Karlsruher Institut für Technologie

Name: _____

Vorname: _____

Matrikel-Nr.: _____

Klausur Formale Systeme

Fakultät für Informatik

WS 2021/22

Prof. Dr. Bernhard Beckert

28. Februar 2022

Die Bearbeitungszeit beträgt 60 Minuten.

A1 (13)	A2 (11)	A3 (5)	A4 (8)	A5 (8)	A6 (8)	A7 (7)	Σ (60)

Bewertungstabelle bitte frei lassen!

Gesamtpunkte:

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

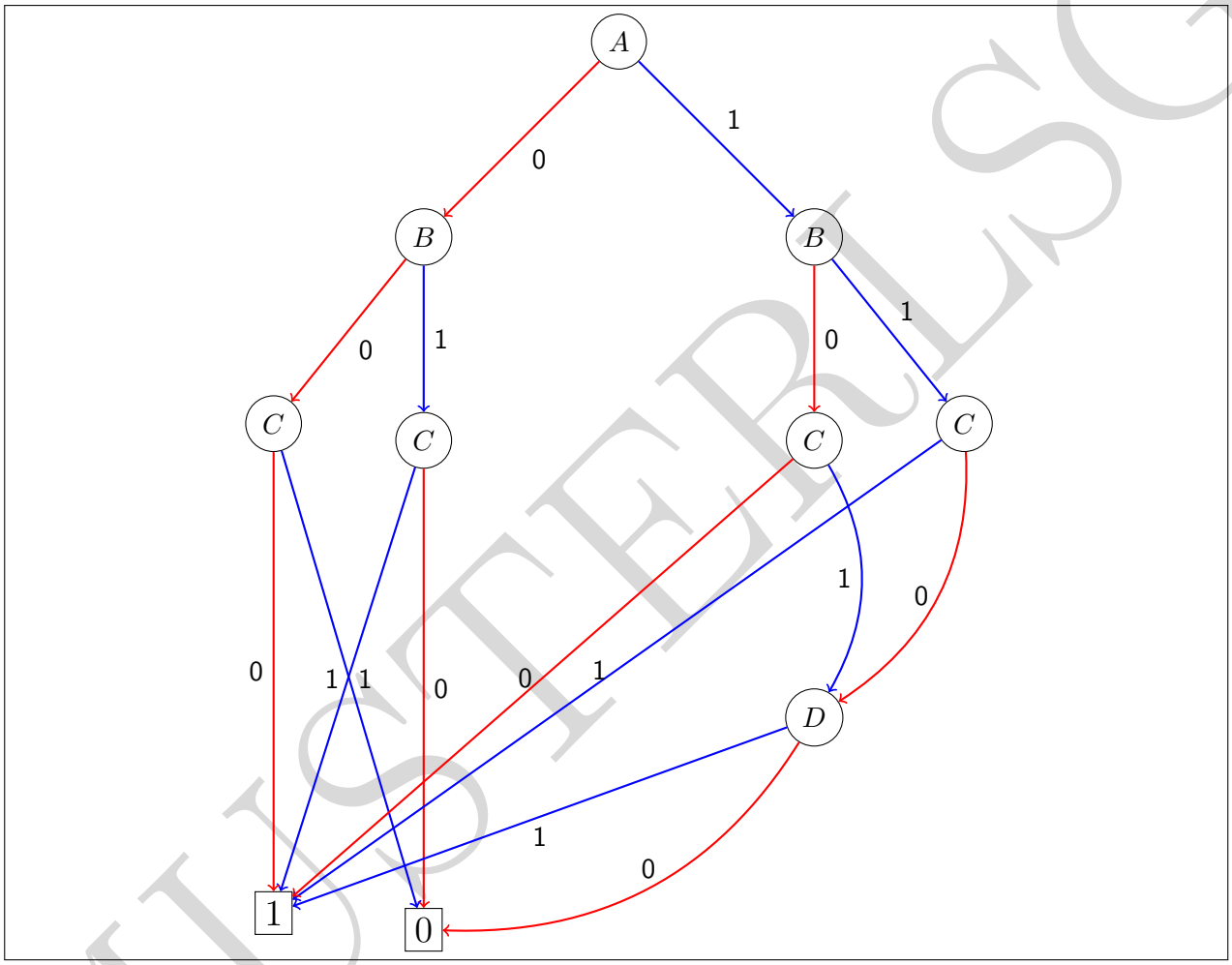
a. Geben Sie den reduzierten Shannongraphen für

$$(A \wedge D) \vee (B \leftrightarrow C)$$

mit der Variablenordnung

$$A < B < C < D$$

an.



