Karlsruher Institut für Technologie

Name: ______________________

Vorname: ______________________

Matrikel-Nr.: ______________________

Klausurnummer:

____

Klausur Formale Systeme

Fakultät für Informatik

SoSe 2025

Prof. Dr. Bernhard Beckert

02. September 2025

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (9)	A2 (7)	A3 (7,5)	A4 (7)	A5 (9)	A6 (10)	A7 (10,5)	Σ (60)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

Matr.-Nr.:

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

a. Was bedeutet es, dass der Tableaukalkül für Prädikatenlogik *vollständig* ist? Was bedeutet es, dass es *korrekt* ist?

Korrekt: ______________________________

____

Vollständig: ______________________________

____

b. Auf der linken Seite sehen Sie die (impl-right) Regel des Sequenzenkalküls. Vervollständigen Sie rechts die (impl-left) Regel:

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

c. Aus der Vorlesung wissen Sie, dass modallogische Formeln in *Kripkestrukturen* $\mathcal{K} = (S, R, I)$ ausgewertet werden. Hierbei ist S die Menge der Zustände, $R \subseteq S^2$ die Zugänglichkeitsrelation und I die Interpretation der aussagenlogischen Variablen. (S, R) wird auch *Kripkerahmen* genannt. Eine Formel F heißt allgemeingültig in einem Kripkerahmen (S, R), wenn für alle möglichen Interpretationen I gilt, dass F in (S, R, I) in allen Zuständen wahr ist.

i. Begründen Sie:
Wenn in einem Krikperahmen (S, R) die Formel $F \equiv P \to \Box\Diamond P$ allgemeingültig ist, dann ist die Relation R symmetrisch.

____

____

____

____

ii. Begründen Sie:
Wenn die Relation R reflexiv ist, dann ist die Formel $F \equiv \Box P \to P$ im Kripkerahmen (S, R) allgemeingültig.

____

____

____

____

Matr.-Nr.:

Fortsetzung 1 Zur Einstimmung

d. In der Vorlesung haben Sie das *Entmischen* von Theorien (im Kontext von SMT) kennengelernt. Die folgende Formel F mischt die prädikatenlogische Theorie für uninterpretierte Funktionen mit der Theorie für lineare Arithmetik. Entmischen Sie diese beiden Theorien, indem Sie eine erfüllbarkeitsäquivalente Formel angeben, die keine gemischten atomaren Formeln beinhaltet.

$$F \quad \equiv \quad f(a + g(c)) \doteq g(a + c) \quad \wedge \quad g(a) + 1 > 0$$

Matr.-Nr.:

2 Formalisierung von Endlichkeit (2 + 2 + 3 = 7 Punkte)

Folgender Satz sei als bekannt vorausgesetzt:

> **Satz.** Eine nichtleere Menge M ist genau dann endlich, wenn jede surjektive Selbstabbildung $f : M \to M$ auch injektiv ist.

a. Formalisieren Sie die Eigenschaften *Surjektivität* und *Injektivität* der Funktion $f : M \to M$ in der Prädikatenlogik erster Stufe (PL1). Gehen Sie davon aus, dass M das Universum der Interpretation ist.

Geben Sie dazu jeweils eine Formel φ_{inj} und φ_{surj} an, die ausdrücken, dass f injektiv bzw. surjektiv ist.

$\varphi_{\text{inj}} \equiv$ ______________________________

____

____

$\varphi_{\text{surj}} \equiv$ ______________________________

____

____

b. Obwohl man Injektivität und Surjektivität in PL1 formalisieren kann, lässt sich die *Eigenschaft der Endlichkeit* des Universums der Interpration nicht in PL1 ausdrücken.

Erläutern Sie, weshalb die Formeln aus Teilaufgabe (a) zusammen mit dem obigen Satz über Endlichkeit *nicht* ausreichen, um Endlichkeit in PL1 zu formalisieren.

____

____

____

____

____

c. In der Prädikatenlogik zweiter Stufe (Second-Order Logic), in der auch über Funktionen und Relationen quantifiziert werden kann, lässt sich Endlichkeit ausdrücken. Geben Sie eine Formel in Prädikatenlogik zweiter Stufe an, die ausdrückt, dass das Universum der Interpretation endlich ist, basierend auf der obigen Charakterisierung mittels Surjektivität und Injektivität.

