Karlsruher Institut für Technologie

Vorname:	**Vorname**
Name:	**Familienname**
Matrikel-Nr.:	**Matr.-Nr.**
Hörsaal:	**Hörsaal** **Sitzplatz**
Code:	**Nonce**

Klausur Formale Systeme

Fakultät für Informatik

SS 2021

Prof. Dr. Bernhard Beckert

04. August 2021

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (14)	A2 (9)	A3 (5)	A4 (8)	A5 (8)	A6 (9)	A7 (7)	Σ (60)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung ((2+3) + 5 + 4 = 14 Punkte)

a. Seien p, q einstellige Prädikatensymbole, c ein nullstelliges Funktionssymbol (Konstante) und f ein einstelliges Funktionssymbol.

Geben Sie für folgende prädikatenlogische Formeln – **falls möglich** – jeweils zwei Interpretationen I über dem Universum $D = \{a, b\}$ an, und zwar jeweils

- eine Interpretation, in der die Formel **wahr** ist, und
- eine Interpretation, in der die Formel **falsch** ist.

In den Fällen, in denen eine Interpretation mit der gesuchten Eigenschaft **nicht existiert**, geben Sie dies an.

Hinweis: Es muss explizit angegeben werden, wenn eine passende Interpretation nicht existiert (schreiben Sie „existiert nicht“ neben den Kasten). Es genügt nicht, die Beschreibung der Interpretation leer zu lassen.

i. $\forall y \exists x (p(y) \to (q(x) \to \neg p(c)))$

Interpretation, in der die Formel wahr ist:

$I(c) = b$	$I(p)(a) = \mathbf{F}$	$I(p)(b) = \mathbf{W}$
	$I(q)(a) = \mathbf{F}$	$I(q)(b) = \mathbf{F}$

Interpretation, in der die Formel falsch ist:

$I(c) = a$	$I(p)(a) = \mathbf{W}$	$I(p)(b) = \mathbf{W}$
	$I(q)(a) = \mathbf{W}$	$I(q)(b) = \mathbf{W}$

ii. $\big(\forall x\ p(x) \leftrightarrow \neg p(f(x))\big) \to \neg p(c)$

Interpretation, in der die Formel wahr ist:

$I(p)(a) = \mathbf{W}$	$I(p)(b) = \mathbf{W}$	
$I(f)(a) = \quad a$	$I(f)(b) = \quad a$	$I(c) = \quad a$

Interpretation, in der die Formel falsch ist:

$I(p)(a) = \mathbf{W}$	$I(p)(b) = \mathbf{F}$	
$I(f)(a) = \quad b$	$I(f)(b) = \quad a$	$I(c) = \quad a$

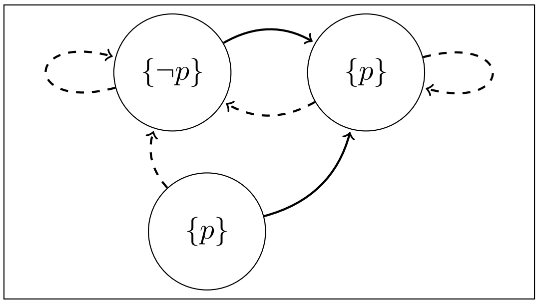
Fortsetzung 1 Zur Einstimmung

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

i. Was ist eine Herbrand-Struktur?

Eine Struktur deren Universum die Menge der Grundterme ist, und in der alle Grundterme als sie selbst interpretiert werden.

ii. **Ergänzen** Sie die Erreichbarkeitsrelationen in der unten gegebenen Kripestruktur $\mathcal{K}$ über $\Sigma = \{p\}$, sodass $\mathcal{K} \models (\Diamond\neg p \wedge \Diamond p)$ gilt.



iii. Geben Sie einen allgemeinsten Unifikator für
$g(h(x), f(c), g(h(x)))$ und
$g(h(f(y)), f(d), g(y))$ an, falls es einen gibt.
(Funktionssymbole: g, h, f; Variablen: x, y; Konstanten: c, d)

Nicht möglich. Unifikation von zwei Konstanten.

iv. Wann heißt eine Theorie T konsistent?

$T \not\models \mathbf{0}$

v. Geben Sie zu der Formel $sh(B, C, \mathbf{1})$ eine äquivalente Formel in disjunktiver Normalform (DNF) an.