Fortsetzung 1 Zur Einstimmung

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

i. Wann heißt eine aussagenlogische Formel A eine Horn-Formel?

A ist in KNF und jede Klausel von A hat höchstens ein positives Literal.

ii. Geben Sie eine **minimale** Teilmenge der logischen Operatoren

$$\{\mathbf{X}, \Diamond, \Box, \mathbf{U}, \neg, \wedge, \vee, \rightarrow, \mathbf{0}, \mathbf{1}\}$$

an, so dass diese Teilmenge eine vollständige Basis der Linearen Temporalen Logik bildet.

$\mathbf{X}, \mathbf{U}, \rightarrow, \mathbf{0}$

iii. Resolvieren Sie folgende zwei Klauseln **in einem Schritt**. Versuchen Sie dabei eine Resolvente mit möglichst wenigen Literalen zu erhalten. Geben Sie die Resolvente und den Unifikator (MGU) an.

$$\{p(x),\ p(g(x)),\ p(g(y))\} \qquad \{\neg p(g(u)),\ \neg p(v)\}$$

Resolvente:

(1) $\{p(g(g(u))\}$ oder (2) $\{p(u)\}$

MGU:

(1) $\{v/g(u), y/u, x/g(u)\}$ oder (2) $\{v/g(u), y/u, x/u\}$

c. Zeigen Sie die Unerfüllbarkeit folgender aussagenlogischer Klauselmenge über $\Sigma = \{A, \ldots, E\}$ mit dem DPLL-Algorithmus:

$$\{\neg A, \neg B, \neg C\},\ \{\neg A, B, \neg C\},\ \{D, E\},\ \{\neg C, A, \neg B\},\ \{A, \neg C, B\},\ \{C\}$$

i. Vereinfachung mit der Unit-Klausel $\{C\}$ ergibt:
$\{\neg A, \neg B\}, \{\neg A, B\}, \{D, E\}, \{A, \neg B\}, \{A, B\}$

ii. Verzweigung über A und $\neg A$.

A:
Vereinfachung mit A ergibt: $\{\neg B\}, \{B\}, \{D, E\}$
Vereinfachung mit B ergibt: $\square, \{D, E\}$

$\neg A$:
Vereinfachung mit $\neg A$ ergibt: $\{D, E\}, \{\neg B\}, \{B\}$
Vereinfachung mit B ergibt: $\{D, E\}, \square$

2 Allgemeingültigkeit aussagenlogischer Formeln (4 + 4 + 3 = 11 Punkte)

Eine aussagenlogische Formel F (über einer endlichen Signatur Σ) ist per Definition entweder allgemeingültig oder nicht. Will man eine Abstufung einführen, so kann man einen „Grad der Allgemeingültigkeit“ $gga(F)$ wie folgt definieren:

$$gga(F) \;=\; \frac{\text{Anzahl der aussagenlogischen Interpretationen } I \text{ über } \Sigma \text{ mit } val_I(F) = W}{\text{Anzahl aller aussagenlogischen Interpretationen über } \Sigma}$$

a. Sei $\Sigma = \{P, Q\}$. Bestimmen Sie $gga(F_i)$ für die folgenden Formeln F_i:

i. $F_1 \;=\; P \vee \neg P$ $\qquad gga(F_1) \;=\;$ 1,0

ii. $F_2 \;=\; P \wedge Q$ $\qquad gga(F_2) \;=\;$ 1/4

iii. $F_3 \;=\; P \to (P \wedge Q)$ $\qquad gga(F_3) \;=\;$ 3/4

iv. $F_4 \;=\; (P \vee Q) \to \neg P$ $\qquad gga(F_4) \;=\;$ 1/2

b. Man kann den Tableaukalkül als ein Hilfsmittel für die Berechnung von $gga(F)$ verwenden, indem man wie folgt vorgeht:

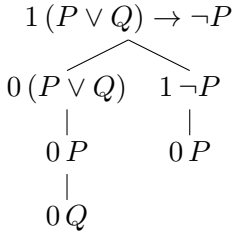
1. ein vollständig expandiertes Tableau aufstellen, das mit dem Knoten $1\,F$ beginnt;
2. aus den offenen Ästen des Tableaus die Interpretationen ablesen, die F wahr machen;
3. diese Interpretationen abzählen und durch die Anzahl aller Interpretationen über Σ teilen.

Hinweis: Die gleiche Interpretation kann in mehreren offenen Ästen enthalten sein; darum genügt es nicht, die Interpretationen getrennt für jeden Ast abzuzählen und dann aufzuaddieren.

Demonstrieren Sie dieses Vorgehen zur Berechnung von $gga(F)$ beispielhaft für die Formel

$$F_4 \;=\; (P \vee Q) \to \neg P$$

1. Schritt: Voll expandiertes Tableau:



2. Schritt: Aus dem linken Ast mit den Atomen $0P, 0Q$ kann man eine Interpretation ablesen: $\bar{P}\bar{Q}$. Aus dem rechten Ast mit dem Atom $0P$ kann man zwei Interpretationen ablesen: $\bar{P}\bar{Q}$ und $\bar{P}Q$.

3. Schritt: Insgesamt gibt es also zwei Interpretationen, die die Formel wahr machen: $\bar{P}\bar{Q}, \bar{P}Q$. Daher ist $gga(F_4) = 2/4 = 1/2$.

Fortsetzung 2 Allgemeingültigkeit aussagenlogischer Formeln

c. Warum ist es für das Vorgehen aus **b.** zur Berechnung von $gga(F)$ vorteilhaft, wenn man statt der üblichen verzweigenden Tableauregeln

$$\frac{1\ F \vee G}{1\ F \mid 1\ G} \qquad \frac{0\ F \wedge G}{0\ F \mid 0\ G} \qquad \frac{1\ F \to G}{0\ F \mid 1\ G}$$

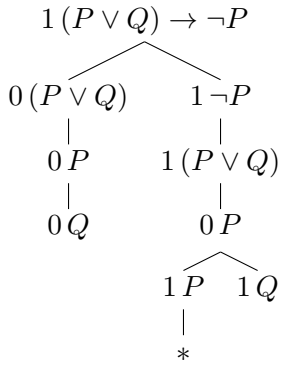
die folgenden modizierten Regeln („Lemma-Regeln“) verwendet?

$$\frac{1\ F \vee G}{1\ F \mid \begin{array}{l}1\ G\\ 0\ F\end{array}} \qquad \frac{0\ F \wedge G}{0\ F \mid \begin{array}{l}0\ G\\ 1\ F\end{array}} \qquad \frac{1\ F \to G}{0\ F \mid \begin{array}{l}1\ G\\ 1\ F\end{array}}$$

Wenn man die Lemma-Regeln verwendet, sind die offenen Äste des voll expandierten Tableaus in dem Sinne disjunkt, dass sie keine Interpretationen gemeinsam haben. Dann genügt es, in Schritt 2 die Anzahl der die Formel wahr machenden Interpretationen jeweils getrennt aus jedem der offenen Äste abzulesen und die Zahlen zusammenzuzählen. Das in dem Hinweis zu Teilaufgabe b. gesagte kann nicht mehr auftreten.