____

____

____

____

Matr.-Nr.:

3 Entscheidungsverfahren für uninterpretierte Funktionssymbole (5 + 1,5 + 1 = 7,5 Punkte)

Gegeben ist die folgende prädikatenlogische Formel:

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

Darin sind a, b Konstantensymbole, und $f(\cdot), g(\cdot)$ sind einstellige Funktionssymbole.

a. Zeigen Sie mithilfe des Algorithmus von *Shostak*, dass die Formel F erfüllbar ist.

Geben Sie dazu den vollständigen Kongruenzgraphen nach Ausführung des *Shostak*-Algorithmus auf die Formel F an.

Durchgezogene Kanten stehen dabei für Gleichheiten und gestrichelte Kanten für Ungleichheiten. Geben Sie für jede Kante an, welche (Un-)Gleichung aus Formel F beziehungsweise, welche bereits existierende(n) Kante(n) die neue Kante begründet.

b. Geben Sie ein Modell (D, I) der Formel F an.

$D =$ ☐

$I(a) =$ ☐ $\qquad I(b) =$ ☐

$I(f)(x) =$ ☐ $\qquad I(g)(x) =$ ☐

Matr.-Nr.:

Fortsetzung 3 Entscheidungsverfahren für uninterpretierte Funktionssymbole

c. Geben Sie ein Universum an, sodass die Formel F für alle Interpretationsfunktionen zu falsch auswertet.

Matr.-Nr.:

4 Formalisieren in Prädikatenlogik (PL1)

(1 + 1 + 2 + 3 = 7 Punkte)

Im Folgenden sollen mit Prädikatenlogik die Verwandtschaftsbeziehungen von Lebewesen formalisiert werden. Hierzu sei die prädikatenlogische Signatur

$$\Sigma = (\{sp(\cdot)\}, \{isSp(\cdot), isOr(\cdot), dirDesc(\cdot,\cdot), desc(\cdot,\cdot)\}, \alpha)$$

gegeben. Sie enthält das einstellige Funktionssymbol $sp(\cdot)$, die einstelligen Prädikatensymbole $isSp(\cdot)$, $isOr(\cdot)$ und die zweistelligen Prädikatensymbole $dirDesc(\cdot,\cdot)$, $desc(\cdot,\cdot)$.

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

- das Universum D eine Menge von Lebewesen und Spezies ist,
- $isSp(x)$ genau dann wahr ist, wenn x eine Spezies ist,
- $isOr(x)$ genau dann wahr ist, wenn x ein Lebewesen ("organism") ist,
- $dirDesc(x, y)$ genau dann wahr ist, wenn x, y Lebewesen sind und y ein direkter Nachfahre von x ist,
- $desc(x, y)$ genau dann wahr ist, wenn x, y Lebewesen sind und y ein Nachfahre von x ist,
- $sp(x)$ die Spezies ist, zu der das Lebewesen x gehört (falls x kein Lebewesen ist, ist der Wert von $sp(x)$ nicht näher bestimmt),

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

a. Jeder direkte Nachfahre ist ein Nachfahre.

b. Es gibt ein Lebewesen, das ein gemeinsamer Vorfahre aller Lebewesen (einschließlich sich selbst) ist.

c. Zwei Lebewesen mit wenigstens einem gemeinsamen direkten Nachfahren gehören zur selben Spezies.

d. Wenn zwei Lebewesen dieselbe nichtleere Menge von Nachfahren haben, haben sie mindestens einen gemeinsamen direkten Nachfahren.
Hinweis: Sie dürfen in dieser Aufgabe den Äquivalenzoperator $\leftrightarrow$ verwenden, wobei die Formel $a \leftrightarrow b$ äquivalent zu $a \to b \land b \to a$ ist.

