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:

Matr.-Nr.:

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

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

Eine (unendliche) Menge von Formeln ist genau dann unerfüllbar, wenn sie eine endliche Teilmenge hat, die unerfüllbar ist.

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.

$\exists y \forall x \; x \doteq y$

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

$\exists x_1 \exists x_2 \; \neg(x_1 \doteq x_2)$

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

Dies ist wegen des Kompaktheitssatz nicht in Prädikatenlogik erster Stufe formalisierbar.
Nicht vollständig erwartet:
Angenommen, es gäbe eine Formel $F_{<\infty}$, die diese Aussage formalisiert. Wir wissen, dass wir die Aussage, dass das Universum mindestens n Elemente hat, mit einer Formel $F_{\geq n}$ ausdrücken können. Alle *endlichen* Teilmengen von $F_{<\infty} \cup \bigcup_{n \in \mathbb{N}} F_{\geq n}$ sind erfüllbar (wähle ein ausreichend großes Universum). Aber $F_{<\infty} \cup \bigcup_{n \in \mathbb{N}} F_{\geq n}$ ist unerfüllbar. Dies widerspricht dem Kompaktheitssatz. Daher muss die Annahme falsch sein, und es kann kein solches $F_{<\infty}$ geben.

Fortsetzung 1 Zur Einstimmung

c. Gegeben sei die folgende aussagenlogische Formel:

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

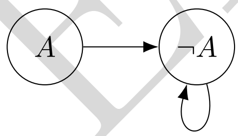
Sie sollen eine erfüllende Belegung für die Formel finden bzw. deren Unerfüllbarkeit zeigen. Nennen Sie jenes Verfahren aus der Vorlesung, das die beste Worst-Case-Komplexität hat und auf diese Formel anwendbar ist. Nutzen Sie dieses Verfahren, um die (Un-)Erfüllbarkeit der Formel zu zeigen. Notieren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

Die Formel ist eine Horn-Formel. Daher können wir mittels des Markierungsalgorithmus die Erfüllbarkeit prüfen.

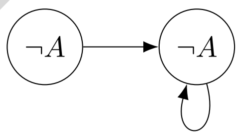
- Markiere Q
- Markiere P
- Markiere R und S
- $(\neg P \vee \neg Q \vee \neg R \vee \neg S) \equiv P \wedge Q \wedge R \wedge S \to \mathbf{0}$:
  Alle Variablen im Rumpf markiert $\Rightarrow$ unerfüllbar.

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.

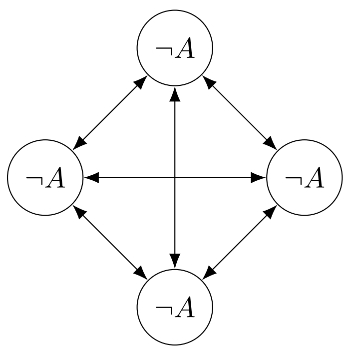
i. $(A \to \Box\neg A) \;\wedge\; (\neg A \to \Box\neg A)$



oder



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



Matr.-Nr.:

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:

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

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.

$\boxed{XXX\Diamond\phi}$

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.

In allen Zuständen, die n oder mehr Schritte in der Zukunft liegen, ist ϕ wahr.

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.

Das ist nicht möglich, wie folgende Überlegung zeigt:
In jeder Formel ψ dieser LTL-Variante kann es nur endlich viele Vorkommen von $\Diamond_{\leq n}$ geben. Sei k die Anzahl dieser Vorkommen in ψ, und sei n_{max} das maximale n, so dass $\Diamond_{\leq n}$ vorkommt. Der Wahrheitswert der Formel ψ kann nicht von der Belegung der Variablen in den Zuständen abhängen, die mehr als $k \cdot n_{max}$ Schritte in der Zukunft liegen (wie sich per Induktion über den Formelaufbau zeigen lässt). Der Wahrheitswert von $\Box\phi$ hängt dagegen (auch) von diesen späteren Zuständen ab, so dass es keine zu $\Box\phi$ äquivalente Formel ψ geben kann. (Zur Lösung der Aufgabe ist als Begründung hinreichend, dass der Wahrheitswert von Formeln in dieser LTL-Variante immer nur von einem endlichen Anfangsstück der Zustandsfolge abhängen kann.)

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

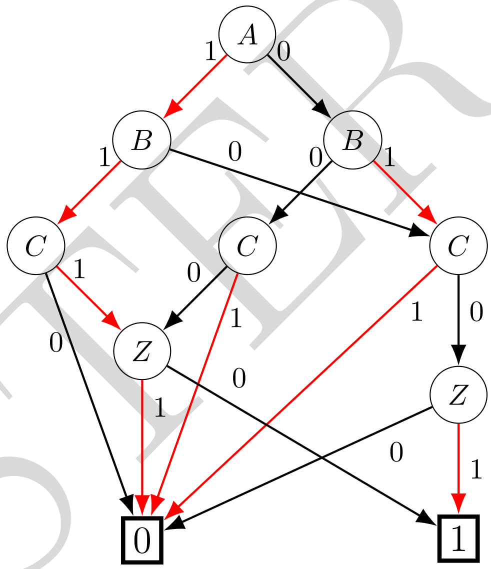
a. Geben Sie eine aussagenlogische Formel F über der Signatur $\Sigma = \{A, B, C, Z\}$ an, die einen 1-Bit-Volladdierer modelliert. Die Formel soll die aussagenlogischen Variablen A, B, C, Z enthalten und genau dann wahr sein, wenn:

> 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 ($\leftrightarrow$).

$$(Z \leftrightarrow ((\neg A \land B) \lor (A \land \neg B))) \ \land \ (C \leftrightarrow (A \land B))$$

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



Matr.-Nr.:

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.

$$\boxed{\forall x\ \exists y\big(vert(x) \to vert(y) \land edge(x,y)\big)}$$

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.

$$\forall x\ \forall y\Bigg(vert(x) \land vert(y) \to \Big(path(x,y) \leftrightarrow \Big(x \doteq y \lor edge(x,y) \lor \exists z\big(vert(z) \land path(x,z) \land edge(z,y)\big)\Big)\Big)\Bigg)$$

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

$$\forall x\Big(vert(x) \land label(x) \doteq child \to \forall y\Big(vert(y) \land (edge(x,y) \lor path(x,y)) \to x \doteq y\Big)\Big)$$

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

$$\forall x\Big(vert(x) \land label(x) \doteq root \to \exists y\ \exists z\big(vert(y) \land vert(z) \land \neg y \doteq z \land label(y) \doteq child \land label(z) \doteq child \land path(x,y) \land path(x,z)\big)\Big)$$

Matr.-Nr.:

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.

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

Die Anwendung von (ex-left) ist falsch. x wurde nicht mit einer frischen Konstante initialisiert.

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.

$$\dfrac{\Gamma,\ F \Rightarrow \Delta \qquad \Gamma,\ G \Rightarrow \Delta}{\Gamma,\ F \lor G \Rightarrow \Delta}\text{(or-left)}$$

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

$$\dfrac{\dfrac{\dfrac{\dfrac{\text{(axiom)}\dfrac{*}{C \Rightarrow C,\ (A\land B)} \qquad \dfrac{*}{C,\ (A\land B) \Rightarrow (A\land B)}\text{(axiom)}}{C,\ (C\to(A\land B)) \Rightarrow (A\land B)}\text{(impl-left)}}{C\land(C\to(A\land B)) \Rightarrow (A\land B)}\text{(and-left)}}{\Rightarrow (A\land B)\,,\ \neg(C\land(C\to(A\land B)))}\text{(not-right)}}{\Rightarrow (A\land B)\lor\neg(C\land(C\to(A\land B)))}\text{(or-right)}$$

Matr.-Nr.:

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) {
     ...
}
```

Wenn beim Aufruf von m die Parameter i und j gültige Indizes für das char-Array a sind, terminiert m ohne Exception und verändert höchstens die Arrayeinträge von a auf dem Heap (tatsächlich wird das durch die Nachbedingungen noch verfeinert). Dabei gilt: Nach Ausführung der Methode sind die Einträge i und j getauscht und alle anderen Einträge unverändert im Vergleich zum Vorzustand (swap).

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

    /*@ nullable @*/ Node left;                 // left subtree

    /*@ nullable @*/ Node right;                // right subtree

    //@ invariant !(\exists int k; 0 < k; reachable(this, this, k));  // no cycles

    /*@ invariant !(\exists Node node;                                // not a DAG
      @                (\exists int k; 0 <= k; reachable(node, left, k)
      @             && (\exists int l; 0 <= l; reachable(node, right, l))));
      @*/

    /*@ invariant (\forall Node node;                                 // sorted
      @                  (\exists int k; 0 <= k; reachable(node, left, k))
      @              ==> node.d <= this.d);
      @*/


    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

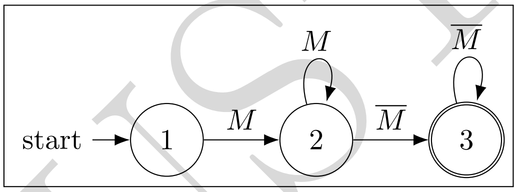
$$F = \quad m \wedge (m \,\mathbf{U}\, (\Box \neg m))$$

i. Übersetzen Sie F in natürliche Sprache.

Der Motor läuft. Irgendwann hört er auf zu laufen.

Wenn der Motor einmal aus ist, läuft er ab da nicht mehr.

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



Matr.-Nr.:

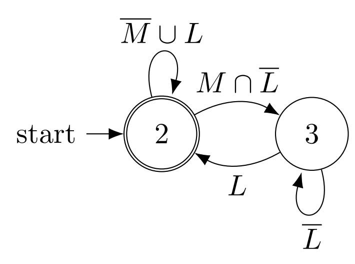
Fortsetzung 7 Lineare Temporale Logik und Büchiautomaten

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.

$\boxed{\Box(m \rightarrow \Diamond l)}$

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.

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

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

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

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

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

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

wobei X eine neue Variable

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

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

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

wobei X eine neue Variable