Dies noch einmal an Formel F_4 nachzuvollziehen ist nicht Teil der Aufgabe, sei aber hier zur Erläuterung gemacht:

1. Schritt: Voll expandiertes Tableau:



2. Schritt: Aus dem linken Ast mit den Atomen $0P, 0Q$ kann man eine Interpretation ablesen: $\bar{P}\bar{Q}$. Der mittlere Ast ist geschlossen. Aus dem rechten Ast mit den Atomen $0P, 1Q$ kann man nun ebenfalls nur eine Interpretationen ablesen: $\bar{P}Q$.

3. Daher ist $gga(F_4) = (1+1)/4 = 1/2$.

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

Gegeben sei die folgende prädikatenlogische Formel:

$$\underset{(A)}{a \doteq f(b)} \quad \wedge \quad \underset{(B)}{b \doteq f(a)} \quad \wedge \quad \underset{(C)}{h(b,b) \doteq b} \quad \wedge \quad \underset{(D)}{\neg f(a) \doteq h\big(f(f(b)), f(a)\big)} \tag{1}$$

Darin sind a, b Konstantensymbole, f ein einstelliges und h ein zweistelliges Funktionssymbol.

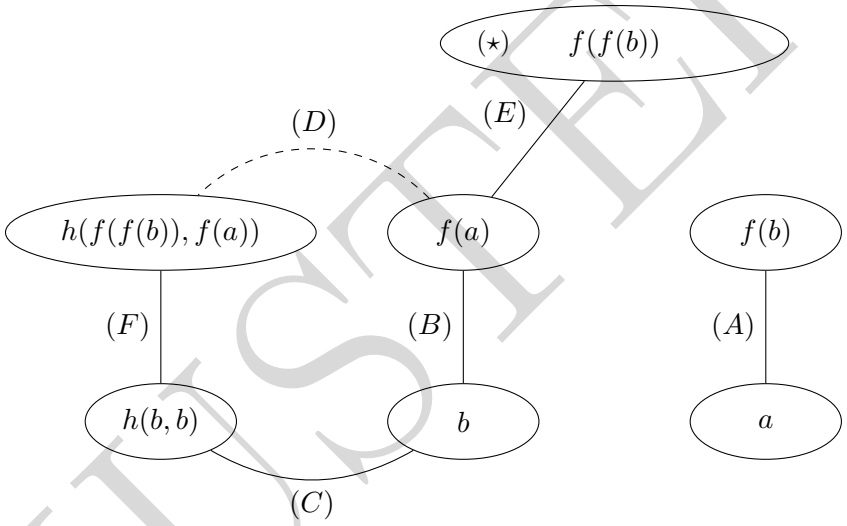
Wenden Sie den Algorithmus nach *Shostak* (wie in der Vorlesung/Übung) auf die Formel (1) an. Ergänzen Sie dazu den unten stehenden Kongruenzgraphen.

a. Ein Term fehlt noch in dem Graphen. Ergänzen Sie ihn im Knoten $(\star)$.

b. Ergänzen Sie nun den Graphen um die Kanten, die durch die Kongruenzeigenschaft hinzugenommen werden müssen. Bezeichnen Sie die neuen Kanten mit $(E), (F)$ usw. Kanten, die sich transitiv aus anderen durchgezogenen Kanten ergeben, nehmen Sie bitte nicht in die Tabelle auf.

Tragen Sie für jede hinzugefügte Kante im Graphen in die Tabelle ein, welche bisherige(n) Kante(n) die neue Kante begründet bzw. begründen.

Hinweis: Es gibt Lösungen, die mit zwei weiteren Kanten auskommen.



Name	*Begründung*
(E)	$\boxed{(A)}$
(F)	$\boxed{(B), (E)}$
(G)	
(H)	

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

Es gibt einen Zyklus im Graphen, der genau eine gestrichelte Kante enthält. Daher enthält (1) einen Widerspruch und ist unerfüllbar.
Ergänzende Erklärung: $f(f(b))$ muss als Unterterm von $h(f(f(b)), f(f(b)))$ hinzugenommen werden, da sich dafür die zusätzliche transitive Kongruenz $b = f(f(b))$ ergibt. Die Kante (F) folgt für das erste Argument aus (B) und (E), und für das zweite aus (B).

