# 1. Übungsblatt (mit Lösungen)

**3.0 VU Formale Modellierung**

Gernot Salzer, Marion Scholz

2. November 2016

## Aufgabe 1 (3 Punkte)

Geben Sie für die folgenden Schlussfolgerungen die zugrundeliegende Inferenzregel an und stellen Sie fest, ob diese gültig ist. Wenn ja, geben Sie unter Verwendung von Alltagsbegriffen eine weitere Schlussfolgerung an, die derselben Regel folgt. Wenn nein, modifizieren Sie die Inferenzregel möglichst geringfügig, um eine gültige Regel zu erhalten, und geben Sie dann eine konkrete Schlussfolgerung mit Alltagsbegriffen an, die dieser Regel entspricht.

(a) Kinder singen gerne, kein Opernsänger ist ein Kind, daher singt kein Opernsänger gerne.

(b) Toni ist nicht schwanger, Männer können nicht schwanger werden, daher ist Toni ein Mann.

(c) In Frankreich spricht man deutsch, Paris liegt in Frankreich, daher spricht man in Paris deutsch.

### Lösung

(a)

$$\begin{array}{l}\text{Alle Kinder singen gerne.}\\ \text{Kein Opernsänger ist ein Kind.}\\ \hline \text{Kein Opernsänger singt gerne.}\end{array} \qquad \text{Inferenzregel: } \begin{array}{l}\text{Alle } x \text{ machen gerne } y.\\ \text{Kein } z \text{ ist ein } x.\\ \hline \text{Kein } z \text{ macht gerne } y.\end{array}$$

Diese Inferenzregel ist **nicht gültig**. Vertauscht man die zweite Prämisse mit der Konklusion, erhält man eine gültige Inferenzregel:

$$\begin{array}{l}\text{Alle } x \text{ machen gerne } y.\\ \text{Kein } z \text{ macht gerne } y.\\ \hline \text{Kein } z \text{ ist ein } x.\end{array}$$

Ein Beispiel, dem diese Inferenzregel zugrunde liegt:

$$\begin{array}{l}\text{Alle Formel-1-Fahrer fahren gerne schnell.}\\ \text{Keine Schnecke fährt gerne schnell.}\\ \hline \text{Keine Schnecke ist ein Formel-1-Fahrer.}\end{array}$$

(b) Toni ist nicht schwanger.
Männer können nicht schwanger werden.
Toni ist ein Mann.

Inferenzregel: $x$ ist nicht $y$.
Alle $z$ sind nicht $y$.
$x$ ist ein $z$.

Diese Inferenzregel ist **nicht gültig**. Vertauscht man die erste Prämisse mit der Konklusion, erhält man eine gültige Inferenzregel:

$x$ ist ein $z$.
Alle $z$ sind nicht $y$.
$x$ ist nicht $y$.

Ein Beispiel, dem diese Inferenzregel zugrunde liegt:

Pia ist ein Mensch.
Alle Menschen können nicht fliegen.
Pia kann nicht fliegen.

(c) Die angegebenen Aussagen enthalten versteckte Quantoren und lassen mehrere Interpretationen zu. Eine Möglichkeit ist die folgende.

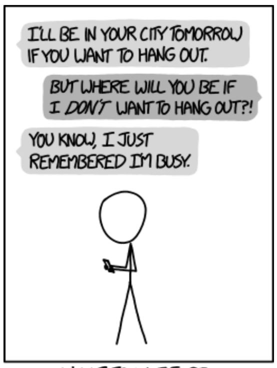
In allen Teilen Frankreichs spricht man deutsch.
Paris ist ein Teil Frankreichs.
In Paris spricht man deutsch.

I.r.: Für alle $x$ gilt $y$.
$z$ ist ein $x$.
Für $z$ gilt $y$.

Diese Inferenzregel ist **gültig**. Andere Schlussfolgerung mit derselben Inferenzregel:

Alle Fische können schwimmen.
Der Hai ist ein Fisch.
Der Hai kann schwimmen.

## Aufgabe 2 (5 Punkte)



WHY I TRY NOT TO BE PEDANTIC ABOUT CONDITIONALS.

(a) Analysieren Sie den Dialog im Cartoon anbei. Übersetzen Sie die erste Aussage sowohl aus Sicht der sendenden Person als auch aus jener der empfangenden in eine logische Formel. Erklären Sie, warum die zweite Person mit dieser Frage antwortet.
(Quelle: xkcd web comic, xkcd.com/1652)

Analysieren Sie die folgenden Sätze und identifizieren Sie ihre logische Struktur sowie die Elementaraussagen.

(b) Max hat sich die Haare nicht gewaschen.

(c) Ich bin müde, hungrig, und langweilig ist mir auch.

(d) Flora wird die Prüfung nicht bestehen, wenn sie nicht mitlernt und nicht aufpasst.

(e) Ich möchte zu Mittag ein Schnitzel oder einen Tofuburger.

(f) Nur wenn er den Einstiegstest besteht, darf er ein Informatikstudium beginnen.

(g) Wenn der Mond ein gelber Käse ist, ist 6 eine Primzahl.

(h) Ich kann nicht gleichzeitig denken und sprechen.

(i) Peter erwischt den Zug oder auch nicht.

## Lösung

(a) *I'll be in your city tomorrow if you want to hang out.*
Die erste Person meint mit diesem Satz:
„Ich bin morgen in der Stadt. Wir können uns verabreden, wenn du fortgehen möchtest."
$A$ ... Ich bin morgen in der Stadt.
$B$ ... Wir können uns verabreden.
$C$ ... Du möchtest fortgehen.
Struktur: $A$, und $B$ falls $C$.
Formel: $A \wedge (B \equiv C)$ (eher nicht $B \subset C$, da sich die beiden wohl nicht verabreden werden, wenn die zweite Person nicht fortgehen möchte).

Die zweite Person hat offenbar wenig über die Mehrdeutigkeit der natürlichen Sprache gehört und versteht den Satz als kausale Implikation:
„Ich komme morgen in die Stadt, wenn du fortgehen möchtest."
Struktur: $A$, wenn $C$.
Formel: $A \subset C$

In dieser engen Interpretation hat die erste Person noch keine Aussage darüber gemacht, ob sie in der Stadt sein wird, wenn die zweite Person nicht fortgehen möchte, da die Formel $A \subset C$ wahr ist, sobald $C$ falsch ist, unabhängig von $A$. Daher ergibt es Sinn zu fragen, was die erste Person in diesem Fall tun wird.

(b) *Max hat sich die Haare nicht gewaschen.*
$A$ ... Max hat sich die Haare gewaschen.
Struktur: Nicht $A$.
Formel: $\neg A$

(c) *Ich bin müde, hungrig, und langweilig ist mir auch.*
$A$ ... Ich bin müde.
$B$ ... Ich bin hungrig.
$C$ ... Mir ist langweilig.
Struktur: $A$ und $B$ und $C$.
Formel: $A \wedge B \wedge C$

(d) *Flora wird die Prüfung nicht bestehen, wenn sie nicht mitlernt und nicht aufpasst.*
$A$ ... Flora wird die Prüfung bestehen.
$B$ ... Flora lernt mit.
$C$ ... Flora passt auf.
Struktur: Nicht $A$, wenn nicht $B$ und nicht $C$.
Formel: $\neg A \subset (\neg B \wedge \neg C)$

(e) *Ich möchte zu Mittag ein Schnitzel oder einen Tofuburger.*
$A$ ... Ich möchte zu Mittag ein Schnitzel.
$B$ ... Ich möchte zu Mittag einen Tofuburger.
Struktur: $A$ oder $B$.
Formel: $A \not\equiv B$ (Eher nicht $A \vee B$, da man bei dieser Aussage in der Regel davon ausgeht, dass nur eine Speise gegessen wird.)

