x.p(x)) \rightarrow (\forall y.p(y))}\,(\text{imp-right})$$

b. Auf Blatt 12 finden Sie die Regeln des Sequenzenkalküls, allerdings fehlt eine Regel. Geben Sie die (or-left) Regel an. Nutzen Sie die gleiche Notation wie auf Blatt 12.

c. Zeigen Sie mit Hilfe des Sequenzenkalküls die Allgemeingültigkeit der unten stehenden aussagenlogischen Formel. Geben Sie jeweils die angewandte Regel an.

$$\dfrac{}{\Rightarrow (A \wedge B) \vee \neg (C \wedge (C \rightarrow (A \wedge B)))}$$

6 Spezifikation mit der Java Modeling Language

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

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

```
public final class A {
    char[] a;

    /*@ normal_behavior
      @  requires 0 <= i && i < a.length;
      @  requires 0 <= j && j < a.length;
      @  ensures (forall int k; 0 <= k && k < a.length;
      @                         k != i && k != j  ==>  a[k] == \old(a[k]));
      @  ensures a[i] == \old(a[j]);
      @  ensures a[j] == \old(a[i]);
      @  assignable a[*];
      @*/
    void m(int i, int j) {
     ...
}
```

Fortsetzung 6 Spezifikation mit der Java Modeling Language

b. Unten sehen Sie eine partielle Implementierung eines sortierten binären Suchbaums.

i. Wie müssen die Felder für Kindknoten (`left` und `right`) annotiert werden, damit die Spezifikation sinnvoll ist? Beachten Sie diesen Spezialfall auch bei den nachfolgenden Invarianten!

Spezifizieren Sie Invarianten (ii) bis (iv) in JML. Verwenden Sie dafür die Methode `reachable`. Diese implementiert ein Prädikat, das genau dann wahr ist, wenn `target` von `start` aus in genau `steps` Schritten erreichbar ist. Wenn `target` oder `start` dabei `null` sind oder `steps < 0` ist, ist der Rückgabewert `false`. Die Methode ist `pure`, ändert also den Heap-Zustand nicht und darf daher in der Spezifikation verwendet werden.

ii. Es gibt keine Zyklen, der aktuelle Knoten `this` ist also nicht von sich selbst aus erreichbar (mit mindestens einem Schritt).

iii. Der Graph ist azyklisch, d.h. kein Knoten kann sowohl über `left` als auch über `right` erreicht werden.

iv. Der Baum ist sortiert, d.h. Werte `d` aller Knoten, die über `left` erreichbar sind, sind kleiner gleich dem Wert von `this`. Die entsprechende Formel für `right` brauchen Sie hier **nicht** anzugeben.
Hinweis: Achten Sie darauf, dass Ihre Invariante auch Fälle wie `this.left.right.d <= this.d` garantiert.

```
final class Node  {
    int d;                                          // data

    /*@ ____________________ @*/ Node left;     // left subtree

    /*@ ____________________ @*/ Node right;    // right subtree


    //@ invariant ____________________________________________________

    //@ ____________________________________________________


    //@ invariant ____________________________________________________

    //@ ____________________________________________________

    //@ ____________________________________________________


    //@ invariant ____________________________________________________

    //@ ____________________________________________________

    //@ ____________________________________________________


    boolean /*@ pure @*/ reachable(Node target, Node start, int steps) { ... }

    ...
}
```

7 Lineare Temporale Logik (LTL) und Büchiautomaten ((1 + 2) + (1 + 3) = 7 Punkte)

Im Folgenden soll mithilfe von LTL das Verhalten einer Maschine beschrieben werden. Die Maschine hat einen Motor und eine Lampe. Dazu wird die Signatur $\Sigma = \{m, l\}$ verwendet:

m ist wahr gdw. der Motor läuft

l ist wahr gdw. die Lampe leuchtet

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

$$M = \{X \in V \mid m \in X\} \subset V \qquad L = \{X \in V \mid l \in X\} \subset V$$
$$\overline{M} = \{X \in V \mid m \notin X\} \subset V \qquad \overline{L} = \{X \in V \mid l \notin X\} \subset V$$

a. Gegeben sei die LTL-Formel

$$F = \quad m \wedge (m \,\mathbf{U}\, (\Box\neg m))$$

i. Übersetzen Sie F in natürliche Sprache.