4 Formalisieren in Prädikatenlogik (PL1)

(1 + 2 + 2 + 3 = 8 Punkte)

Wir modellieren den Bestand einer Bücherei in der Prädikatenlogik erster Ordnung mit Gleichheit.

Gegeben sei dazu folgende prädikatenlogische Signatur:

$$\Sigma = (\{arist, autorVon\}, \{buch, autor, besitzt\}, \alpha)$$

Sie enthält

- das Konstantensymbol *arist*,
- das einstellige Funktionssymbol *autorVon* und
- die einstelligen Prädikatensymbole *buch*, *autor*, *besitzt*.

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 Bücher und Autoren.	
$I(arist)$	ist der Autor Aristoteles.
$I(autorVon)(x)$	ist der Autor von x, falls x ein Buch ist.
$I(autor)(x) = W$	gdw. x ein Autor ist.
$I(buch)(x) = W$	gdw. x ein Buch ist.
$I(besitzt)(x) = W$	gdw. x ein Buch ist, das die Bücherei besitzt.

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

a. Autoren sind keine Bücher und Bücher keine Autoren.

$$\forall x\ ((buch(x) \rightarrow \neg autor(x)) \land (autor(x) \rightarrow \neg buch(x)))$$

b. Die Bücherei besitzt mindestens ein Buch jedes Autors.

$$\forall a\ (autor(a) \rightarrow \exists b\ (a \doteq autorVon(b) \land besitzt(b)))$$

c. Die Bücherei besitzt mindestens zwei Bücher von Aristoteles.

$$\exists b_0\ \exists b_1\ (besitzt(b_0) \land besitzt(b_1) \land arist \doteq authorVon(b_0) \land arist \doteq autorVon(b_1) \land \neg(b_0 \doteq b_1))$$

d. Besitzt die Bücherei ein Buch eines Autors, besitzt sie jedes Buch dieses Autors.

$$\forall a\ (\exists b\ (a \doteq autorVon(b) \land besitzt(b)) \rightarrow \forall c\ (a \doteq autorVon(c) \land buch(c) \rightarrow besitzt(c)))$$

5 Resolution (3 + 5 = 8 Punkte)

a. Gegeben sei folgende prädikatenlogische Klausel mit Variablen x, z und Konstante a:

$$\{\neg p(z,a),\ \neg p(z,x),\ \neg p(x,z)\} \qquad (1)$$

Welche **drei** verschiedenen Klauseln lassen sich aus der Klausel (1) mit Hilfe der folgenden Faktorisierungsregel ableiten?

$$\frac{C}{\mu(C)} \quad \text{wobei} \begin{cases} C \text{ eine Klausel ist und} \\ \mu \text{ ein allgemeinster Unifikator von } \textbf{zwei oder mehr} \text{ Literalen in } C \end{cases}$$

Geben Sie jeweils auch den verwendeten allgemeinsten Unifikator μ an.

Hinweis: „Verschiedene Klauseln“ bedeutet hier, dass die Klauseln auch durch Variablenumbenennung nicht gleich gemacht werden können.

$\mu_1(C) = \boxed{\{\neg p(z,a),\ \neg p(a,z)\}}$ $\qquad$ $\mu_1 = \boxed{\mu = \{x/a\}}$

$\mu_2(C) = \boxed{\{\neg p(a,a)\}}$ $\qquad$ $\mu_2 = \boxed{\mu = \{x/a, z/a\}}$

$\mu_3(C) = \boxed{\{\neg p(z,a),\ \neg p(z,z)\}}$ $\qquad$ $\mu_3 = \boxed{\mu = \{x/z\}}$

b. Zeigen Sie mit Hilfe des Resolutionskalküls, dass die folgende prädikatenlogische Klauselmenge unerfüllbar ist. Dabei sind u, v, x, y, z Variablen und a ist eine Konstante.

