Karlsruher Institut für Technologie

Name: ____________________

Vorname: ____________________

Matrikel-Nr.: ____________________

Klausur Formale Systeme

Fakultät für Informatik

WS 2020/21

Prof. Dr. Bernhard Beckert

06. April 2021

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (13)	A2 (9)	A3 (8)	A4 (8)	A5 (6)	A6 (8)	A7 (8)	Σ (60)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

1 Zur Einstimmung ((2+2) + 5 + 4 = 13 Punkte)

a. Seien p, q einstellige Prädikatensymbole, r ein zweistelliges Prädikatensymbol 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. $\forall x \big(((\exists y\, p(y)) \land q(x)) \to \forall z\, (p(z) \land q(z))\big)$

Interpretation, in der die Formel wahr ist:

$I(p)(a) =$	$I(p)(b) =$	$I(q)(a) =$	$I(q)(b) =$

Interpretation, in der die Formel falsch ist:

$I(p)(a) =$	$I(p)(b) =$	$I(q)(a) =$	$I(q)(b) =$

ii. $(\exists x \forall y\, r(f(x), y)) \to (\exists x \forall y\, r(x, f(y)))$

Interpretation, in der die Formel wahr ist:

$I(r)(a, a) =$	$I(r)(a, b) =$	$I(r)(b, a) =$	$I(r)(b, b) =$

$I(f)(a) =$	$I(f)(b) =$

Interpretation, in der die Formel falsch ist:

$I(r)(a, a) =$	$I(r)(a, b) =$	$I(r)(b, a) =$	$I(r)(b, b) =$

$I(f)(a) =$	$I(f)(b) =$

Fortsetzung 1 Zur Einstimmung

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

i. Geben Sie einen allgemeinsten Unifikator für folgende zwei Terme an:

- $g(x, f(a, h(y), c))$
- $g(z, f(a, h(c), z))$

(x, y, z sind Variablen, a, b, c Konstanten und f, g, h Funktionssymbole).

ii. Wann heißt ein Reduktionssystem $(D, \succ)$ noethersch?

iii. Was beschreibt eine `assignable`-Klausel in JML-Methodenverträgen?

iv. Nennen Sie eine Konsequenz des Gödelschen Unvollständigkeitssatzes.

v. Geben Sie eine zu $(A \wedge B) \to C$ äquivalente Formel an, die als logische Operatoren nur den Shannon-Operator $sh(\cdot, \cdot, \cdot)$ und die logischen Konstanten 0 und 1 enthält.

c. Man könnte meinen, dass folgende Variante der aussagenlogischen Resolutionsregel gut ist, weil sie zwei Resolutionsschritte zu einem zusammenfasst.

Zeigen Sie, dass diese Regel jedoch **nicht** korrekt ist.

Doppel-Resolution:

$$\frac{C_1 \cup \{P, Q\} \quad C_2 \cup \{\neg P, \neg Q\}}{C_1 \cup C_2}$$

für beliebige Klauseln C_1, C_2 und Atome P, Q

2 Modallogik (5 + 2 + 2 = 9 Punkte)

Für diese Aufgabe betrachten wir die Gebot-Modallogik. In dieser Logik gibt es einen Box-Operator $\Box_a$ für Gebote (engl. *command*):

$\Box_a \phi$ bedeutet: Agent a muss dafür sorgen,
dass ϕ im nächsten Zustand wahr ist.

a. Die Aussage

„Wenn es brennt, muss a den Alarm auslösen."

könnte man auf die folgenden zwei verschiedenen Arten formalisieren:

(1) $\mathit{Brennt} \to \Box_a \mathit{Alarm}$

(2) $\Box_a(\mathit{Brennt} \to \mathit{Alarm})$

i. Beschreiben Sie in natürlicher Sprache den Unterschied in der Bedeutung von (1) und (2).

ii. Welche der beiden Formalisierungen ist besser? Begründen Sie ihre Antwort!

b. Formalisieren Sie folgende Aussage in der Gebot-Modallogik:

„Agent a muss dafür sorgen, dass Agent b verhindert, dass es Alarm gibt."

c. Beschreiben Sie in natürlicher Sprache die Bedeutung folgender Formel in der Gebot-Modallogik:

$\Diamond_a \phi$

3 Entscheidungsverfahren für uninterpretierte Funktionssymbole (3 + 3 + 2 = 8 Punkte)

Gegeben ist die folgende prädikatenlogische Formel:

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

Darin sind a, b, c Konstantensymbole und f ist ein einstelliges Funktionssymbol.

a. Welche Eigenschaften muss eine Formel der Prädikatenlogik (PL1) besitzen, damit ihre Erfüllbarkeit mit dem Algorithmus nach *Shostak* geprüft werden kann?