ii. Geben Sie einen nicht-deterministischen Büchi-Automaten an, dessen akzeptierte Sprache den Modellen (ω-Wörtern) von F über der Signatur Σ entspricht.

Fortsetzung 7 Lineare Temporale Logik und Büchiautomaten

b. Gegeben sei die folgende Aussage in natürlicher Sprache:
"Immer wenn der Motor läuft, leuchtet die Lampe entweder zu diesem oder zu einem zukünftigen Zeitpunkt."

i. Formalisieren Sie diese Aussage in LTL.

ii. Geben Sie einen nicht-deterministischen Büchi-Automaten an, dessen akzeptierte Sprache der obigen Aussage entspricht.

Matr.-Nr.:

Hilfestellung: Regeln des Sequenzenkalkül

Hinweis: Sie dürfen diese Seite von der Klausur ablösen.

$$\frac{}{\Gamma,\ F \Rightarrow F,\ \Delta}\ \text{(axiom)}$$

$$\frac{\Gamma,\ F,\ G \Rightarrow \Delta}{\Gamma,\ F \wedge G \Rightarrow \Delta}\ \text{(and-left)} \qquad \frac{\Gamma \Rightarrow \Delta,\ F \qquad \Gamma \Rightarrow \Delta,\ G}{\Gamma \Rightarrow \Delta,\ F \wedge G}\ \text{(and-right)}$$

$$\frac{\Gamma \Rightarrow \Delta,\ F,\ G}{\Gamma \Rightarrow \Delta, F \vee G}\ \text{(or-right)}$$

$$\frac{\Gamma \Rightarrow \Delta, F}{\Gamma,\ \neg F \Rightarrow \Delta}\ \text{(not-left)} \qquad \frac{\Gamma,\ F \Rightarrow \Delta}{\Gamma \Rightarrow \Delta,\ \neg F}\ \text{(not-right)}$$

$$\frac{\Gamma \Rightarrow \Delta,\ F \qquad \Gamma,\ G \Rightarrow \Delta}{\Gamma,\ F \to G \Rightarrow \Delta}\ \text{(impl-left)} \qquad \frac{\Gamma,\ F \Rightarrow \Delta,\ G}{\Gamma \Rightarrow \Delta,\ F \to G}\ \text{(impl-right)}$$

$$\frac{\Gamma,\ \forall x F,\ \{x/X\}\, F \Rightarrow \Delta}{\Gamma,\ \forall x F \Rightarrow \Delta}\ \text{(all-left)}$$

wobei X eine neue Variable

$$\frac{\Gamma, \Rightarrow \Delta, \{x/f\,(\overline{x})\}\, F}{\Gamma \Rightarrow \Delta,\ \forall x F}\ \text{(all-right)}$$

wobei f neues Funktionssymbol
und $\overline{x} = x_1, \ldots, x_n$
alle freien Variablen in $\forall x F$

$$\frac{\Gamma,\ \{x/f\,(\overline{x})\}\, F \Rightarrow \Delta}{\Gamma,\ \exists x F \Rightarrow \Delta}\ \text{(ex-left)}$$

wobei f neues Funktionssymbol
und $\overline{x} = x_1, \ldots, x_n$
alle freien Variablen in $\forall x F$

$$\frac{\Gamma, \Rightarrow \Delta,\ \exists x F,\ \{x/X\}\, F}{\Gamma \Rightarrow \Delta,\ \exists x F}\ \text{(ex-right)}$$

wobei X eine neue Variable