Hinweis: Sie dürfen die Faktorisierungsregel verwenden. Starten Sie mit den Klauseln aus Teilaufgabe (a).

$$\{\quad \{\neg p(z,a),\ \neg p(z,x),\ \neg p(x,z)\},\quad \{p(u,f(u)),\ p(u,a)\},\quad \{p(f(v),v),\ p(v,a)\}\quad \}$$

Seien die gegebenen Klauseln mit $(1), (2), (3)$ bezeichnet.
(4) $\{\neg p(a,a)\}$ durch Faktorisierung aus (1)
(5) $\{\neg p(z,a),\ \neg p(a,z)\}$ durch Faktorisierung aus (1)
(6) $\{p(a,f(a))\}$ durch Resolution von $(4), (2)$
(7) $\{p(f(a),a)\}$ durch Resolution von $(4), (3)$
(8) $\{\neg p(f(a),a)\}$ durch Resolution von $(5), (6)$
(9) $\square$ durch Resolution von $(7), (8)$

6 Spezifikation mit der Java Modeling Language

(4 + 4 = 8 Punkte)

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

Hinweis: Der JML-Quantor `(\num_of int x; P(x))` zählt die Anzahl der Werte x (vom Typ `int`), für die die Bedingung $P(x)$ wahr ist.

```
public final class A {
    /*@ public normal_behavior
      @ requires 1 <= a.length;
      @ requires !(\exists int k; true; a.length == k * 2);
      @ requires (\forall int i; 0 <= i && i < a.length;
      @              (\forall int j; 0 <= j && j < a.length;
      @                              i != j ==> a[i] != a[j]));
      @ assignable \nothing;
      @ ensures (\exists int i; 0 <= i && i < a.length; \result == a[i]);
      @ ensures (\num_of int i; 0 <= i && i < a.length && a[i] < \result)
      @             == (a.length - 1) / 2;
      @ ensures (\num_of int i; 0 <= i && i < a.length && a[i] > \result)
      @             == (a.length - 1) / 2;
      @*/
    public static int m(int[] a) { ... }
}
```

Wenn `m` für

- ein von `null` verschiedenes Array `a` (implizit, nicht gefordert),
- dessen Länge ungerade sowie mindestens 1 ist und
- dessen Elemente paarweise verschiedenen sind,

aufgerufen wird, terminiert die Methode, verursacht keine Exception und ändert keine (existierenden) Speicherstellen auf dem Heap. Nach Ausführung von `m` gilt:

- Die Methode liefert als Ergebnis eine Zahl, die im Array enthalten ist.
- Das Ergebnis der Methode ist der Median (der wegen der Vorbedingung immer existiert) der Arrayelemente: Die Anzahl der Elemente, die jeweils größer bzw. kleiner als das Ergebnis sind, ist jeweils gleich der halben Länge (1 subtrahiert, da ungerade) des Arrays.

Fortsetzung 6 Spezifikation mit der Java Modeling Language

b. Gegeben sind eine Implementierung sowie ein Methodenvertrag der untenstehenden Methode `substring`, die einen Teilstring von `original` (gegeben als `char`-Array) in einem neuen `char`-Array zurückgibt. Der Methodenvertrag spezifiziert, dass die einzelnen Zeichen des neuen Strings denen von `original` ab dem Index `start` (inklusive) bis `end` (exklusive) entsprechen. Die Vorbedingung spezifiziert, dass `start` und `end` sich innerhalb des Arrays sowie in der richtigen Reihenfolge befinden.

Um den Vertrag verifizieren zu können, benötigt man eine Hilfsspezifikation für die Schleife. Die untenstehende Schleifeninvariante ist unvollständig. Ergänzen Sie sie zu einer korrekten **Invariante** für die Schleife, mit der die Nachbedingung des Methodenvertrags bewiesen werden kann.