$(\neg B \wedge C) \vee B$

c. Zeigen Sie die Unerfüllbarkeit bzw. Erfüllbarkeit folgender Hornklauselmenge über den aussagenlogischen Variablen $\{A, B, C, D, E, F\}$ mittels des Markierungsalgorithmus:

- $E \to F$
- $A \wedge C \to E$
- $C \wedge B \to A$
- A
- $A \wedge B \to C$
- $A \wedge F \to E$
- $F \to \mathbf{0}$
- C

i. Geben Sie Ihre Markierungsreihenfolge der Variablen an:

A, C, E, F

ii. Geben Sie ein Modell der Formelmenge aus Ihren Markierungen an oder begründen Sie die Unerfüllbarkeit:

Unerfüllbar: F ist markiert aber $F \to \mathbf{0}$

2 Modellierung von Fairness in LTL (2 + 3 + (2 + 2) = 9 Punkte)

Beim Scheduling unterscheidet man häufig zwei Arten von Fairness:

Schwache Fairness: Es gilt immer, dass eine Aktion, die **hinreichend (unbeschränkt) lange** ausführbar (waiting) ist, irgendwann ausgeführt wird (executing).

Starke Fairness: Es gilt immer, dass eine Aktion, die **hinreichend (unbeschränkt) oft** ausführbar (waiting) ist, irgendwann ausgeführt wird (executing).

Intuitiv ist der Unterschied zwischen beiden Begriffen, dass schwache Fairness ununterbrochenes Warten verlangt, während starke Fairness Unterbrechungen beim Warten zulässt.

a. Schwache Fairness kann in Linearer Temporaler Logik (LTL) beispielsweise wie folgt modelliert werden:

$$\Box(\ \mathit{Waiting}\ \rightarrow\ (\Diamond \mathit{Executing} \vee \Diamond\neg \mathit{Waiting})\ )$$

Diese Formel lässt sich vereinfachen. Geben Sie eine einfachere, äquivalente Formel an.

Hinweis: „Einfacher“ ist hier syntaktisch zu verstehen (weniger Symbole, geringere Schachtelungstiefe); ob die einfachere Formel auch besser verständlich ist, soll hier keine Rolle spielen.

$\Box(\ \Box \mathit{Waiting} \rightarrow \Diamond \mathit{Executing}\ )$ oder äquivalent
$\Box(\ \Diamond \mathit{Executing} \vee \Diamond\neg \mathit{Waiting}\ )$ oder $\Box\Diamond \mathit{Executing} \vee \Box\Diamond\neg \mathit{Waiting}$ oder $\Box\Diamond(\ \mathit{Waiting} \rightarrow \mathit{Executing})$
oder $\Box(\mathit{Waiting} \rightarrow \Diamond(\ \mathit{Waiting} \rightarrow \mathit{Executing}))$ oder $\Box(\mathit{Waiting}\ \mathbf{U}\ (\mathit{Waiting} \rightarrow \mathit{Executing}))$

b. Geben Sie eine entsprechende LTL-Formel an, die starke Fairness modelliert.

$\Box(\ \Box\Diamond \mathit{Waiting} \rightarrow \Diamond \mathit{Executing})\ )$ oder auch $\Box\Diamond \mathit{Executing} \vee \Diamond\Box\neg \mathit{Waiting}$

c. Betrachten wir das Beispiel einer Telefon-Hotline.

i. Welche Art der Fairness (starke oder schwache) können Anrufer gewöhnlich von einer Telefon-Hotline erwarten? Begründen Sie Ihre Antwort.

Schwache Fairness. Man kommt dran, wenn man lange genug kontinuierlich in der Leitung wartet. Wenn man zwischendurch auflegt und neu anruft, kommt man ggf. nie dran.

ii. Wie müsste eine Telefon-Hotline funktionieren, um die andere Art von Fairness zu implementieren?

Um starke Fairness zu implementeiren, müsste die Hotline sich merken, wer angerufen hat, ohne dranzukommen. Dann müsste das Scheduling auch die Anzahl vergeblicher Anrufe berücksichtigen bzw. die Wartezeit über die Anrufe hinweg aufsummieren.

3 Entscheidungsverfahren für uninterpretierte Funktionssymbole (1 + 2 + 2 = 5 Punkte)

Gegeben sei die Signatur, die nur die beiden einstelligen Funktionssymbole f und g und die beiden Konstanten a und b enthält.