(f) *Nur wenn er den Einstiegstest besteht, darf er ein Informatikstudium beginnen.*
$A$ ... Er besteht den Einstiegstest.
$B$ ... Er darf ein Informatikstudium beginnen.
Struktur: Nur wenn $A$, dann $B$. Äquivalent: Wenn nicht $A$, dann nicht $B$.
Formel: $A \subset B$ bzw. $\neg A \supset \neg B$

Beachten Sie, dass sich durch das Wort „nur" die Richtung der Implikation umdreht. Die Aussage ist äquivalent zu: Wenn er ein Informatikstudium beginnen durfte, hat er den Einstiegstest bestanden. Die Formulierung „nur wenn – dann" ist ziemlich eindeutig als Implikation und nicht als Äquivalenz zu verstehen. Das Bestehen des Einstiegstests ist notwendig, aber nicht hinreichend: Auch eine fehlende Matura könnte ein Informatikstudium verhindern.

(g) *Wenn der Mond ein gelber Käse ist, ist 6 eine Primzahl.*
$A$ ... Der Mond ist ein gelber Käse.
$B$ ... 6 ist eine Primzahl.
Struktur: Wenn $A$, dann $B$.
Formel: $A \supset B$

(h) *Ich kann nicht gleichzeitig denken und sprechen.*
$A$ ... Ich kann denken.
$B$ ... Ich kann sprechen.
Struktur: $A$ nicht gleichzeitig mit $B$.
Formel: $\neg(A \wedge B)$, bzw. äquivalent dazu $A \uparrow B$, $A \supset \neg B$ oder $B \supset \neg A$

(i) *Peter erwischt den Zug oder auch nicht.*
$A$ ... Peter erwischt den Zug.
Struktur: Entweder $A$ oder nicht $A$. Das Oder kann auch inklusiv interpretiert werden, da $A$ nicht gleichzeitig wahr und falsch sein kann.
Formel: $A \not\equiv \neg A$ bzw. $A \vee \neg A$

## Aufgabe 3 **(3 Punkte)**

(a) Zeigen Sie, dass die Menge $\{\mathsf{and}, \mathsf{iff}, \mathsf{false}\}$ vollständig ist für die Klasse der aussagenlogischen Funktionen.

(b) Zeigen Sie, dass die Menge $\{\mathsf{and}, \mathsf{or}\}$ nicht vollständig ist.

### Lösung

(a) Wir wissen aus der Vorlesung, dass die Funktionsmenge $\{\mathsf{and}, \mathsf{not}\}$ vollständig ist. Es reicht daher zu zeigen, dass die Funktion $\mathsf{not}$ durch die Funktionen $\{\mathsf{and}, \mathsf{iff}, \mathsf{false}\}$ darstellbar ist. Tatsächlich gilt $\mathsf{not}(x) = \mathsf{iff}(x, \mathsf{false})$, wie sich durch Auswertung der beiden Seiten in den zwei möglichen Wahrheitsbelegungen überprüfen lässt:

| $x$ | $\mathsf{not}(x)$ | | $=$ | $\mathsf{iff}(x,\mathsf{false})$ | | |
|---|---|---|---|---|---|---|
| 0 | **1** | 0 | ✓ | **1** | 0 | 0 |
| 1 | **0** | 1 | ✓ | **0** | 1 | 0 |

(b) Es genügt von einer einzigen Funktion zu zeigen, dass sie nicht durch $\mathsf{and}$ und $\mathsf{or}$ darstellbar ist. Wir untersuchen, welche einstelligen Funktionen darstellbar sind, indem wir vom einzigen Argument $x$ ausgehend die beiden Funktionen in beliebiger Reihenfolge anwenden. Wir stellen fest, dass $\mathsf{and}(x, x) = x$ und $\mathsf{or}(x, x) = x$ gilt (Idempotenz). Wir stellen daher folgende Behauptung auf.

*Behauptung:* Jeder Ausdruck bestehend aus $\mathsf{and}$, $\mathsf{or}$ und $x$ ist äquivalent zu $x$.

Der Beweis erfolgt induktiv nach der Anzahl $n$ der Anwendungen der Funktionen $\mathsf{and}$ bzw. $\mathsf{or}$.

*Induktionsanfang $n = 0$:* Der einzige Ausdruck ohne Anwendung von $\mathsf{and}$ und $\mathsf{or}$ ist $x$ selber. Dafür gilt unsere Behauptung.

*Induktionshypothese:* Unsere Behauptung gelte für alle Ausdrücke mit $n$ oder weniger Anwendungen von $\mathsf{and}$ bzw. $\mathsf{or}$.

*Induktionsschritt:* Wir zeigen, dass unter der Annahme, dass die Induktionshypothese zutrifft, unsere Behauptung auch für Ausdrücke mit $n + 1$ Anwendungen von $\mathsf{and}$ bzw. $\mathsf{or}$ gilt. Wir haben also einen Ausdruck $\mathsf{and}(f(x), g(x))$ bzw. $\mathsf{or}(f(x), g(x))$ vor uns, bei dem sowohl $f$ als auch $g$ mit $n$ oder weniger Anwendungen von $\mathsf{and}$ bzw. $\mathsf{or}$ definiert sind. Laut Hypothese ist jeder der beiden Ausdrücke äquivalent zu $x$. Wie wir oben aber festgestellt haben, liefert $\mathsf{and}(x, x)$ bzw. $\mathsf{or}(x, x)$ mit dem Argument $x$ wieder nur $x$.

Da somit die einzigen darstellbaren einstelligen Funktionen äquivalent zu $x$ (identische Abbildung) sind, ist z.B. die einstellige Funktion $\mathsf{not}$ nicht darstellbar. Die Menge $\{\mathsf{and}, \mathsf{or}\}$ ist daher nicht funktional vollständig.

## Aufgabe 4 (3 Punkte)

Sei $\mathcal{M}$ die kleinste Menge mit folgenden Eigenschaften:

(m1) $\{\mathtt{a}, \mathtt{b}\} \subseteq \mathcal{M}$

(m2) Wenn $u \in \mathcal{M}$, dann auch $u\texttt{+}u \in \mathcal{M}$.

(m3) Wenn $u, v \in \mathcal{M}$, dann auch $\texttt{*}u\texttt{*}v \in \mathcal{M}$.

(a) Geben Sie die Mengen $\mathcal{M}_0$, $\mathcal{M}_1$ und $\mathcal{M}_2$ der stufenweise Konstruktion von $\mathcal{M}$ an. Die Menge $\mathcal{M}_2$ enthält bereits mehr als 60 Elemente; es genügt, 10 typische Elemente anzugeben.

(b) Zeigen Sie, dass die Zeichenkette `**a*b+b+*a*b+b*a` in der Menge $\mathcal{M}$ liegt.

(c) Erklären Sie, warum die Zeichenkette `*b*a*b+*b*a*b` nicht in der Menge $\mathcal{M}$ liegen kann.

