Aufgabe 3.1 [Model Checking] **8 Punkte**

Model Checking ist ein Teilbereich der Informatik, der sich mit der algorithmischen Systemverifikation beschäftigt; es soll also für ein gegebenes Computerprogramm automatisch überprüft werden, ob jede Ausführung des Programms eine gegebene Eigenschaft erfüllt.
Zu diesem Zweck modelliert man das gegebene Programm durch eine Kripkestruktur, in der jede Welt einen möglichen *Programmzustand* darstellt und eine Kante von Welt s nach Welt t führt, wenn es möglich ist, dass in einer Ausführung des Programms Zustand t auf Zustand s folgt. Ein Zustand kann dabei auch mehrere Nachfolgezustände haben, zum Beispiel weil Benutzereingaben oder Systemvariablen abgefragt werden.
In der Praxis werden diese Kripkestrukturen meist aus dem Programmcode automatisch generiert und können sehr komplex werden; die aussagenlogischen Variablen, die in einer Welt gelten können, entsprechen dort möglichen Werten (oder Wertebereichen) von Programmvariablen. Wir betrachten hier eine stark vereinfachte Variante, in der in einem Programmzustand die folgenden Aussagen gelten können:

- S: es wird auf den Speicher zugegriffen
- W: es wird auf eine Eingabe gewartet
- E: es wird eine Eingabe gemacht
- A: es erfolgt eine Ausgabe

Eine erwünschte oder unerwünschte Eigenschaft eines Programms lässt sich dann als modallogische Formel modellieren, welche genau dann in einer Welt s gilt, wenn der durch s modellierte Zustand die entsprechende Eigenschaft hat. Beispielsweise gilt die Formel $E \wedge \Diamond A$ in jeder Welt, für die in dem zugehörigen Programmzustand eine Eingabe gemacht wird und in mindestens einem seiner möglichen Nachfolgezustände eine Ausgabe erfolgt.

a) Geben Sie für die folgenden Eigenschaften jeweils eine modallogische Formel an, die diese beschreibt. Denken Sie daran, Ihre Antworten zu begründen. **(4 Punkte)**

(i) Wenn im aktuellen Zustand auf eine Eingabe gewartet wird, dann wird in jedem Nachfolgezustand entweder auf eine Eingabe gewartet oder eine Eingabe gemacht. **[2 Punkte]**

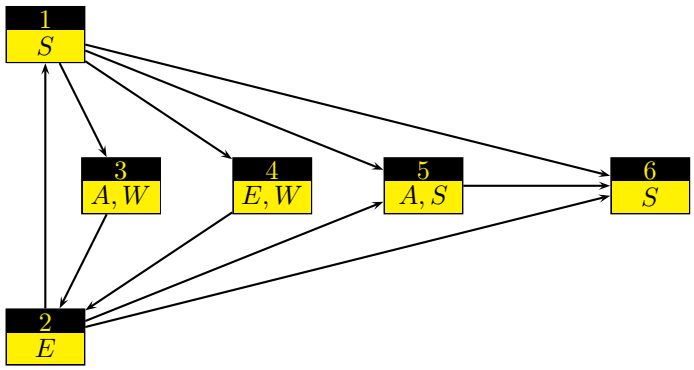
(ii) Wenn auf den aktuellen Zustand ein Zustand mit einem Speicherzugriff folgen kann, dann erfolgt im aktuellen Zustand eine Ausgabe, und im nächsten und übernächsten Zustand wird mit Sicherheit weder auf eine Eingabe gewartet noch eine Eingabe gemacht. **[2 Punkte]**

b) Welche Eigenschaft wird von der Formel

$$\Box(E \wedge \neg W) \rightarrow (W \vee \Box\Diamond S)$$

ausgedrückt? Beschreiben Sie diese Eigenschaft so wie in Teilaufgabe (a) mit eigenen Worten. **(2 Punkte)**

c) Die folgende Kripkestruktur modelliert ein fiktives Programm.