Die folgende Formel ist zu untersuchen:

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

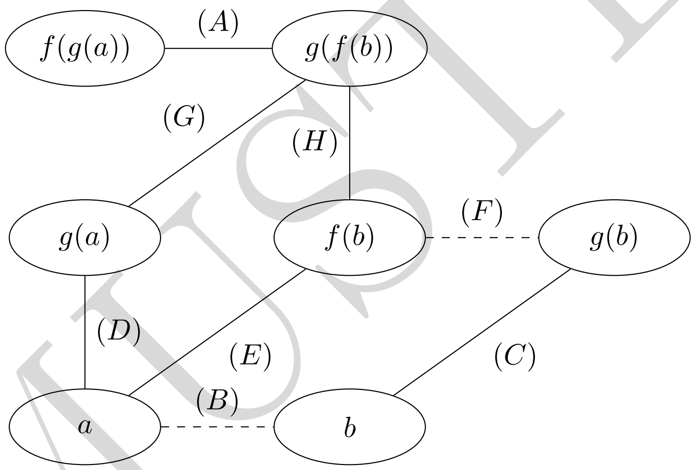
Wenden Sie nun den Algorithmus nach *Shostak* auf die Formel (1) an, indem Sie den Kongruenzgraphen (wie in der Vorlesung/Übung) erstellen. Durchgezogene Kanten stehen für Gleichheiten und gestrichelte Kanten für Ungleichheiten.

a. Tragen Sie die Informationen aus (1) als Kanten zwischen den untenstehenden Knoten ein. Beschriften Sie die Kanten mit (A) bis (F).

b. Ergänzen Sie den Graphen um weitere Kanten gemäß dem Algorithmus. Benennen Sie diese Kanten mit (G), (H) usw. Tragen Sie für jede hinzugefügte Kante im Graphen in die Tabelle ein:

1. welchen Typs die neue Kante ist: „T" für Transitive Kante, „K" für Kongruenzkante und
2. welche bisherigen Kante(n) die neue Kante begründet.

Es gibt Lösungen, die nicht die volle Anzahl an Tabellenzeilen benötigen.



Name	*Typ*	*Begründung*
(G)	$\boxed{\text{K}}$	$\boxed{(E)}$
(H)	$\boxed{\text{T}}$	$\boxed{(D), (E), (G)}$
(I)		
(K)		

c. Lesen Sie aus dem fertigen Graphen ab, ob (1) erfüllbar ist oder nicht. Begründen Sie.

Es gibt nach Terminierung des Algorithmus keinen Zyklus im Graphen, der genau eine gestrichelte Kante enthält. Daher enthält (1) keinen Widerspruch und ist erfüllbar.
Ergänzende Erklärung: Eine mögliche erfüllende Belegung wäre mit der Domäne $D = \{1, 2\}$ z.B. $I(a) = 1, I(f)(x) = 1, I(g)(1) = 1$ und davon abgegrenzt $I(b) = 2, I(g)(2) = 2$. Damit sind die Werte der Knoten genau dann identisch, wenn die Knoten in einer Äquivalenzklasse liegen.

4 Formalisieren in Prädikatenlogik (PL1) (1 + 2 + 2 + 3 = 8 Punkte)

Wir betrachten ein Universum aus verschiedenen Sportmannschaften, die gegeneinander Spiele ausgetragen haben. Jede Mannschaft hat höchstens einmal gegen jede andere Mannschaft gespielt. Jedes Spiel endete mit einem Sieg für eine der beiden Mannschaften oder mit einem Unentschieden.

Gegeben sei dazu folgende prädikatenlogische Signatur:

$$\Sigma = (\{a, b\}, \{Play, Win\}, \alpha)$$

Sie enthält

- die Konstantensymbole a und b,
- die zweistelligen Prädikatensymbole $Play$ und Win.

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

das Universum D	ist die Menge aller Mannschaften
$I(a)$	die Mannschaft a
$I(b)$	die Mannschaft b
$(x, y) \in I(Play)$	gdw. x gegen y gespielt hat
$(x, y) \in I(Win)$	gdw. x das Spiel gegen y gewonnen hat

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

a. Die Mannschaft a hat mindestens ein Spiel gewonnen und mindestens ein Spiel verloren.

$$\boxed{(\exists x \ Win(a, x)) \land (\exists x \ Win(x, a))}$$

b. Die Mannschaften a und b haben gegeneinander unentschieden gespielt.

$$\boxed{Play(a, b) \land \neg Win(a, b) \land \neg Win(b, a)}$$

c. Keine Mannschaft hat all ihre bisherigen Spiele verloren.