Matr.-Nr.:

5 Tableau (2 + 7 = 9 Punkte)

a. Betrachten Sie das folgende Tableau:

$$
\begin{array}{c}
1\ \forall x\, p(x) \land \neg p(f(x))\ {}_{1} \\
| \\
1\ p(X) \land \neg p(f(X))\ {}_{2\ \gamma(1)} \\
| \\
1\ p(X)\ {}_{3\ \alpha(2)} \\
| \\
1\ \neg p(f(X))\ {}_{4\ \alpha(2)} \\
| \\
0\ p(f(X))\ {}_{5\ \alpha(4)}
\end{array}
$$

Existiert eine schließende Substitution? Falls ja, geben Sie diese an. Falls nicht, begründen Sie, ob sich das Tableau dennoch zu einem geschlossenen Tableau erweitern lässt.
Hinweis: Sie müssen in diesem Fall das Tableau nicht vervollständigen, eine Erläuterung genügt.

Fortsetzung 5 Tableau

b. Seien p und q einstellige Prädikatensymbole.

Vervollständigen und schließen Sie den folgenden Tableau-Beweis.

Notieren Sie dabei:

- den Regeltyp $(\alpha, \beta, \gamma, \delta)$ und die Formel, auf die eine Regel angewendet wird,
- bei Abschlüssen die beiden Partner,
- sowie die schließende Substitution.

$$1\ \forall y\ (q(y) \lor \forall x\ (p(y) \to q(f(x))))\ {}_1$$
$$|$$
$$1\ \forall x\ (p(f(c)) \land \neg q(f(x)))\ {}_2$$

Matr.-Nr.:

6 Spezifikation mit der Java Modeling Language (JML) (5 + 5 = 10 Punkte)

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

```
public final class A {
    /*@ normal_behavior
      @  requires a.length >= 2;
      @  requires a[0] == ' ';
      @  requires (\forall int i; 0 <= i < a.length;
      @                   'a' <= a[i] <= 'z' || a[i] == ' ');
      @
      @  ensures (\forall int i; 1 <= i < a.length;
      @                   a[i] == (\old(a[i]) != ' ' && \old(a[i-1]) == ' ')
      @                         ? (\old(a[i]) - 32)
      @                         :  \old(a[i]));
      @  assignable a[*];
      @*/
    static void m(char[] a) { ... }
}
```

Hinweis: Einzelne Zeichen (`char`) in Java/JML verhalten sich wie Ganzzahlen, wobei Kleinbuchstaben und Großbuchstaben jeweils zusammenhängende Bereiche sind. Das Leerzeichen wird durch `' '` dargestellt. Wenn man von einem `char`, der einen Kleinbuchstaben repräsentiert, 32 abzieht, erhält man den entsprechenden Großbuchstaben.

Fortsetzung 6 Spezifikation mit der Java Modeling Language

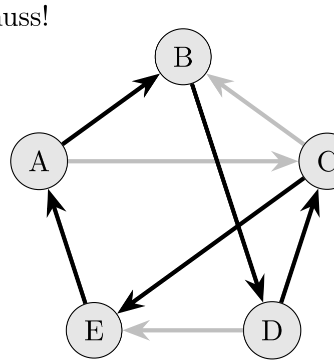
b. In der Graphentheorie ist ein Hamiltonkreis ein zyklischer Pfad in einem Graphen, der jeden Knoten genau einmal enthält.

Gegeben sei die Klasse `Graph`, die einen gerichteten Graphen implementiert. Dabei speichert jeder Graph seine Knoten in einem Array. Jeder Knoten wiederum speichert ein Array seiner Nachfolgerknoten (ausgehende Kanten).

Gehen Sie davon aus, dass der Graph (mindestens) einen Hamiltonkreis hat (dargestellt durch das hier nicht weiter definierte Prädikat in der Vorbedingung). Die Methode `hamilton` soll einen Hamiltonkreis als **Array von Indizes von `kn`** zurückgeben, wobei jeder Index genau einmal enthalten ist. Außerdem muss sichergestellt sein, dass für im Ergebnis direkt aufeinanderfolgende Indizes deren Knoten auch wirklich durch eine Kante verbunden sind.

Vervollständigen Sie die Nachbedingung der Methode `hamilton` entsprechend.

Hinweis: Beachten Sie, dass der Pfad geschlossen sein muss!