- ______________________________
- ______________________________
- ______________________________

b. Wenden Sie nun den Algorithmus nach *Shostak* auf die Formel (1) an:

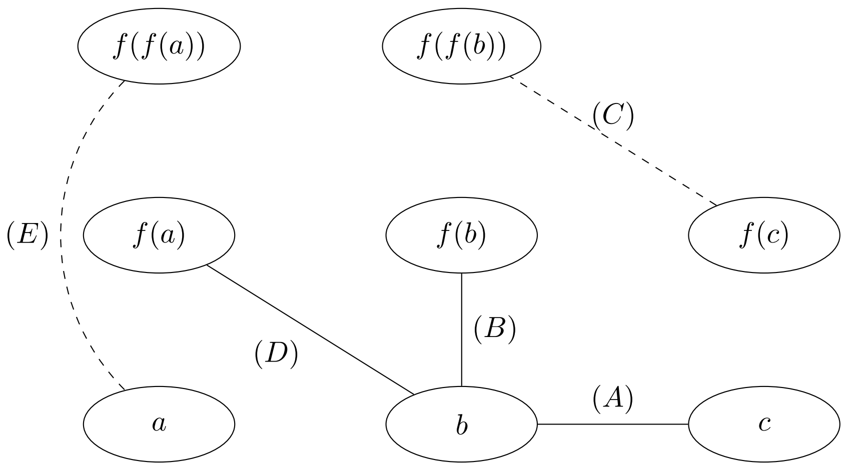
Ergänzen Sie dazu den unten stehenden Kongruenzgraphen (wie in der Vorlesung/Übung) um die Kanten, die durch die Kongruenzeigenschaft hinzugenommen werden müssen.

Durchgezogene Kanten stehen für Gleichheiten und gestrichelte Kanten für Ungleichheiten.

Ergänzen Sie dazu den Graphen um Kanten mit Namen ((F) und folgende). Tragen Sie für jede hinzugefügte Kante im Graphen in die Tabelle ein, welche bisherigen Kante(n) die neue begründet.

Kanten, die sich transitiv aus anderen durchgezogenen Kanten ergeben, müssen Sie nicht in die Tabelle aufnehmen.

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



Name	*Begründung*
(F)	
(G)	
(H)	
(I)	
(K)	
(L)	
(M)	

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

____

____

4 Formalisieren in Prädikatenlogik (PL1) (5 + 3 = 8 Punkte)

Wir wollen das Konzept eines Stacks von natürlichen Zahlen modellieren. Ein Stack ist eine Struktur, auf der Objekte abgelegt werden können. Wenn der Stack nicht leer ist, können wir ein Objekt von dem Stack nehmen (das letzte, das wir abgelegt haben) und erhalten den Stack ohne sein erstes Element.

Gegeben sei folgende prädikatenlogische Signatur:

$$\Sigma = (\{empty\_stack, push, pop, peek\}, \{stack, nat, is\_empty\}, \alpha)$$

Sie enthält

- das Konstantensymbol *empty*,
- die Funktionssymbole $push(\cdot,\cdot)$, $pop(\cdot)$, $peek(\cdot)$,
- die Prädikatensymbole $isStack(\cdot)$, $isNat(\cdot)$, $isEmpty(\cdot)$.

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

das Universum D ist die Menge aller Stacks und aller natürlichen Zahlen	
$I(empty)$	der leere Stack
$I(push(s, n))$	Stack, der entsteht, wenn man n oben auf s legt (falls s Stack, n Zahl)
$I(pop(s))$	Stack, der entsteht, wenn man das oberste Element von s nimmt (falls s nicht-leerer Stack)
$I(peek(s))$	das oberste Element von s (falls s nicht-leerer Stack)
$I(isStack(s)) = W$	gdw. s ist ein Stack
$I(isNat(n)) = W$	gdw. n ist eine Zahl
$I(isEmpty(s)) = W$	gdw. s ist der leere Stack

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

i. Ein leerer Stack ist ein Stack.

ii. Wenn man eine natürliche Zahl auf einen Stack legt, erhält man einen Stack.

iii. Nimmt man das oberste Element von einem nicht-leeren Stack, hat man wieder einen Stack.

iv. Für jeden nicht-leeren Stack gibt es eine natürliche Zahl, die oben auf dem Stack liegt.

v. Die natürliche Zahl, die man mit *peek* erhält, ist die letzte, die auf dem Stack abgelegt wurde.