$$\boxed{\neg \exists x \ \forall y \ (Play(x, y) \to Win(y, x))}$$

d. Alle, die gegen eine Mannschaft gespielt haben, die all ihre bisherigen Spiele gewonnen hat, haben mindestens ein Spiel gewonnen.

$$\boxed{\forall x \ \big((\exists y \ Play(x, y) \land (\forall z \ Play(y, z) \to Win(y, z))) \to \exists v \ Win(x, v) \big)}$$

5 Resolution (4 + 4 = 8 Punkte)

a. Normalform für den prädikatenlogischen Resolutionskalkül

Es soll geprüft werden, ob die untenstehende Aussage über den logischen Zusammenhang zwischen den beiden Formeln gilt. Bringen Sie dazu die Aussage in Klauselnormalform, so dass der prädikatenlogische Resolutionskalkül angewendet werden kann.

Notieren Sie die entstehenden Klauseln sowie alle Symbole, die durch Skolemisierung entstehen.

Hinweis: Sie müssen den Resolutionsbeweis **nicht** durchführen!

$$\big(\forall x \forall y\ (p(x,y) \to \exists z\ (q(x,z) \lor q(y,z)))\big) \quad \models \quad \forall x(\exists z\ q(x,f(z)) \lor \neg(p(x,x)))$$

Prämisse: Auflösung der Implikation und Skolemisierung von z
$\forall x \forall y\ (\neg p(x,y) \lor q(x,g(x,y)) \lor q(y,g(x,y)))$
Konklusion: Negation und Umbennenung
$\neg\forall w\ (\exists v\ q(w,f(v)) \lor \neg(p(w,w)))$
Konklusion: Pränexnormalform, Skolemisierung
$\exists w\ \neg(\exists v\ q(w,f(v)) \lor \neg(p(w,w)))$
$\exists w\ (\neg\exists v\ q(w,f(v)) \land (p(w,w)))$
$\exists w\ (\forall v\ \neg q(w,f(v)) \land (p(w,w)))$
$\forall v\ \neg q(c,f(v)) \land (p(c,c))$
Entstehende Klauseln:
(1) $\neg p(x,y),\ q(x,g(x,y)),\ q(y,g(x,y))$
(2) $\neg q(c,f(v))$
(3) $p(c,c)$
Durch Skolemisierung entstandene Symbole: g, c

b. Aussagenlogischer Resolutionskalkül

Beweisen Sie die Unerfüllbarkeit der folgenden aussagenlogischen Formel mittels des Resolutionskalküls.

$$(B\lor\neg E)\ \land\ (A\lor B)\ \land\ \neg E\ \land\ (\neg B\lor\neg C)\ \land\ (\neg B\lor C\lor\neg D)\ \land\ (\neg A\lor B\lor E)\ \land\ (\neg B\lor C\lor D\lor E)$$

Aus der Formel:
(1) $\{B, \neg E\}$
(2) $\{A, B\}$
(3) $\{\neg E\}$
(4) $\{\neg B, \neg C\}$
(5) $\{\neg B, C, \neg D\}$
(6) $\{\neg A, B, E\}$
(7) $\{\neg B, C, D, E\}$
Resolution:
(8) $\{\neg B, C, D\}$ (3, 7)
(9) $\{\neg B, C\}$ (5, 8)
(10) $\{\neg B\}$ (4, 9)
(11) $\{\neg A, B\}$ (1, 6)
(12) $\{B\}$ (2, 11)
(13) $\square$ (10, 12)

6 Spezifikation mit der Java Modeling Language (4 + 5 = 9 Punkte)

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

```
public class A {

    /*@ public normal_behaviour
      @ requires u >= 0;
      @ assignable \nothing;
      @ ensures (\forall int i; 0 <= i && i < \result.length;
      @                  2 <= \result[i] && \result[i] <= u);
      @ ensures (\forall int i; 0 <= i && i < \result.length;
      @                  !(\exists int x; 2 <= x && x <= \result[i];
      @                          \result[i] != x && \result[i] % x == 0));
      @*/
    public static int[] m(int u) { ... }
}
```

Hinweis: „%" ist der Modulo-Operator in Java (bspw. ist `8 % 3 == 2`).

Wenn `m` mit einer ganzen Zahl größer oder gleich 0 aufgerufen wird, terminiert die Methode, verursacht keine Exception und ändert keine existierenden Speicherstellen auf dem Heap. Nach der Ausführung von `m` gilt:

- Alle Elemente im Ergebnisarray sind aus dem Intervall $[2, u]$.
- Alle Elemente im Ergebnisarray sind prim: Für ein Element e gibt es keine Zahl aus dem Intervall $[2, e)$, die e teilt.