## Lösung

(a) $\mathcal{M}_0 = \{$`a`, `b`$\}$,
$\mathcal{M}_1 = \mathcal{M}_0 \cup \{$`a+a`, `b+b`, `*a*a`, `*a*b`, `*b*a`, `*b*b`$\}$,
$\mathcal{M}_2 = \mathcal{M}_1 \cup \{$`a+a+a+a`, `b+b+b+b`, `*a*a+*a*a`, `*a*b+*a*b`, `*b*a+*b*a`,
`*b*b+*b*b`, `*a*a+a`, `*a*b+b`, `*a**a*a`, `*a**a*b`,
`*a**b*a`, `*a**b*b`, `*b*a+a`, `*b*b+b`, `*b**a*a`,
`*b**a*b`, `*b**b*a`, `*b**b*b`, `*a+a*a`, `*a+a*b`,
`*a+a*a+a`, `*a+a*b+b`, `*a+a**a*a`, `*a+a**a*b`, `*a+a**b*a`,
`*a+a**b*b`, `*b+b*a`, `*b+b*b`, `*b+b*a+a`, `*b+b*b+b`,
`*b+b**a*a`, `*b+b**a*b`, `*b+b**b*a`, `*b+b**b*b`, `**a*a*a`,
`**a*a*b`, `**a*a*a+a`, `**a*a*b+b`, `**a*a**a*a`, `**a*a**a*b`,
`**a*a**b*a`, `**a*a**b*b`, `**a*b*a`, `**a*b*b`, `**a*b*a+a`,
`**a*b*b+b`, `**a*b**a*a`, `**a*b**a*b`, `**a*b**b*a`, `**a*b**b*b`,
`**b*a*a`, `**b*a*b`, `**b*a*a+a`, `**b*a*b+b`, `**b*a**a*a`,
`**b*a**a*b`, `**b*a**b*a`, `**b*a**b*b`, `**b*b*a`, `**b*b*b`,
`**b*b*a+a`, `**b*b*b+b`, `**b*b**a*a`, `**b*b**a*b`, `**b*b**b*a`,
`**b*b**b*b`$\}$ .

(b)

| | | |
|---|---|---|
| i. | `b` $\in \mathcal{M}$ | Eigenschaft m1 |
| ii. | Wegen `b` $\in \mathcal{M}$ (i) gilt `b+b` $\in \mathcal{M}$. | Eigenschaft m2 |
| iii. | `a` $\in \mathcal{M}$ | Eigenschaft m1 |
| iv. | Wegen `a` $\in \mathcal{M}$ (iii) und `b+b` $\in \mathcal{M}$ (ii) gilt `*a*b+b` $\in \mathcal{M}$. | Eigenschaft m3 |
| v. | Wegen `*a*b+b` $\in \mathcal{M}$ (iv) gilt `*a*b+b+*a*b+b` $\in \mathcal{M}$. | Eigenschaft m2 |
| vi. | Wegen `*a*b+b+*a*b+b` $\in \mathcal{M}$ (v) und `a` $\in \mathcal{M}$ (iii) gilt `**a*b+b+*a*b+b*a` $\in \mathcal{M}$. | Eigenschaft m3 |

Dieselbe Argumentation in Form eines Baumes:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathtt{a} \in \mathcal{M}}\,m1 \quad \dfrac{\dfrac{}{\mathtt{b} \in \mathcal{M}}\,m1}{\mathtt{b+b} \in \mathcal{M}}\,m2}{\mathtt{*a*b+b} \in \mathcal{M}}\,m3}{\mathtt{*a*b+b+*a*b+b} \in \mathcal{M}}\,m2 \quad \dfrac{}{\mathtt{a} \in \mathcal{M}}\,m1}{\mathtt{**a*b+b+*a*b+b*a} \in \mathcal{M}}\,m3$$

Die horizontalen Linien sind als wenn-dann zu lesen, wobei die Prämissen des Schlusses oberhalb und die Konklusion unterhalb der Linie angegeben werden. Neben dem Strich steht die angewendete Regel.

(c) Um zu zeigen, dass ein bestimmtes Wort nicht in der definierten Menge liegt, muss man eine Eigenschaft finden, die alle Wörter in der Menge besitzen, nicht aber das bestimmte Wort. Für die Wahl dieser Eigenschaft gibt es verschiedene Möglichkeiten; hier zwei davon.

*Argumentation über die Länge der Wörter.* Bei der stufenweisen Konstruktion der gegebenen Menge $\mathcal{M}$ besitzen jene Wörter in $\mathcal{M}_i$, die neu hinzukommen (die also noch nicht in $\mathcal{M}_{i-1}$ vorhanden sind), mindestens die Länge $3i$; das lässt sich mit einem kurzen Induktionsbeweis zeigen. Da das Wort `*b*a*b++*b*a*b` die Länge 13 besitzt, müsste es spätestens ab $i = 4$ in den stufenweise konstruierten Mengen $\mathcal{M}_i$ liegen. Wenn man also $\mathcal{M}_4$ konstruiert und feststellt, dass das gesuchte Wort nicht vorkommt, liegt es nicht in $\mathcal{M}$.

Diese Art von Argumentation lässt sich immer dann verwenden, wenn alle Abschlusseigenschaften der induktiven Definition zu längeren Wörtern führen. Allerdings kann es aufwändig sein, alle Wörter bis zur benötigten Länge tatsächlich zu generieren. Etwa besitzt $\mathcal{M}_4$ in diesem Beispiel bereits 30 830 258 Elemente.

*Argumentation über die `+`- und `*`-Struktur der Wörter.* Das Symbol `*` kommt in jedem Wort der Menge $\mathcal{M}$ in einer geraden Anzahl vorkommt, wie sich durch folgende induktive Überlegung zeigen lässt:

(m1) Die Wörter `a` und `b` enthalten keinen Stern, also eine gerade Anzahl.

(m2) Das Wort $u$`+`$u$ enthält doppelt so viele Sterne wie $u$, also eine gerade Anzahl.

(m3) Wenn $u$ und $v$ jeweils eine gerade Anzahl von Sternen enthalten, also $2i$ bzw. $2j$ Sterne wobei $i, j \geq 0$, dann enthält `*`$u$`*`$v$ auch eine gerade Anzahl, nämlich $1 + 2i + 1 + 2j = 2(1 + i + j)$.

Das Zeichen `+` kann nur durch die Eigenschaft m2 in ein Wort gelangen; in diesem Fall steht links und rechts davon dasselbe Wort $u \in \mathcal{M}$. Für $u$ gibt es in dieser Aufgabe nur zwei Möglichkeiten, nämlich `*b` oder `*b*a*b`, da alle anderen Zeichenfolgen vor `+` verschieden von den Zeichen danach sind. Jedes dieser beiden Wörter enthält aber eine ungerade Zahl von Sternen, es kann daher nicht aus $\mathcal{M}$ stammen.

Diese Argumentation ist kurz, allerdings benützt sie spezielle Eigenschaft der vorliegenden Menge $\mathcal{M}$ und ist nicht auf andere Fälle übertragbar; dort muss eine andere spezifische Eigenschaft gefunden werden.

## Aufgabe 5 (3 Punkte)

Sei $F$ die Formel $(((A \downarrow B) \wedge (B \downarrow C)) \supset \neg(A \vee (B \vee C)))$.

(a) Zeigen Sie, dass $F$ syntaktisch korrekt ist.

(b) Berechnen Sie schrittweise $\mathsf{val}_I(F)$ für $I(A) = 0$, $I(B) = 1$ und $I(C) = 0$.

(c) Verwenden Sie eine Wahrheitstafel um festzustellen, ob die Formel $F$ gültig, erfüllbar, widerlegbar und/oder unerfüllbar ist.

## Lösung

(a) Laut Vorlesung ist die Menge $\mathcal{A}$ der aussagenlogischen Formeln die kleinste Menge, für die gilt:

(a1) $\mathcal{V} \subseteq \mathcal{A}$

(a2) $\{\top, \bot\} \subseteq \mathcal{A}$

(a3) $\neg F \in \mathcal{A}$, wenn $F \in \mathcal{A}$.

(a4) $(F * G) \in \mathcal{A}$, wenn $F, G \in \mathcal{A}$ und $* \in \{\wedge, \uparrow, \vee, \downarrow, \equiv, \not\equiv, \supset, \subset\}$.

wobei $\mathcal{V} = \{A, B, C, \dots\}$ die Menge der aussagenlogischen Variablen ist.