In den Zuständen dieses Programms soll nun überprüft werden, ob eine Ausgabe genau dann erfolgt wenn keine Eingabe erfolgt und jeder Nachfolgezustand, in dem eine Eingabe gemacht wird, von mindestens einem Zustand mit einem Speicherzugriff gefolgt wird.

Diese Eigenschaft lässt sich durch die folgende modallogische Formel ψ modellieren:

$$\psi = A \leftrightarrow (\neg E \wedge \Box(E \rightarrow \Diamond S))$$

Ermitteln Sie für **jede** Teilformel von ψ, in welchen Welten der obigen Kripkestruktur sie gilt. Schließen Sie dann daraus, in welchen Welten die Gesamtformel gilt. **(2 Punkte)**

Bemerkung: In der Praxis werden Eigenschaften üblicherweise in ausdrucksstärkeren Logiken als der Modallogik modelliert, z.B. in Temporallogiken wie LTL (vgl. Zusatzaufgabe).

Lösungsvorschlag:

a) **(i)** Die Teilaussage „es wird entweder auf eine Eingabe gewartet oder eine Eingabe gemacht“ kann modelliert werden durch die Formel $W \leftrightarrow \neg E$; die Teilaussage „es wird in jedem Nachfolgezustand entweder auf eine Eingabe gewartet oder eine Eingabe gemacht“ entspricht der Fomel $\Box(W \leftrightarrow \neg E)$. Insgesamt ergibt sich die Formel

$$W \rightarrow \Box(W \leftrightarrow \neg E).$$

(ii) Die Bedingung „im nächsten und übernächsten Zustand wird mit Sicherheit weder auf eine Eingabe gewartet noch eine Eingabe gemacht“ kann modelliert werden durch $\Box(\neg W \wedge \neg E \wedge \Box(\neg W \wedge \neg E))$; die Teilaussage „im aktuellen Zustand erfolgt eine Ausgabe, und im nächsten und übernächsten Zustand wird mit Sicherheit weder auf eine Eingabe gewartet noch eine Eingabe gemacht“ entspricht $A \wedge \Box(\neg W \wedge \neg E \wedge \Box(\neg W \wedge \neg E))$,

Die Teilaussage „auf den aktuellen Zustand kann ein Zustand mit einem Speicherzugriff folgen“ entspricht der Formel $\Diamond S$.

Zusammen erhalten wir die Formel

$$\Diamond S \rightarrow \big(A \wedge \Box(\neg W \wedge \neg E \wedge \Box(\neg W \wedge \neg E))\big).$$

b) Die Teilformel $\Box(E \wedge \neg W)$ besagt, dass in allen möglichen Nachfolgezuständen des aktuellen Zustands eine Eingabe erfolgt und nicht auf eine Eingabe gewartet wird. Die Teilformel $\Box\Diamond S$ besagt, dass jeder mögliche Nachfolgezustand des aktuellen Zustands mindestens einen Nachfolgezustand hat, in dem ein Speicherzugriff erfolgt. Insgesamt ergibt sich also folgende Eigenschaft:

„Wenn in jedem Nachfolgezustad des aktuellen Zustands eine Eingabe erfolgt und nicht auf eine Eingabe gewartet wird, dann wird im aktuellen Zustand auf eine Eingabe gewartet oder jeder Nachfolgezustand des aktuellen Zustands hat einen Nachfolgezustand, in dem auf den Speicher zugegriffen wird.“

c) Analog zur Vorlesung stellen wir eine Tabelle auf:

	1	2	3	4	5	6
E	×	✓	×	✓	×	×
$\neg E$	✓	×	✓	×	✓	✓
S	✓	×	×	×	✓	✓
$\Diamond S$	✓	✓	×	×	✓	×
$E \to \Diamond S$	✓	✓	✓	×	✓	✓
$\Box(E \to \Diamond S)$	×	✓	✓	✓	✓	✓
$\neg E \wedge \Box(E \to \Diamond S)$	×	×	✓	×	✓	✓
A	×	×	✓	×	✓	×
ψ	✓	✓	✓	✓	✓	×

Somit gilt die durch ψ beschriebene Eigenschaft in allen Welten außer Welt 6.

