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:

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?

Korrektheit bedeutet: Wenn eine Formel mit dem Tableaukalkül bewiesen werden kann, dann ist diese auch allgemeingültig.
Vollständigkeit bedeutet: Wenn eine Formel allgemeingültig ist, dann kann diese auch mit dem Tableaukalkül bewiesen werden.

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

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

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.

Angenommen, R sei nicht symmetrisch, dann gibt es $s, s' \in S$, sodass sRs', aber nicht $s'Rs$. Wähle nun I so, dass $P \in I(s)$ und dass P in keinem von s' erreichbaren Zustand wahr ist. Das ist möglich, da nicht $s'Rs$. Dann ist F in s falsch. Dies ist ein Widerspruch zur Annahme.

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

Wir müssen zeigen, dass F in jedem Zustand $s \in S$ gilt; sei s beliebig gewählt. Nehmen wir an, $\Box P$ gilt in s (sonst ist die Implikation in s trivial wahr). Daraus folgt, dass P in allen s' mit sRs' wahr ist. Da aber R reflexiv ist (und somit sRs), gilt dann auch $P \in I(s)$. Somit ist die Implikation in s wahr.

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.

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

$f\left(z\right) \doteq g\left(w\right) \wedge y + 1 > 0 \wedge y \doteq g\left(a\right) \wedge z \doteq a + x \wedge x \doteq g\left(c\right) \wedge w \doteq a + c$

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.

Die gewünschten Formeln in PL1 sind:

$$\varphi_{\mathsf{inj}}(f) := \forall x \, \forall y \, (f(x) \doteq f(y) \to x \doteq y)$$
$$\varphi_{\mathsf{surj}}(f) := \forall y \, \exists x \, (f(x) \doteq y)$$

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.

Obwohl Surjektivität und Injektivität einzelner Funktionen in PL1 formulierbar sind, ist die Aussage

> "jede surjektive Selbstabbildung ist injektiv"

in PL1 nicht ausdrückbar, da es in PL1 nicht möglich ist, *über alle Funktionen* $f : M \to M$ zu quantifizieren.
In PL1 sind Quantifizierungen nur über Elemente des Universums erlaubt, nicht jedoch über Funktionssymbole oder Abbildungen zwischen Elementen. Dies verhindert die vollständige Charakterisierung von Endlichkeit über die gegebene Bedingung.

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.

In der Prädikatenlogik zweiter Stufe kann man über Funktionen quantifizieren. Die gewünschte Formel zur Charakterisierung von Endlichkeit lautet:

$$\forall f \, ([\forall y \, \exists x \, (f(x) \doteq y)] \to [\forall x \, \forall y \, (f(x) \doteq f(y) \to x \doteq y)])$$

Diese Formel drückt aus, dass jede surjektive Funktion $f : M \to M$ auch injektiv ist – also dass M endlich ist.

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

Gegeben ist die folgende prädikatenlogische Formel:

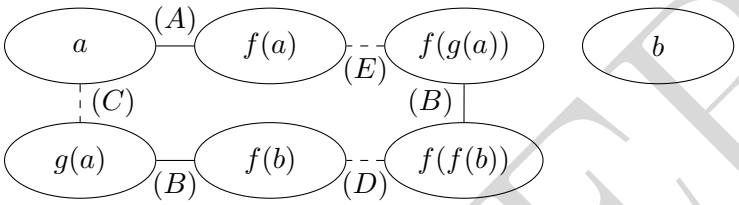
$$F \quad \equiv \quad \underset{(A)}{a \doteq f(a)} \quad \wedge \quad \underset{(B)}{f(b) \doteq g(a)} \quad \wedge \quad \underset{(C)}{\neg g(a) \doteq a} \quad \wedge \quad \underset{(D)}{\neg f(f(b)) \doteq f(b)} \quad \wedge \quad \underset{(E)}{\neg f(g(a)) \doteq f(a)}$$

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.