Wir zeigen, dass $(((A \downarrow B) \wedge (B \downarrow C)) \supset \neg(A \vee (B \vee C)))$ eine aussagenlogische Formel gemäß dieser Definition ist.

(1) Die Variablen $A$, $B$ und $C$ sind Formeln (a1).

(2) Da $A$ und $B$ Formeln sind (Punkt 1), ist auch $(A \downarrow B)$ eine Formel (a4).

(3) Da $B$ und $C$ Formeln sind (Punkt 1), ist auch $(B \downarrow C)$ eine Formel (a4).

(4) Da $(A \downarrow B)$ und $(B \downarrow C)$ Formeln sind (Punkt 2 bzw. 3), ist auch $((A \downarrow B) \wedge (B \downarrow C))$ eine Formel (a4).

(5) Da $B$ und $C$ Formeln sind (Punkt 1), ist auch $(B \vee C)$ eine Formel (a4).

(6) Da $A$ und $(B \vee C)$ Formeln sind (Punkt 1 bzw. 5), ist auch $(A \vee (B \vee C))$ eine Formel (a4).

(7) Da $(A \vee (B \vee C))$ eine Formel ist (Punkt 6), ist auch $\neg(A \vee (B \vee C))$ eine solche (a3).

(8) Da $((A \downarrow B) \wedge (B \downarrow C))$ und $\neg(A \vee (B \vee C))$ Formeln sind (Punkt 4 bzw. 6), ist auch $(((A \downarrow B) \wedge (B \downarrow C)) \supset \neg(A \vee (B \vee C)))$ eine Formel (a4).

Dieselbe Argumentation in Form eines Baumes:

$$\dfrac{\dfrac{\dfrac{\dfrac{A \in \mathcal{V}}{A \in \mathcal{A}}\,a1 \quad \dfrac{B \in \mathcal{V}}{B \in \mathcal{A}}\,a1}{(A \downarrow B) \in \mathcal{A}}\,a4 \quad \dfrac{\dfrac{B \in \mathcal{V}}{B \in \mathcal{A}}\,a1 \quad \dfrac{c \in \mathcal{V}}{c \in \mathcal{A}}\,a1}{(B \downarrow C) \in \mathcal{A}}\,a4}{((A \downarrow B) \wedge (B \downarrow C)) \in \mathcal{A}}\,a4 \quad \dfrac{\dfrac{\dfrac{\dfrac{B \in \mathcal{V}}{B \in \mathcal{A}}\,a1 \quad \dfrac{C \in \mathcal{V}}{C \in \mathcal{A}}\,a1}{(B \vee C) \in \mathcal{A}}\,a4 \quad \dfrac{A \in \mathcal{V}}{A \in \mathcal{A}}\,a1}{(A \vee (B \vee C)) \in \mathcal{A}}\,a4}{\neg(A \vee (B \vee C)) \in \mathcal{A}}\,a3}{((((A \downarrow B) \wedge (B \downarrow C)) \supset \neg(A \vee (B \vee C)))) \in \mathcal{A}}\,a4$$

Die horizontalen Linien sind als wenn-dann zu lesen, wobei die Prämissen des Schlusses oberhalb und die Konklusion unterhalb der Linie angegeben werden. Neben dem Strich steht die angewendete Regel.

(b) $\mathsf{val}_I(((A \downarrow B) \wedge (B \downarrow C)) \supset \neg(A \vee (B \vee C)))$
$= \mathsf{val}_I((A \downarrow B) \wedge (B \downarrow C))$ implies $\mathsf{val}_I(\neg(A \vee (B \vee C)))$
$= (\mathsf{val}_I(A \downarrow B)$ and $\mathsf{val}_I(B \downarrow C))$ implies not $\mathsf{val}_I(A \vee (B \vee C))$
$= ((\mathsf{val}_I(A)$ nor $\mathsf{val}_I(B))$ and $(\mathsf{val}_I(B)$ nor $\mathsf{val}_I(C)))$ implies not$(\mathsf{val}_I(A)$ or $\mathsf{val}_I(B \vee C))$
$= ((0$ nor $1)$ and $(1$ nor $0))$ implies not$(0$ or $(\mathsf{val}_I(B)$ or $\mathsf{val}_I(C)))$
$= (0$ and $0)$ implies not$(0$ or $(1$ or $0))$
$= 0$ implies not$(0$ or $1)$
$= 0$ implies not $1$
$= 0$ implies $0$
$= 1$

(c) Wir berechnen den Wert der Formel für alle Interpretationen mittels einer Wahrheitstafel. An dieser lassen sich dann die Eigenschaften der Formel ablesen.

| $A$ | $B$ | $C$ | $(((A \downarrow$ | $B) \wedge$ | $(B \downarrow C))$ | $\supset$ | $\neg$ | $(A \vee (B \vee C)))$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | **1** | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 | **1** | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | **1** | 0 | 1 |
| 0 | 1 | 1 | 0 | 0 | 0 | **1** | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 | 1 | **1** | 0 | 1 |
| 1 | 0 | 1 | 0 | 0 | 0 | **1** | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 | 0 | **1** | 0 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | **1** | 0 | 1 |

Die Formel ist somit erfüllbar und gültig, aber weder widerlegbar noch unerfüllbar.

## Aufgabe 6 (3 Punkte)

Zeigen Sie, dass die beiden Formeln

$$((\neg A \equiv B) \wedge (B \wedge (A \vee C))) \quad \text{und} \quad ((\neg A \wedge (C \supset B)) \wedge (A \vee C))$$

äquivalent sind, und zwar

(a) mit Hilfe einer Wahrheitstafel.

(b) durch algebraische Umformungen.

Hinweis: Wie sich leicht überprüfen lässt, gelten die Äquivalenzen $(F \vee G) \wedge \neg G = F \wedge \neg G$ und $(F \vee \neg G) \wedge G = F \wedge G$ sowie die dualen Gleichungen.

## Lösung

(a) Wahrheitstafel:

| $A$ | $B$ | $C$ | $((\neg A{\equiv}B){\wedge}(B{\wedge}(A{\vee}C)))$ | $=$ | $((\neg A{\wedge}(C{\supset}B)){\wedge}(A{\vee}C))$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 0 0 0 **0** 0 0 0 0 0 | ✓ | 1 0 1 0 1 0 **0** 0 0 0 |
| 0 | 0 | 1 | 1 0 0 0 **0** 0 0 0 1 1 | ✓ | 1 0 0 1 0 0 **0** 0 1 1 |
| 0 | 1 | 0 | 1 0 1 1 **0** 1 0 0 0 0 | ✓ | 1 0 1 0 1 1 **0** 0 0 0 |
| 0 | 1 | 1 | 1 0 1 1 **1** 1 1 0 1 1 | ✓ | 1 0 1 1 1 1 **1** 0 1 1 |
| 1 | 0 | 0 | 0 1 1 0 **0** 0 0 1 1 0 | ✓ | 0 1 0 0 1 0 **0** 1 1 0 |
| 1 | 0 | 1 | 0 1 1 0 **0** 0 0 1 1 1 | ✓ | 0 1 0 1 0 0 **0** 1 1 1 |
| 1 | 1 | 0 | 0 1 0 1 **0** 1 1 1 1 0 | ✓ | 0 1 0 0 1 1 **0** 1 1 0 |
| 1 | 1 | 1 | 0 1 0 1 **0** 1 1 1 1 1 | ✓ | 0 1 0 1 1 1 **0** 1 1 1 |

Da beide Formeln in sämtlichen Wahrheitsbelegungen denselben Wert liefern, sind sie äquivalent.