Aufgabe 3.2 [Äquivalent! Oder doch nicht?] **2 Punkte**

Entscheiden Sie, ob die unten angegebenen Formeln φ und ψ äquivalent sind oder nicht. Falls sie äquivalent sind, zeigen Sie die Äquivalenz, indem Sie die in der Vorlesung eingeführten Äquivalenzen verwenden. Begründen Sie jeden Zwischenschritt! Falls sie nicht äquivalent sind, so geben Sie eine Kripkestruktur und eine Welt s in dieser Struktur an, so dass eine der Formeln in s gilt und die andere nicht. Erklären Sie genau, warum die eine Formel in s gilt und die andere nicht!

$$\varphi = \Diamond(\Box A \to (\Box B \to \Diamond C)) \text{ und } \psi = \Diamond\Diamond((A \wedge B) \to C)$$

Lösungsvorschlag:

Die Formeln sind äquivalent, wie die folgende Kette von Äquivalenzen zeigt:

$$\begin{aligned}
\varphi &= \Diamond(\Box A \to (\Box B \to \Diamond C)) \\
&\equiv \Diamond(\neg\Box A \vee (\neg\Box B \vee \Diamond C)) && \text{(Implikation auflösen)} \\
&\equiv \Diamond(\Diamond\neg A \vee \Diamond\neg B \vee \Diamond C)) && \text{(Dualität, unnötige Klammern entfernen)} \\
&\equiv \Diamond\Diamond(\neg A \vee \neg B \vee C) && \text{(Distributivität)} \\
&\equiv \Diamond\Diamond(\neg(A \wedge B) \vee C) && \text{(DeMorgan)} \\
&\equiv \Diamond\Diamond((A \wedge B) \to C) && \text{(Implikation einführen)} \\
&= \psi
\end{aligned}$$

Aufgabe 3.3 [Aussagenlogischer Tableaukalkül] **4 Punkte**

Sei die Formel

$$\varphi = A \wedge \Big(\neg(B \to A) \vee \big(\neg(B \wedge C) \wedge (C \vee B)\big)\Big)$$

gegeben.

a) Konstruieren Sie ein saturiertes Tableau T für φ. *Wählen Sie dazu bei jeder Regelanwendung jeweils einen möglichst dicht an der Wurzel des Tableaus liegenden unmarkierten Knoten.* Sobald sich auf einem Zweig des Tableaus ein Widerspruch ergibt, können Sie diesen Zweig direkt schließen und müssen ihn nicht weiter verfolgen. **(3 Punkte)**

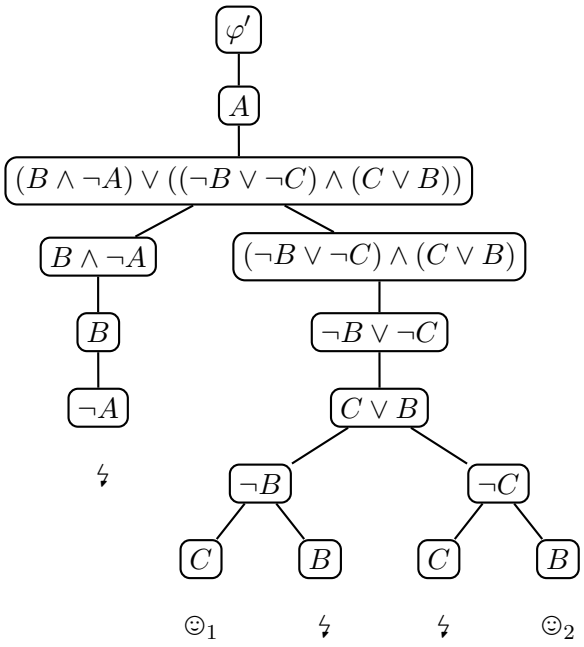
b) Entscheiden Sie mit Hilfe des Tableaus T, ob die Formel φ erfüllbar ist. Wenn die Formel erfüllbar ist, markieren Sie die offenen Blätter und schreiben an jedes offene Blatt eine zu ihm korrespondierende erfüllende Belegung. **(1 Punkt)**