Anmerkung: Die Spezifikation beschreibt nicht, dass die Methode *alle* Primzahlen von 2 bis u zurückgibt, sondern nur, dass sie *höchstens* die Primzahlen von 2 bis u zurückgibt. Insbesondere erfüllt die Implementierung, die immer ein leeres Array zurückgibt, die Spezifikation. Außerdem wird auch keine Aussage über die Reihenfolge der Zahlen im Ergebnisarray getroffen.

Fortsetzung 6 Spezifikation mit der Java Modeling Language

b. Vervollständigen Sie den folgenden Methodenvertrag für die Methode `kgv`, die für ein gegebenes, nichtleeres Array von ganzen Zahlen größer als 0 das **kleinste gemeinsame Vielfache** aller Arrayelemente berechnet. Die Methode terminiert immer, wirft keine Exception und ändert keine bestehenden Speicherstellen auf dem Heap.

Hinweis: Für $n > 0$ ist das **kleinste gemeinsame Vielfache** der Zahlen $a_1, \ldots, a_n$, die jeweils größer als 0 sind, definiert als die kleinste Zahl, die ein Vielfaches aller a_i (für $0 \leq i \leq n$) ist.

```
public class A {

    /*@ public normal_behavior
      @ requires a.length > 0;
      @ requires (\forall int i; 0 <= i && i < a.length; a[i] > 0);
      @ assignable \nothing;
      @ ensures (\forall int i; 0 <= i && i < a.length; \result % a[i] == 0);
      @ ensures !(\exists int x; 0 <= x && x < \result;
      @             (\forall int j; 0 <= j && j < a.length; x % a[j] == 0));
      @*/
    public static int kgv(int[] a) { ... }
}
```

7 Lineare Temporale Logik (LTL) und Büchi-Automaten (4 + 3 = 7 Punkte)

a. Die LTL-Formel $\underbrace{(a\,\mathbf{U}\,b)\,\mathbf{U}\,c}_{X}$ ist nicht äquivalent zu $\underbrace{a\,\mathbf{U}\,(b\,\mathbf{U}\,c)}_{Y}$ über der Signatur $\Sigma = \{a, b, c\}$.

Ergänzen Sie die Variablenbelegung für die Zeitpunkte (Rechtecke) in den untenstehenden ω-Strukturen β_1 und β_2,so dass ...

i. $\beta_1 \models X$ und $\beta_1 \models Y$
ii. $\beta_2 \not\models X$ und $\beta_2 \models Y$

Hinweis: Nicht spezifizierte Belegungen gelten als beliebig ("don't-care"). Sie können **bei Bedarf weitere Zeitpunkte** in den ω-Strukturen ergänzen.

β_1: → ...

$a =$	-
$b =$	-
$c =$	w

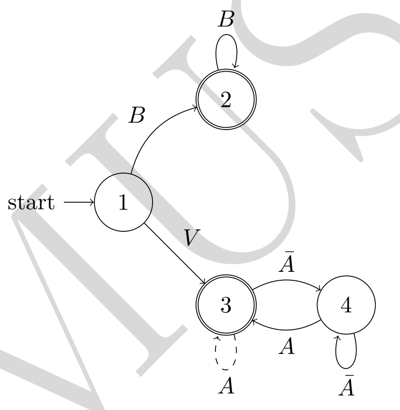
β_2: → → ...

$a =$	w	$a =$	f
$b =$	f	$b =$	f
$c =$	f	$c =$	w

b. Geben Sie für den untenstehenden nicht-deterministischen Büchi-Automaten eine äquivalente LTL-Formel über $\Sigma = \{a, b\}$ an.

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

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



LTL-Formel:

$\boxed{(\Diamond\Box\neg a) \to \Box b}$

Korrektur: Die gestrichelte Kante wurde in der ursprünglichen Klausur vergessen.

*Name: **Vorname** **Familienname***
*Matr.-Nr.: **Matr.-Nr.***

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

Zum Abreißen und Mitnehmen — Klausur Formale Systeme SS 2021

Vorname:	**Vorname**	**Name:**	**Familienname**
Matrikel-Nr.:	**Matr.-Nr.**	**Hörsaal:**	**Hörsaal** **Sitzplatz**
Code:	**Nonce**		