Tatsächlich ist es gar nicht notwendig, die Wahrheitstabelle vollständig zu befüllen. Wir wissen zum Beispiel, dass $\wedge$ den Wert 0 liefert, wenn ein Argument den Wert 0 besitzt. Damit ist das Ergebnis der ersten Formel für jene Interpretationen, in denen $B$ den Wert 0 besitzt, bereits mit 0 festgelegt. Selbiges gilt für $\neg A$ in der zweiten Formel.

| $A$ | $B$ | $C$ | $((\neg A{\equiv}B){\wedge}(B{\wedge}(A{\vee}C)))$ | $=$ | $((\neg A{\wedge}(C{\supset}B)){\wedge}(A{\vee}C))$ |
|---|---|---|---|---|---|
| 0 | 0 | 0 | **0** 0 0 | ✓ | 1 0 1 0 1 0 **0** 0 0 0 |
| 0 | 0 | 1 | **0** 0 0 | ✓ | 1 0 0 1 0 0 **0** 0 1 1 |
| 0 | 1 | 0 | **0** 1 0 0 0 0 | ✓ | 1 0 1 0 1 1 **0** 0 0 0 |
| 0 | 1 | 1 | 1 0 1 1 **1** 1 1 0 1 1 | ✓ | 1 0 1 1 1 1 **1** 0 1 1 |
| 1 | 0 | 0 | **0** 0 0 | ✓ | 0 1 0 **0** |
| 1 | 0 | 1 | **0** 0 0 | ✓ | 0 1 0 **0** |
| 1 | 1 | 0 | 0 1 0 1 **0** 1 1 1 1 0 | ✓ | 0 1 0 **0** |
| 1 | 1 | 1 | 0 1 0 1 **0** 1 1 1 1 1 | ✓ | 0 1 0 **0** |

(b) Wir vereinfachen beide Formeln. Da wir dabei identische Formeln erhalten, sind die ursprünglichen Formeln äquivalent.

$$
\begin{aligned}
&(\neg A \equiv B) \wedge B \wedge (A \vee C) && F \equiv G = (\neg F \vee G) \wedge (F \vee \neg G)\\
&= (\neg\neg A \vee B) \wedge (\neg A \vee \neg B) \wedge B \wedge (A \vee C) && \neg\neg F = F\\
&= (A \vee B) \wedge (\neg A \vee \neg B) \wedge B \wedge (A \vee C) && (F \vee G) \wedge F = F\\
&= (\neg A \vee \neg B) \wedge B \wedge (A \vee C) && (F \vee \neg G) \wedge G = F \wedge G\\
&= \neg A \wedge B \wedge (A \vee C) && \neg F \wedge (F \vee G) \wedge G = \neg F \wedge G\\
&= \neg A \wedge B \wedge C
\end{aligned}
$$

$$
\begin{aligned}
&\neg A \wedge (C \supset B) \wedge (A \vee C) && F \supset G = \neg F \vee G\\
&= \neg A \wedge (\neg C \vee B) \wedge (A \vee C) && \neg F \wedge (F \vee G) = \neg F \wedge G\\
&= \neg A \wedge (\neg C \vee B) \wedge C && (\neg F \vee G) \wedge F = G \wedge F\\
&= \neg A \wedge B \wedge C
\end{aligned}
$$

## Aufgabe 7 (3 Punkte)

Ist die Formel $\neg(A \wedge (B \equiv C))$ eine logische Konsequenz der beiden Formeln $A \uparrow C$ und $B \supset (A \vee C)$? Wie sieht die Formel aus, deren Gültigkeit/Nichtgültigkeit zeigen würde, dass die Konsequenzbeziehung gilt/nicht gilt?

### Lösung

| $I(A)$ | $I(B)$ | $I(C)$ | $A \uparrow C,$ | $B \supset (A \vee C)$ | $\models_I$ | $\neg(A$ | $\wedge (B$ | $\equiv C))$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | ✓ | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 | ✓ | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 0 | ✓ | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | ✓ | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | $\lightning$ | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 | ✓ | 1 | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | ✓ | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 | 1 | ✓ | 0 | 1 | 1 |

Die Formel $\neg(A \wedge (B \equiv C))$ ist keine logische Konsequenz der beiden Prämissen. Ein Gegenbeispiel ist die Interpretation $I$ mit $I(A) = 1$, $I(B) = I(C) = 0$, in der die beiden Prämissen wahr sind, die Konklusion aber falsch.

*Arbeitsvereinfachung:* Ist in einer Interpretation eine der Prämissen falsch oder die Konklusion wahr, müssen die übrigen Formeln nicht mehr ausgewertet werden, da die Beziehung $\models_I$ dann bereits erfüllt ist. Umgekehrt kann man die Erstellung der Tabelle abbrechen, sobald man eine Interpretation $I$ findet, für die $\models_I$ nicht gilt.

Beginnt man in dieser Aufgabe die Auswertung mit der Konklusion, reduziert sich die weitere Analyse auf jene zwei Interpretationen, in denen $\neg(A \wedge (B \equiv C))$ falsch ist. Nur für diese Interpretationen müssen wir den Wert der Prämissen bestimmen. Bei der Auswertung der ersten dieser beiden Interpretationen erhält man lauter wahre Prämissen, somit kann abgebrochen werden.

| $I(A)$ | $I(B)$ | $I(C)$ | $A \uparrow C,$ | $B \supset (A \vee C)$ | $\models_I$ | $\neg(A$ | $\wedge (B$ | $\equiv C))$ |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | | | ✓ | 1 | 0 | 1 |
| 0 | 0 | 1 | | | ✓ | 1 | 0 | 0 |
| 0 | 1 | 0 | | | ✓ | 1 | 0 | 0 |
| 0 | 1 | 1 | | | ✓ | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | $\lightning$ | 0 | 1 | 1 |
| 1 | 0 | 1 | | | ✓ | 1 | 0 | 0 |
| 1 | 1 | 0 | | | ✓ | 1 | 0 | 0 |
| 1 | 1 | 1 | | | | 0 | 1 | 1 |

*Formel zur Konsequenzbeziehung:* $\neg(A \wedge (B \equiv C))$ ist genau dann eine logische Konsequenz der beiden Formeln $A \uparrow C$ und $B \supset (A \vee C)$, wenn die Formel

$$((A \uparrow C) \wedge (B \supset (A \vee C))) \supset \neg(A \wedge (B \equiv C))$$

gültig ist.

## Aufgabe 8 (2 Punkte)

Sei $f$ folgende dreistellige Funktion.

| $x$ | $y$ | $z$ | $f(x,y,z)$ |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 |

| $x$ | $y$ | $z$ | $f(x,y,z)$ |
|---|---|---|---|
| 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 0 |

Stellen Sie $f$ durch eine Formel in

(a) disjunktiver

(b) konjunktiver

Normalform dar.

### Lösung

(a) $(A_1 \wedge \neg A_2 \wedge \neg A_3) \vee (\neg A_1 \wedge A_2 \wedge A_3) \vee (\neg A_1 \wedge \neg A_2 \wedge A_3)$

(b) $(\neg A_1 \vee \neg A_2 \vee \neg A_3) \wedge (\neg A_1 \vee \neg A_2 \vee A_3) \wedge (\neg A_1 \vee A_2 \vee \neg A_3) \wedge (A_1 \vee \neg A_2 \vee A_3) \wedge (A_1 \vee A_2 \vee A_3)$

## Aufgabe 9 (2 Punkte)

Sei $F$ die Formel $(C \supset (A \vee B)) \wedge (A \equiv (B \downarrow C))$.