Lösungsvorschlag:

a) Wir konstruieren jetzt ein saturiertes Tableau für φ. Dazu muss φ zuerst in NNF gebracht werden:

$$\begin{aligned}
\varphi &= A \wedge \big(\neg(B \to A) \vee (\neg(B \wedge C) \wedge (C \vee B))\big) \\
&\equiv A \wedge \big(\neg(\neg B \vee A) \vee (\neg(B \wedge C) \wedge (C \vee B))\big) && \text{(Abkürzungen auflösen)} \\
&\equiv A \wedge \big((B \wedge \neg A) \vee ((\neg B \vee \neg C) \wedge (C \vee B))\big) && \text{(DeMorgan)} \\
&= \varphi'
\end{aligned}$$

Das folgende Tableau ist ein mögliches saturiertes Tableau für φ, welches sich mit der vorgegebenen Auswertungsreihenfolge ergibt.



b) Die mit den Smileys markierten Pfade sind offen. Alle übrigen Pfade sind geschlossen. Da das Tableau saturiert ist und mindestens einen offenen Pfad hat, folgt aus Satz 6.2, dass φ erfüllbar ist. Die Literale, die auf dem mit $\☺_1$ markierten offenen Pfad vorkommen, sind A, $\neg B$ und C. Daraus folgt, dass die Belegung α_1 mit $\alpha_1(A) = 1$, $\alpha_1(B) = 0$ und $\alpha_1(C) = 1$ eine

erfüllende Belegung für φ ist. Entsprechend folgt aus dem mit ☺_2 markierten offenen Pfad, dass die Belegung α_2 mit $\alpha_2(A) = 1$, $\alpha_2(B) = 1$ und $\alpha_2(C) = 0$ eine erfüllende Belegung für φ ist.

Aufgabe 3.4 [Modallogisches Folgern] **6 Punkte**

In dieser Aufgabe sollen Sie entscheiden, ob aus der Formel

$$\varphi = \Box(\neg\Box A \wedge \Diamond(B \to A))$$

die Formel $\psi = \Box\Diamond\neg B$ folgt.

a) Stellen Sie eine Formel in NNF auf, die genau dann unerfüllbar ist, wenn ψ aus φ folgt. **(2 Punkte)**

b) Beweisen oder widerlegen Sie mit Hilfe des modallogischen Tableaukalküls, dass ψ aus φ folgt. *Wählen Sie dazu bei jeder Regelanwendung jeweils einen möglichst dicht an der Wurzel des Tableaus liegenden unmarkierten Knoten.* Sobald sich auf einem Zweig des Tableaus ein Widerspruch ergibt, können Sie diesen Zweig direkt schließen und müssen ihn nicht weiter verfolgen.

Wenn ψ nicht aus φ folgt, geben Sie eine Kripke-Struktur $\mathcal{K}$ und eine Welt s an, so dass $\mathcal{K}, s \models \varphi$ und $\mathcal{K}, s \not\models \psi$. **(4 Punkte)**

Lösungsvorschlag:

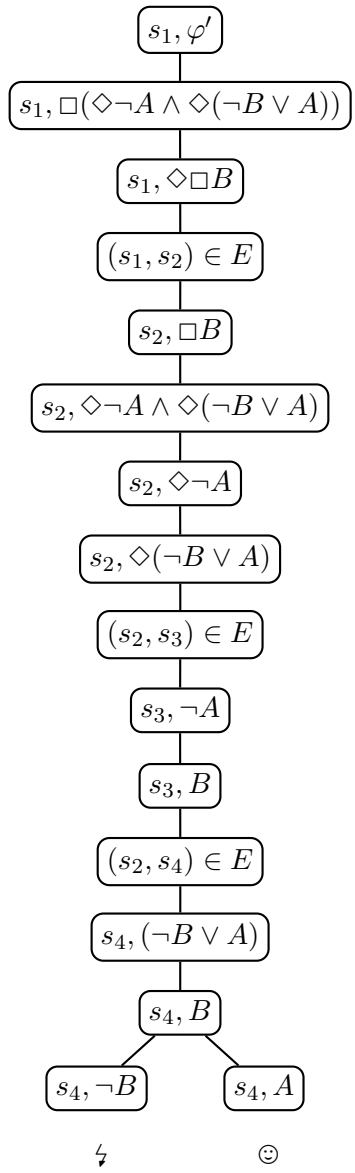
a) Wie in dem Beispiel auf Folie 6.19 folgt ψ genau dann aus φ, wenn $\varphi \wedge \neg\psi$ unerfüllbar ist. Die gesuchte Formel ist also die Negationsnormalform von $\varphi \wedge \neg\psi$:

$$\begin{aligned}
\varphi \wedge \neg\psi &= \Box(\neg\Box A \wedge \Diamond(B \to A)) \wedge \neg\Box\Diamond\neg B \\
&\equiv \Box(\neg\Box A \wedge \Diamond(\neg B \vee A)) \wedge \neg\Box\Diamond\neg B && \text{(Abkürzungen auflösen)} \\
&\equiv \Box(\Diamond\neg A \wedge \Diamond(\neg B \vee A)) \wedge \Diamond\neg\Diamond\neg B && \text{(Dualität)} \\
&\equiv \Box(\Diamond\neg A \wedge \Diamond(\neg B \vee A)) \wedge \Diamond\Box B && \text{(Dualität)} = \varphi'
\end{aligned}$$

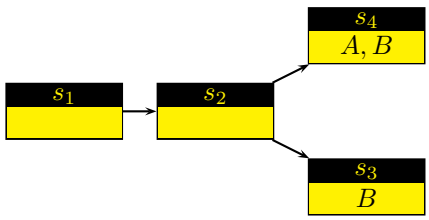
b) Das Tableau, welches der Kalkül mit der vorgegebenen Auswertungsreihenfolge für die Formel

$$\varphi' = \Box(\Diamond\neg A \wedge \Diamond(\neg B \vee A)) \wedge \Diamond\Box B$$

erzeugt, ist das folgende:



Es gibt also einen offenen Pfad. Damit ist φ' erfüllbar, und somit folgt ψ nicht aus φ. Der mit ☺markierte offene Pfad ergibt die folgende Kripkestruktur $\mathcal{K}$, für die gilt $\mathcal{K}, s_1 \models \varphi$ aber $\mathcal{K}, s_1 \not\models \psi$:



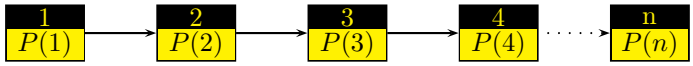
Zusatzaufgabe [The Times They Are a-Changin'...]
Wir wollen in dieser Aufgabe zeitlich getaktete Systeme beschreiben. Als Beispiel verwenden wir einen Drucker, für den wir sagen wollen: Wird der Einschaltknopf des Druckers betätigt, so ist er irgendwann bereit und bleibt es danach auch. Ist der Drucker bereit und erhält einen Auftrag, so

beginnt er sofort danach zu drucken und druckt weiter, bis er keinen Auftrag mehr hat oder sein Papier alle ist.
Für die Modellierung solcher Situationen eignet sich die *lineare temporale Logik (LTL)*. LTL-Formeln sind syntaktisch von folgender Form:

- Jede Proposition $p \in \text{Prop}$ ist eine LTL-Formel.
- Sind ψ und ζ LTL-Formeln, so sind auch
  - $\neg\psi$, $\psi \wedge \zeta$ und $\psi \vee \zeta$,
  - $\mathbf{X}\psi$, $\mathbf{F}\psi$, $\mathbf{G}\psi$ und $\psi\mathbf{U}\zeta$

  LTL-Formeln.