Bewertungshinweis: Graph sollte vollständig bzgl. der Kanten und Knoten sein (vllt. kann man Knoten b weglassen)

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

$$D = \boxed{\{1, 2, 3\}}$$

$$I(a) = \boxed{1} \qquad I(b) = \boxed{2}$$

$$I(f)(x) = \boxed{\begin{cases} 1, & x = 1 \\ 3, & x = 2 \\ 2, & x = 3 \end{cases}} \qquad I(g)(x) = \boxed{\begin{cases} 3, & x = 1 \\ 3, & x = 2 \\ 3, & x = 3 \end{cases}}$$

Fortsetzung 3 Entscheidungsverfahren für uninterpretierte Funktionssymbole

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

z.B. $D = \{a, b\}$ oder ein beliebiges Universum mit $|D| < 3$. Universen dürfen nicht leer sein. Im Kongruenzgraphen gibt es 3 durch Ungleichungen paarweise getrennte Äquivalenzklassen.

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.

$$\forall a \forall d\ (dirDesc(a, d) \rightarrow desc(a, d))$$

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

$$\exists a\ \forall c\ (isOr(c) \rightarrow desc(a, c))$$

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

$$\forall c_1 \forall c_2\ (\exists d\ (dirDesc(c_1, d) \land dirDesc(c_2, d)) \rightarrow sp(c_1) \doteq sp(c_2))$$

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 $a \rightarrow b \land b \rightarrow a$ ist.

$$\forall c_1 \forall c_2\ ((\forall d\ (desc(c_1, d) \leftrightarrow desc(c_2, d))) \land \exists d\ (desc(c_1, d)) \rightarrow \exists d\ (dirDesc(c_1, d) \land dirDesc(c_2, d)))$$

5 Tableau (2 + 7 = 9 Punkte)

a. Betrachten Sie das folgende Tableau:

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

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.

Da $\sigma(X) = f(\sigma(X))$ gelten müsste, gibt es keine schließende Substitution. Das Tableau lässt sich aber durch erneutes Anwenden von γ auf 1 vervollständigen.

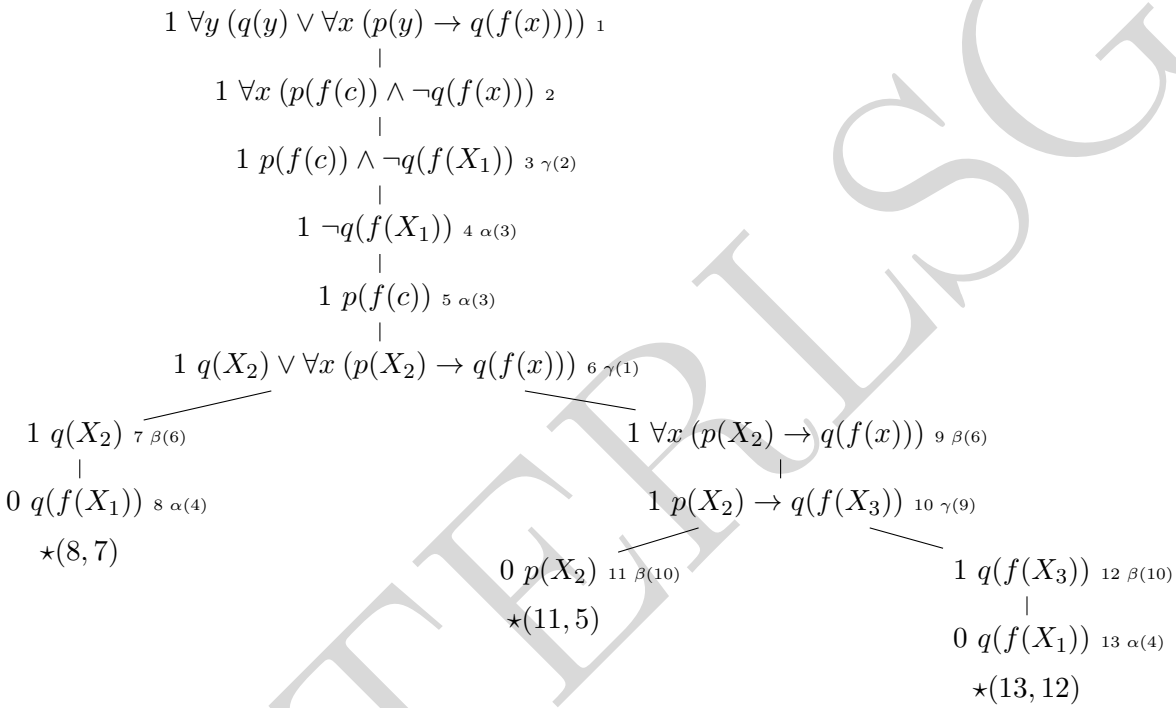
Fortsetzung 5 Tableau

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.