(a) Bestimmen Sie eine zu $F$ äquivalente Formel in disjunktiver Normalform. Verwenden Sie die semantische Methode.

(b) Bestimmen Sie eine zu $F$ äquivalente Formel in konjunktiver Normalform. Verwenden Sie die algebraische Methode.

### Lösung

(a) DNF mittels semantischer Methode:

| $A$ | $B$ | $C$ | $(C \supset$ | $(A \vee B))$ | $\wedge$ | $(A \equiv$ | $(B \downarrow C))$ |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 0 | **0** | 0 | 1 |
| 0 | 0 | 1 | 0 | 0 | **0** | 1 | 0 |
| 0 | 1 | 0 | 1 | 1 | **1** | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | **1** | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | **1** | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | **0** | 0 | 0 |
| 1 | 1 | 0 | 1 | 1 | **0** | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | **0** | 0 | 0 |

Aus dieser Tafel lässt sich folgende DNF ablesen:

$$(\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge C) \vee (A \wedge \neg B \wedge \neg C)$$

(b) KNF mittels algebraischer Methode:

$$\begin{array}{ll}
(C \supset (A \vee B)) \wedge (A \equiv (B \downarrow C)) & F \equiv G = (\neg F \vee G) \wedge (F \vee \neg G) \\
= (C \supset (A \vee B)) \wedge ((\neg A \vee (B \downarrow C)) \wedge (A \vee \neg(B \downarrow C))) & (F \downarrow G) = \neg F \wedge \neg G \\
= (C \supset (A \vee B)) \wedge ((\neg A \vee (\neg B \wedge \neg C)) \wedge (A \vee \neg(\neg B \wedge \neg C))) & \neg(F \wedge G) = \neg F \vee \neg G \\
= (C \supset (A \vee B)) \wedge ((\neg A \vee (\neg B \wedge \neg C)) \wedge (A \vee (\neg\neg B \vee \neg\neg C))) & \neg\neg F = F \\
= (C \supset (A \vee B)) \wedge ((\neg A \vee (\neg B \wedge \neg C)) \wedge (A \vee B \vee C)) & F \vee (G \wedge H) = (F \vee G) \wedge (F \vee H) \\
= (C \supset (A \vee B)) \wedge (\neg A \vee \neg B) \wedge (\neg A \vee \neg C) \wedge (A \vee B \vee C) & F \supset G = \neg F \vee G \\
= (\neg C \vee A \vee B) \wedge (\neg A \vee \neg B) \wedge (\neg A \vee \neg C) \wedge (A \vee B \vee C) & \\
= (A \vee B \vee \neg C) \wedge (\neg A \vee \neg B) \wedge (\neg A \vee \neg C) \wedge (A \vee B \vee C) &
\end{array}$$

## Aufgabe 10 (4 Punkte)

Thomas wird im Sommer in das Ferienlager am Wolfgangsee fahren. Um den Organisatoren die Planung zu erleichtern, soll er davor bekannt geben, welche Sportarten (maximal drei) er dort ausüben möchte. Thomas kann sich nur schwer entscheiden und schreibt folgende Email.

> *Liebe Organisatoren,*
> *ich freue mich schon sehr auf den Sommer und das Camp!*
> *Fußball und Handball sind beide ok, ich möchte aber auf keinen Fall beide machen, das wäre mir zu anstrengend. Mein Freund Florian kommt auch ins Camp und geht Bogenschießen, das möchte ich auf jeden Fall auch machen! Falls ich beim Rudern teilnehme, möchte ich auch segeln, damit sich der Weg zum See auszahlt; Segeln ohne Rudern kann ich mir aber schon vorstellen, das ist alleine auch spannend genug. Ich möchte gerne meine Oberarme trainieren, dafür sind wohl Rudern oder Handball ganz gut geeignet; auch beide gleichzeitig wären in Ordnung.*
> *Bis bald, Thomas*

(a) Helfen Sie den Organisatoren die Email zu analysieren, indem Sie die beschriebenen Wünsche mit allen Anhaltspunkte durch aussagenlogische Formeln ausdrücken. Geben Sie die Bedeutung der von Ihnen verwendeten Aussagenvariablen an.

(b) Lassen sich Thomas' Wünsche berücksichtigen? Welche Varianten sind möglich? Begründen Sie die Antwort mit Hilfe Ihrer aussagenlogischen Modellierung.

## Lösung

(a) Aussagenvariablen und ihre Bedeutung:

$H$ ... Thomas nimmt Handball.
$F$ ... Thomas nimmt Fußball.
$B$ ... Thomas nimmt Bogenschießen.
$R$ ... Thomas nimmt Rudern.
$S$ ... Thomas nimmt Segeln.

Aussagenlogische Formeln:

$F_1 := (\neg H \wedge \neg F) \vee (\neg H \wedge \neg B) \vee \cdots \vee (\neg B \wedge \neg S) \vee (\neg R \wedge \neg S)$
max. drei Sportarten (10 Disjunkte)

oder

$F_1 := (\neg H \wedge \neg F) \vee (\neg H \wedge \neg R) \vee (\neg H \wedge \neg S) \vee (\neg F \wedge \neg R) \vee (\neg F \wedge \neg S) \vee (\neg R \wedge \neg S)$
max. zwei weitere Sportarten zusätzlich zu $B$

$F_2 := F \uparrow H$ Fußball und Handball nicht gleichzeitig
$F_3 := B$ Bogenschießen auf jeden Fall
$F_4 := R \supset S$ wenn Rudern, dann auch Segeln
$F_5 := R \vee H$ Rudern oder Handball oder beides

(b) Wir suchen alle Wahrheitsbelegungen für die Variablen $H$, $F$, $B$, $R$ und $S$, sodass die Formeln $F_1, \ldots, F_5$ wahr werden. Wegen der Formel $F_1$ genügt es Belegungen zu betrachten, in denen höchstens drei Variablen wahr sind; wegen Formel $F_3$ interessieren nur Belegungen, in denen $B$ wahr ist.

| $B$ | $F$ | $H$ | $R$ | $S$ | $F_1$ | $F \uparrow H$ | $B$ | $R \supset S$ | $R \vee H$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | |
| 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | ✓ |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | ✓ |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | ✓ |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | |
| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | |

Es gibt drei Sportkombinationen, die die Wünsche von Thomas erfüllen: Er nimmt auf jeden Fall Bogenschießen und zusätzlich entweder Handball, oder Handball und Segeln, oder Rudern und Segeln.

## Aufgabe 11 (4 Punkte)

Dienstag, 13:00 Uhr. Es ist wieder Zeit für die Vorlesung aus Formale Modellierung. Doch als eine Gruppe Studierender pünktlich zu Vorlesungsbeginn das AudiMax betritt, findet sie anstelle des Professors eine Bombe mit fünf Kippschaltern sowie einen Zettel mit folgender Nachricht.

> Liebe Studierende,
>
> um euch zur Aussagenlogik zu motivieren, habe ich diese Bombe gebaut. Sie hat fünf Kippschalter (A, B, C, D und E), die jeweils in der Position „oben“ bzw. „unten“ stehen können. Die fünf Kippschalter müssen um 14:00 alle an der richtigen Position stehen, sonst explodiert die Bombe und ihr werdet alle mit dem wohlriechenden Schleim überzogen, den ich in die Bombe gefüllt habe. Hier sind eure Hinweise.
>
> - Mindestens drei Schalter müssen oben sein, sonst knallt’s.
> - B darf nicht in der gleichen Position sein wie C, sonst knallt’s.
> - Wenn D oben ist, müssen C und E unten sein, sonst knallt’s.
> - A muss oben sein, sonst … eh schon wissen.
> - Zumindest C oder D müssen oben sein, vielleicht sogar beide, sonst …

(a) Formalisieren Sie die beschriebene Situation inklusive aller Anhaltspunkte mittels aussagenlogischer Formeln. Geben Sie die Bedeutung der von Ihnen verwendeten Aussagenvariablen an.