Geben Sie zusätzlich eine korrekte Schleifen**variante** (`decreases`-Klausel) an, mit der die Terminierung der Methode bewiesen werden kann.

```
public final class A {
    /*@ public normal_behavior
      @ requires 0 <= start && start < original.length;
      @ requires start <= end;
      @ requires 0 <= end && end <= original.length;
      @ ensures \result.length == end - start;
      @ ensures (\forall int k; 0 <= k && k < \result.length;
      @                         \result[k] == original[start + k]);
      @ assignable \nothing;
      @*/
    public static char[] substring(char[] original, int start, int end) {
        char[] result = new char[end - start];
        /*@ loop_invariant start <= i && i <= end;
          @       // oder: 0 <= i - start && i - start <= result.length
          @ loop_invariant (\forall int j; start <= j && j < i;
          @                                result[j - start] == original[j]);
          @       // oder: (\forall int j; 0 <= j < i - start;
          @                                result[j] == original[start + j]);
          @ decreases end - i;
          @                        // oder: end - i + start;
          @                        // oder: original.length - i;
          @ assignable result[*];
          @*/
        for (int i = start; i < end; i++) {
            result[i - start] = original[i];
        }
        return result;
    }
}
```

7 Lineare Temporale Logik (LTL) (1+1+2) + 3 = 7 Punkte)

a. Im Folgenden sollen einige Sachverhalte im Zusammenhang mit Zugriffsrechten auf Dateien temporallogisch formalisiert werden. Verwenden Sie dafür folgende Signatur:

$$\Sigma = \{P_F, Q_G, R_G\}$$

mit der Bedeutung

P_F ist wahr gdw. Person P hat Zugriff auf Datei F

Die Bedeutung von Q_G, R_G ist analog definiert für Personen Q, R und Datei G.

Formalisieren Sie folgende Sachverhalte in LTL:

i. Person P hat niemals unendlich lange Zugriff auf Datei F.

$\boxed{\Box\Diamond\neg P_F}$

ii. Person P hat immer höchstens drei zusammenhängende Zeitschritte lang Zugriff auf die Datei F.

$\boxed{\Box(P_F \rightarrow \neg(\mathbf{X}P_F \wedge \mathbf{XX}P_F \wedge \mathbf{XXX}P_F))}$

iii. Person P hat jetzt Zugriff auf Datei F. Und spätestens, wenn P keinen Zugriff mehr auf F hat, haben Q und R Zugriff auf Datei G.

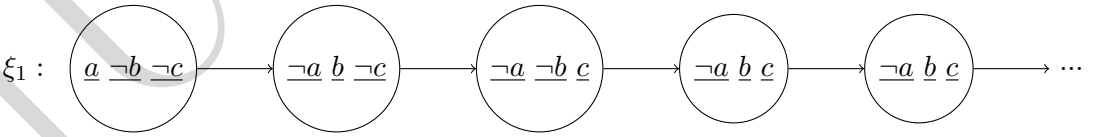
$\boxed{P_F \wedge (P_F \,\mathbf{U_W}\, (Q_G \wedge R_G))}$

b. Es sei die folgende LTL-Formel gegeben:

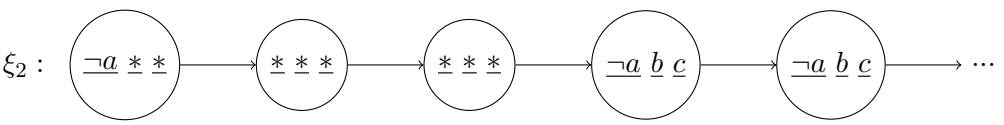
$$\phi := \quad \Box(\neg a \vee \neg c) \;\wedge\; (c \,\mathbf{U}\, \neg b) \;\wedge\; \Diamond(\neg b \wedge c) \;\wedge\; a \;\wedge\; \mathbf{X}b \tag{2}$$

Gegeben sei nun jeweils eine Omega-Struktur ξ_i, in der die Belegung der Variablen ab dem vierten Zustand vorgegeben ist und sich danach nicht mehr verändert.

i. Vervollständigen Sie die ersten drei Zustände von ξ_1 mit Belegungen der Variablen a, b, c, so dass $\xi_1 \models \phi$.



ii. Vervollständigen Sie ξ_2, so dass $\xi_2 \not\models \phi$.



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