Schließende Substitution: $\sigma = \{X_1/c, X_2/f(c), X_3/c\}$

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.

Wenn die Methode `m` mit einem char-Array aufgerufen wird, dass mindestens Länge 2 hat, dessen erster Eintrag ein Leerzeichen ist und dessen Einträge alle Kleinbuchstaben oder Leerzeichen sind, terminiert die Methode ohne Exception und verändert dabei höchstens die Speicherstellen des Arrays (oder die Speicherstellen neu erzeugter Objekte). Nach Ausführung der Methode gilt dabei: Die Einträge im Array, die vor Aufruf der Methode der erste Buchstabe eines Wortes waren (also ein (Klein-)Buchstabe direkt nach einem Leerzeichen), sind jetzt die dazu passenden Großbuchstaben. Alle anderen Einträge sind unverändert.
Hinweis: Die Vorbedingung, dass der erste Eintrag ein Leerzeichen ist, sorgt dafür, dass man den Quantor in der Nachbedingung bei 1 starten kann und sich eine Fallunterscheidung spart.

Fortsetzung 6 Spezifikation mit der Java Modeling Language

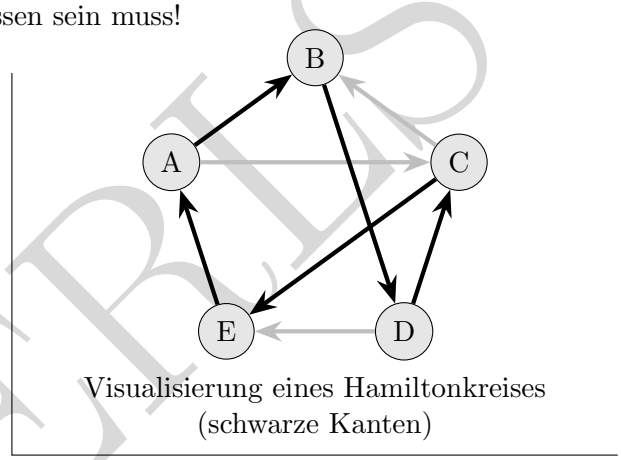
b. In der Graphentheorie ist ein Hamiltonkreis ein zyklischer Pfad in einem Graphen, der jeden Knoten genau einmal enthält.

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);
    @
    @  // Pfad hat genauso viele Knoten wie der Graph:
    @  ensures \result.length == kn.length;
    @
    @  // Jeder Index von kn kommt im Ergebnis vor:
    @  ensures (\forall int i; 0 <= i < kn.length;
    @            (\exists int j; 0 <= j < \result.length; \result[j] == i));
    @
    @  // Knoten von zwei aufeinanderfolgenden Indizes sind mit Kante verbunden:
    @  ensures (\forall int k; 0 <= k < \result.length;
    @             (\exists int l; 0 <= l < kn[\result[k]].nachfolger.length;
    @               kn[\result[k]].nachfolger[l] == kn[\result[(k+1)%\result.length]]));
    @  assignable \nothing;
    @*/
  int[] hamilton() { ... }

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

  ...
}
```

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.

$winA_2 \wedge winB_2 \rightarrow \Diamond tiebreak$

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

$\neg winA_2 \, \mathbf{U}_w \, winA_1$

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:

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

Hat Mannschaft A noch keinen Satz gewonnen und wechseln sich die Punkte der Mannschaften ab, so wird Mannschaft A nie einen Satz gewinnen.

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

$$\Box\left((\mathit{pointA} \rightarrow \neg \mathit{pointB} \wedge \mathbf{X}(\neg \mathit{pointA}\ \mathbf{U}_w\ \mathit{pointB})) \wedge (\mathit{pointB} \rightarrow \mathbf{X}(\neg \mathit{pointB}\ \mathbf{U}_w\ \mathit{pointA}))\right)$$

d. Geben Sie einen nicht-deterministischen Büchi-Automaten an, dessen akzeptierte Sprache der Formel *pointA* **A** *pointB* entspricht.

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

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

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