Hierbei ist Prop eine Menge von Propositionen. Propositionen entsprechen den Variablen in der Modallogik. Die Symbole **X**, **F**, **G** und **U** heißen temporale Operatoren.
Im Allgemeinen können LTL-Formeln in beliebigen Kripke-Strukturen ausgewertet werden. Zur Einfachheit beschränken wir uns hier auf lineare Strukturen:



Lineare Strukturen sind Kripkestrukturen mit $V = \{1, \ldots, n\}$ und $E = \{(1,2), (2,3), \ldots, (n-1,n)\}$. Die Welt 1 heißt *initiale Welt*.
Die Semantik von LTL Formeln für solche Strukturen ist induktiv definiert. Eine lineare Struktur $\mathcal{K}$ erfüllt eine Formel φ in Welt i, geschrieben $\mathcal{K}, i \models \varphi$, falls:

(Atom)	$\varphi := p$	und	$p \in P(i)$
(Negation)	$\varphi := \neg\psi$	und	nicht $\mathcal{K}, i \models \psi$
(und)	$\varphi := \psi \wedge \zeta$	und	$\mathcal{K}, i \models \psi$ und $\mathcal{K}, i \models \zeta$
(oder)	$\varphi := \psi \vee \zeta$	und	$\mathcal{K}, i \models \psi$ oder $\mathcal{K}, i \models \zeta$
(nächste Welt)	$\varphi := \mathbf{X}\psi$	und	$i+1$ existiert und $\mathcal{K}, i+1 \models \psi$
(in der Zukunft)	$\varphi := \mathbf{F}\psi$	und	es gibt $j \geq i$, so dass $\mathcal{K}, j \models \psi$
(immer)	$\varphi := \mathbf{G}\psi$	und	für alle $j \geq i$ gilt $\mathcal{K}, j \models \psi$
(solange ... bis)	$\varphi := \psi\mathbf{U}\zeta$	und	es gibt $j \geq i$, so dass $\mathcal{K}, j \models \zeta$, und für alle k mit $j > k \geq i$ gilt $\mathcal{K}, k \models \psi$.

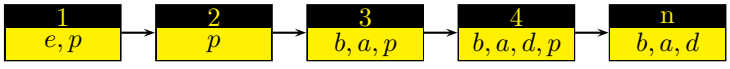
Eine Struktur $\mathcal{K}$ ist ein *Modell von* φ, falls $\mathcal{K}, 1 \models \varphi$ gilt.
Das obige Beispiel lässt sich nun wie folgt mit Hilfe von LTL modellieren. Als Propositionen nehmen wir $\{e, b, a, d, p\}$ mit den intendierten Bedeutungen

- e: Einschaltknopf wird betätigt.
- b: Der Drucker ist bereit.
- a: Der Drucker hat einen Auftrag.
- d: Der Drucker druckt.
- p: Der Drucker hat noch Papier.

Die Formel

$$\mathbf{G}((e \rightarrow \mathbf{F}(b \wedge \mathbf{G}b)) \wedge ((b \wedge a) \rightarrow \mathbf{X}(d\mathbf{U}(\neg a \vee \neg p))))$$

beschreibt nun die oben angegebene Situation. Strukturen, die diese Formel in Welt 1 erfüllen, sind beispielsweise



und

(Man sieht, dass die Formel keineswegs nur reale Abläufe widerspiegelt.)

a) Ist die folgende Struktur ein Modell der Formel $\varphi = \mathbf{F}(b \wedge ((\neg e)\mathbf{U}d))$?

b) Geben Sie eine LTL-Formel ψ über den Propositionen $\{p, q, r\}$ an, die folgende Situation beschreibt:

- Wenn eine Welt die Proposition r trägt, so trägt die darauffolgende Welt mindestens eine der Propositionen $\{p, q\}$.
- Trägt eine Welt höchstens zwei der drei Propositionen $\{p, q, r\}$, so folgt darauf irgendwann eine Welt mit der Proposition r; dazwischen muss allerdings mindestens eine Welt liegen, die keine Propositionen trägt.