(b) Wie müssen die Kippschalter stehen? Ist es möglich, die Bombe zu entschärfen? Gibt es eine Lösung oder mehrere? Welche? Begründen Sie die Antwort mit Hilfe Ihrer aussagenlogischen Modellierung.

### Lösung

(a) Aussagenvariablen und ihre Bedeutung:

$A$ … Schalter A ist oben.
$B$ … Schalter B ist oben.
$C$ … Schalter C ist oben.
$D$ … Schalter D ist oben.
$E$ … Schalter E ist oben.

Aussagenlogische Formeln:

| | |
|---|---|
| $F_0 := (A \wedge B \wedge C) \vee (A \wedge B \wedge D)$ $\vee(A \wedge B \wedge E) \vee (A \wedge C \wedge D)$ $\vee(A \wedge C \wedge E) \vee (A \wedge D \wedge E)$ $\vee(B \wedge C \wedge D) \vee (B \wedge C \wedge E)$ $\vee(B \wedge D \wedge E) \vee (C \wedge D \wedge E)$ | Mind. drei Schalter oben |
| $F_1 := B \not\equiv C$ | B nicht gleiche Pos. wie C |
| $F_2 := D \supset (\neg C \wedge \neg E)$ | wenn D oben, dann C und E unten |
| $F_3 := A$ | A auf jeden Fall oben |
| $F_4 := C \vee D$ | C oder D oben |

(b) Wir suchen alle Wahrheitsbelegungen für die Variablen $A$, $B$, $C$, $D$ und $E$, sodass die Formeln $F_0, \ldots, F_4$ wahr werden. Wegen Formel $F_3$ genügt es jene Belegungen zu betrachten, in denen $A$ wahr ist.

| $A$ | $B$ | $C$ | $D$ | $E$ | $F_0$ | $B \not\equiv C$ | $D \supset (\neg C \wedge \neg E)$ | $C \vee D$ | |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | | | | |
| 1 | 0 | 0 | 0 | 1 | 0 | | | | |
| 1 | 0 | 0 | 1 | 0 | 0 | | | | |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | | | |
| 1 | 0 | 1 | 0 | 0 | 0 | | | | |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | ✓ |
| 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | | |
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | | |
| 1 | 1 | 0 | 0 | 0 | 0 | | | | |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | |
| 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | ✓ |
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | | |
| 1 | 1 | 1 | 0 | 0 | 1 | 0 | | | |
| 1 | 1 | 1 | 0 | 1 | 1 | 0 | | | |
| 1 | 1 | 1 | 1 | 0 | 1 | 0 | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | | | |

Es gibt zwei Möglichkeiten, die Bombe zu entschärfen.

- Schalter A, C und E sind oben, der Rest unten
- Schalter A, B und D sind oben, der Rest unten

## Aufgabe 12 (3 Punkte)

SAT-Solver sind Programme, die als Eingabe eine *einzige* Formel in *konjunktiver Normalform* verarbeiten und diese auf Erfüllbarkeit testen. Als Ausgabe liefern sie die Information „erfüllbar„ bzw. „unerfüllbar“. Im ersten Fall wird eine erfüllende Variablenbelegung als Nachweis für die Erfüllbarkeit ausgegeben.
Wie lässt sich ein SAT-Solver verwenden um zu festzustellen, ob eine Formel $F$ von zwei Formeln $G$ und $H$ subsumiert wird? Man sagt, dass eine Formel $F$ von Formeln

$G_1$, …, $G_n$ subsumiert wird, wenn in jeder Variablenbelegung, in der die Formeln $G_1$, …, $G_n$ wahr sind, auch die Formel $F$ wahr ist. Intuitiv gesprochen steuert Formel $F$ in diesem Fall keine neue Information bei. Beschreiben Sie alle erforderlichen Schritte, von der Aufbereitung der Fragestellung für den SAT-Solver bis zur Interpretation des Ergebnisses des SAT-Solvers. Was bedeutet es in Ihrer Methode, wenn der SAT-Solver eine erfüllende Variablenbelegung findet?

### Lösung

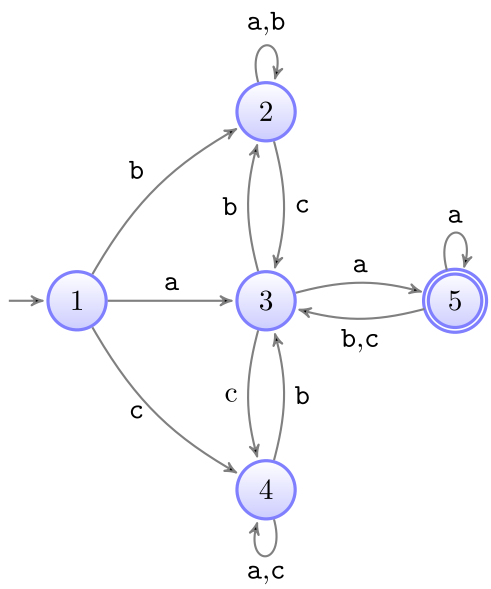
Es gilt:

> $F$ wird von $G$ und $H$ subsumiert
> genau dann, wenn die logische Kosequenz $G, H \models F$ gilt;
> genau dann, wenn die Formel $(G \wedge H) \supset F$ gültig ist;
> genau dann, wenn die Formel $\neg((G \wedge H) \supset F)$ unerfüllbar ist;
> genau dann, wenn die Formel $G \wedge H \wedge \neg F$ unerfüllbar ist.

Um also festzustellen, ob $F$ von $G$ und $H$ subsumiert wird, wandeln wir die Formel $G \wedge H \wedge \neg F$ in konjunktive Normalform um und übergeben sie dem SAT-Solver. Liefert dieser „unerfüllbar“, wird $F$ von $G$ und $H$ subsumiert. Liefert der SAT-Solver hingegen „erfüllbar“, wird $F$ nicht subsumiert; die vom SAT-Solver gelieferte erfüllende Wahrheitsbelegung stellt ein Gegenbeispiel dar: in dieser Interpretation sind die Formeln $G$ und $H$ wahr, $F$ aber falsch.

## Aufgabe 13 (4 Punkte)

Sei $\mathcal{A}$ der folgende endliche Automat.



(a) Geben Sie 5 Wörter an, die von $\mathcal{A}$ akzeptiert werden.

(b) Geben Sie an, welche der folgenden Wörter der Automat akzeptiert: $\varepsilon$, `aa`, `aaa`, `b`, `baca`.

(c) Berechnen Sie schrittweise $\delta^*(1, \texttt{babca})$.

(d) Spezifizieren Sie $\mathcal{A}$ in tabellarischer Form. Handelt es sich bei $\mathcal{A}$ um einen determinstischen oder indeterministischen Automaten?

### Lösung

(a) $\mathcal{A}$ akzeptiert zum Beispiel `aa`, `aaa`, `cba`, `caba` und `cbaba`.

(b) $\mathcal{A}$ akzeptiert `aa`, `aaa` und `baca`, nicht aber $\varepsilon$ und `b`.