Visualisierung eines Hamiltonkreises
(schwarze Kanten)

```
class Knoten {
  Knoten[] nachfolger;

  ...
}


class Graph {
  Knoten[] kn;

  /*@ normal_behavior
    @  requires hatHamiltonKreis(this);
    @  ensures ______________________________________
    @ ________________________________________________
    @ ________________________________________________
    @ ________________________________________________
    @ ________________________________________________
    @ ________________________________________________
    @ ________________________________________________
    @ ________________________________________________
    @ ________________________________________________
    @  assignable \nothing;
    @*/
  int[] hamilton() { ... }

  // gibt genau dann true zurück, wenn g (mindestens) einen Hamiltonkreis hat
  static boolean hatHamiltonKreis(Graph g) { ... }

  ...
}
```

 Matr.-Nr.:

7 Lineare Temporale Logik (LTL)
(2 * 1 + 2 + 2,5 + 4 = 10,5 Punkte)

In einem Volleyballspiel treten zwei Mannschaften A und B gegeneinander an. Das Spiel ist untergliedert in Sätze. Um einen Satz zu gewinnen, muss eine Mannschaft mindestens 25 Punkte erzielen und zusätzlich mindestens zwei Punkte Vorsprung haben. Um das Spiel zu gewinnen, muss eine Mannschaft drei Sätze für sich entscheiden (Best of Five). Der fünfte Satz ist dabei der Entscheidungssatz.

Gegeben ist die Signatur

$$\Sigma = \{winA_i, winB_i \mid i \in \mathbb{N}, 0 \leq i \leq 5\} \cup \{pointA, pointB\} \cup \{tiebreak\}$$

Es gilt:

- $winA_i$ bzw. $winB_i$ für $0 \leq i \leq 5$ ist genau dann wahr, wenn Mannschaft A bzw. B bisher *genau* i Sätze gewonnen hat
- $pointA$ bzw. $pointB$ genau dann wahr, wenn Mannschaft A bzw. B einen Punkt erzielt.
- $tiebreak$ ist genau dann wahr, wenn ein Entscheidungssatz gespielt wird

a. Übersetzen Sie die folgenden natürlichsprachlichen Aussagen in LTL-Formeln.

i. Haben beide Mannschaften bisher genau zwei Sätze gewonnen, wird zu einem späteren Zeitpunkt ein Entscheidungssatz gespielt.

ii. Bevor Mannschaft A zwei Sätze gewonnen haben kann, muss sie einen Satz gewonnen haben.

Im folgenden Teil der Aufgabe definieren wir einen neuen LTL-Operator *alternate*, geschrieben **A**. Für zwei LTL-Formeln ϕ, ψ und eine Omega-Struktur ξ gilt $\xi \models \phi \mathbf{A} \psi$ genau dann, wenn folgende Eigenschaften erfüllt sind:

- Sowohl ϕ als auch ψ gelten nie für zwei oder mehr aufeinander folgende Zeitschritte
- ϕ und ψ sind niemals im gleichen Zeitschritt wahr
- Gilt $\xi_i \models \phi$ und $\xi_j \models \phi$ mit $i < j$, so gibt es genau ein $k \in [i, j]$ mit $\xi_k \models \psi$
- Gilt $\xi_i \models \psi$ und $\xi_j \models \psi$ mit $i < j$, so gibt es genau ein $k \in [i, j]$ mit $\xi_k \models \phi$

b. Übersetzen Sie die folgende Formel in natürliche Sprache:

$$(winA_0 \land (pointA \mathbf{A} pointB)) \rightarrow \neg\Diamond winA_1$$

c. Geben Sie eine zu $pointA\ \mathbf{A}\ pointB$ äquivalente Formel F an, welche nur Modaloperatoren aus $\{\Box, \Diamond, \mathbf{X}, \mathbf{U}, \mathbf{U}_w\}$ enthält.

d. Geben Sie einen nicht-deterministischen Büchi-Automaten an, dessen akzeptierte Sprache der Formel $pointA\ \mathbf{A}\ pointB$ entspricht.

Für das Vokabular $V = \mathcal{P}(\Sigma)$ (die Potenzmenge von Σ) werden die folgenden Abkürzungen definiert:

$$P_A = \{X \in V \mid pointA \in X\} \qquad P_B = \{X \in V \mid pointB \in X\}$$
$$\bar{P}_A = \{X \in V \mid pointA \notin X\} \qquad \bar{P}_B = \{X \in V \mid pointB \notin X\}$$

Sie dürfen an den Kanten des Automaten auch Mengen-Ausdrücke wie $P_A \cap P_B$ verwenden.