Fortsetzung 4 Formalisieren in Prädikatenlogik (PL1)

b. Nehmen wir nun an, dass zusätzliche das Prädikat

$$concat(\cdot,\cdot,\cdot)$$

in Σ enthalten sei. Es hat die Bedeutung:

$I(concat(s_1, s_2, s)) = W$	gdw. Stack s ist die Konkatenation der Stacks s_1, s_2, d.h., s entsteht, indem alle Elemente von s_1 oben auf s_2 gelegt werden unter Beibehaltung der Reihenfolge.

Geben Sie eine Axiomatisierung von *concat* an.

Hinweis: Geben Sie zwei Axiome an, eines für den Basisfall und eines für den Rekursionsfall.

5 Tableaukalkül (6 Punkte)

Es sei eine prädikatenlogische (PL1) Signatur gegeben, die das zweistellige Prädikatensymbol p sowie die Konstantensymbole a, b, c enthält.

Zeigen Sie mithilfe des Tableaukalküls der Prädikatenlogik, dass folgende Formel **allgemeingültig** ist.

Notieren Sie dabei bei jeder Erweiterung, durch welche Regelanwendung eine Formel auf dem Tableau enstanden ist. Notieren Sie außerdem jeweils, welche Formeln für einen Abschluss verwendet wurden. Geben Sie die schließende Substitution an.

$$((\forall x\ p(x,a) \vee p(x,b)) \wedge \exists y\ \neg p(y,a)) \rightarrow \exists z\ p(z,b)$$

6 Spezifikation mit der Java Modeling Language (3 + 5 = 8 Punkte)

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

```
public class A {
    public int[] a;

    /*@ public normal_behaviour
      @ assignable \nothing;
      @ ensures (0 <= \result && \result < a.length)
      @     <==> (\exists int j; 0 <= j && j < a.length; a[j] == val);
      @ ensures \result == -1
      @     <==> (\forall int j; 0 <= j && j < a.length; a[j] != val);
      @ ensures (0 <= \result && \result < a.length) ==> (a[\result] == val);
      @ ensures -1 <= \result && \result < a.length;
      @*/
    public int m(int val) { ... }
}
```

Fortsetzung 6 Spezifikation mit der Java Modeling Language

b. Sei eine Methode `f` der Klasse `A` durch die untenstehende Implementierung gegeben. Der untenstehende Vertrag spezifiziert, dass die Methode `f` eines der Elemente aus dem Array `a` zurückgibt (im Fall der untenstehenden Implementierung ist dies das maximale Element). Als Vorbedingung vorausgesetzt ist, dass `a` nicht leer ist und ausschließlich nicht-negative Werte enthält.

Die untenstehende Schleifeninvariante ist unvollständig. Ergänzen Sie diese, sodass daraus eine korrekte **Invariante** für die Schleife entsteht, mit der die Nachbedingung des Methodenvertrags bewiesen werden kann. Geben Sie hierbei auch eine korrekte Schleifen**variante** (`decreases`-Klausel) an, mithilfe derer die Terminierung gezeigt werden kann.

```
public class A {
    public int[] a;

   /*@ requires 0 < a.length
     @   && (\forall int i; 0 <= i && i < a.length; 0 <= a[i]);
     @ assignable \nothing;
     @ ensures (\exists int i; 0 <= i && i < a.length; \result == a[i]);
     @*/
   public int f() {
     int max = 0; int k = 0;

     /*@  loop_invariant ________________________________________

       @  ______________________________________________________

       @  ______________________________________________________

       @  ______________________________________________________

       @  ______________________________________________________

       @  assignable \nothing;

       @  decreases ____________________________________________

       @*/
     while (k < a.length) {
         if (max < a[k]) max = a[k];
         k++;
     }
     return max;
   }
}
```

7 Lineare Temporale Logik (LTL) und Büchi-Automaten ((2+2) + 4 = 8 Punkte)

a. Formalisieren Sie folgenden Sachverhalt einer Verkehrsampel mit LTL. Verwenden Sie dabei folgende Signatur:

G_X, Y_X, R_X – Ampel X ist grün, gelb oder rot.

i. Ampel A und Ampel B sind niemals gleichzeitig oder direkt nacheinander grün.

---

ii. Wenn Ampel A immer wieder (unendlich oft) grün zeigt, dann gilt das auch für Ampel B.

---

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

$$(\Diamond a) \to (\Diamond b)$$

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

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

$A = \{M \in V \mid a \in M\} \subset V$ $\quad$ $B = \{M \in V \mid b \in M\} \subset V$

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

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

Zum Abreißen und Mitnehmen — Klausur Formale Systeme WS 2020/2021

Vorname: ______________ **Name:** ______________

Matrikel-Nr.: ______________ **Code:** ______________