(c)
$$\begin{aligned}
\delta^*(1, \texttt{babca}) &= \delta^*(\delta(1, \texttt{b}), \texttt{abca}) \\
&= \delta^*(2, \texttt{abca}) \\
&= \delta^*(\delta(2, \texttt{a}), \texttt{bca}) \\
&= \delta^*(2, \texttt{bca}) \\
&= \delta^*(\delta(2, \texttt{b}), \texttt{ca}) \\
&= \delta^*(2, \texttt{ca}) \\
&= \delta^*(\delta(2, \texttt{c}), \texttt{a}) \\
&= \delta^*(3, \texttt{a}) \\
&= \delta^*(\delta(3, \texttt{a}), \varepsilon) \\
&= \delta^*(5, \varepsilon) \\
&= 5
\end{aligned}$$

(d) $\mathcal{A} = \langle\{1,2,3,4,5\},\ \{\texttt{a},\texttt{b},\texttt{c}\},\ \delta,\ 1,\ \{5\}\rangle$, wobei die Übergangsfunktion $\delta$ durch folgende Tabelle definiert ist:

| $\delta$ | a | b | c |
|---|---|---|---|
| 1 | 3 | 2 | 4 |
| 2 | 2 | 2 | 3 |
| 3 | 5 | 2 | 4 |
| 4 | 4 | 3 | 4 |
| 5 | 5 | 3 | 3 |

$\mathcal{A}$ ist ein deterministischer Automat, da der momentane Zustand und die nächste Eingabe immer eindeutig den Folgezustand bestimmen. Das äußert sich in der Tabelle dadurch, dass jeder Eintrag genau einen Zustand enthält.

## Aufgabe 14 (4 Punkte)

Geben Sie einen endlichen Automaten an, der genau jene JAVA-Oktalnumerale akzeptiert, deren Wert durch fünf teilbar ist. Beispielsweise sollen die Zeichenketten `0`, `05` und `012` akzeptiert werden, da sie den durch 5 teilbaren Zahlen 0, 5 und 10 entsprechen, nicht aber die Zeichenfolgen `5` (kein JAVA-Oktalnumeral) und `010` (8 ist nicht durch 5 teilbar).

### Lösung

Oktalliterale in JAVA sind Zeichenketten über dem Alphabet $\Sigma = \{\texttt{0}, \texttt{1}, \ldots, \texttt{7}\}$, die mit dem Zeichen `0` beginnen. Es sollen jene Oktalliterale akzeptiert werden, die einen durch fünf teilbaren Wert besitzen. Für die Konstruktion des Automaten sind zwei Beobachtungen nützlich.

- Kennt man den Wert $W$ eines Numerals $N$, dann lässt sich der Wert eines Numerals $Nn$, das durch Anhängen einer weiteren Oktalziffern $n$ mit dem Wert $w$ entsteht, berechnen als $W \cdot 8 + w$. Beispielsweise hat das Numeral $N = \texttt{012}$ den Wert $W = 10$; hängt man die Oktalziffer $n = \texttt{3}$ mit dem Wert $w = 3$ daran, erhält man den Wert des Numerals $Nn = \texttt{0123}$ durch $W \cdot 8 + w = 10 \cdot 8 + 3 = 83$.

- Restklassenarithmetik: Um die Restklasse einer Summe aus zwei Zahlen zu bestimmen, genügt es, die Restklassen der Summanden zu addieren. Dasselbe gilt für die Multiplikation. Um also die Restklasse von $W \cdot 8 + w$ modulo 5 zu bestimmen, genügt es, die Restklassen von $W$, 8 und $w$ zu betrachten. Im obigen Beispiel liegt $W = 10$ in der Restklasse 0 modulo 5 (da durch 5 teilbar), 8 und 3 liegen beide in der Restklasse 3. Das Ergebnis von $10 \cdot 8 + 3$ liegt also in derselben Restklasse wie $0 \cdot 3 + 3 = 3$; somit wissen wir, dass 83 in der Restklasse drei liegt.

  Die Relevanz dieser Beobachtung liegt darin, dass der Automat ja feststellen soll, ob der Wert des Numerals durch 5 teilbar, also in der Restklasse 0 liegt. Die Gesetze der Restklassenarithmetik erlauben es uns, bei der Überprüfung dieser Eigenschaft im Zahlenraum $0, \ldots, 4$ (den Restklassen modulo 5) zu bleiben.

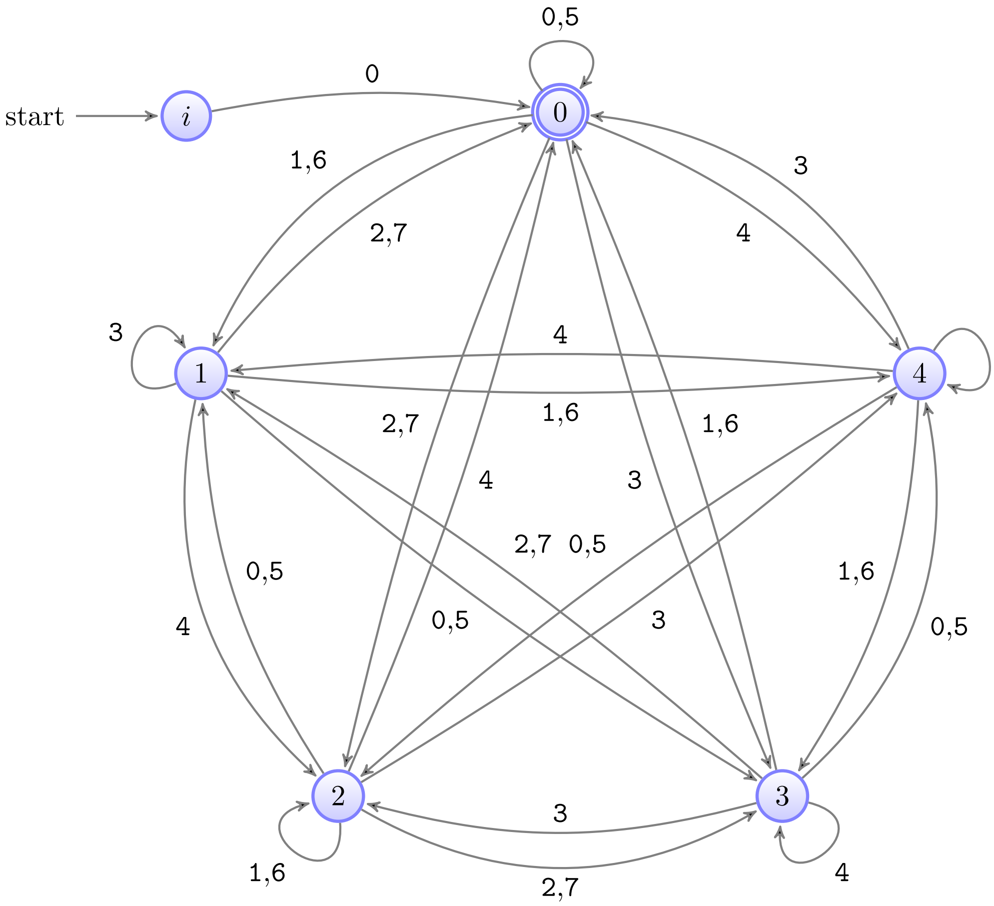
Der Automat besitzt also für jede Restklasse einen Zustand, $0, \ldots, 4$. Befindet sich der Automat im Zustand $r$, so heißt das, dass das bisher verarbeitete Numeral in der Restklasse $r$ modulo 5 liegt. Endzustand ist der Zustand 0, da Numerale genau dann akzeptiert werden sollen, wenn ihr Wert durch 5 teilbar ist. Da Numerale aus mindestens einem Zeichen bestehen müssen, benötigen wir noch einen eigenen Anfangszustand $i$. Numerale müssen mit `0` beginnen, daher benötigen wir für die übrigen Symbole einen Fehlerzustand $f$. Wir erhalten den Automaten $\langle Q, \Sigma, \delta, i, \{0\} \rangle$, wobei

$$
\begin{aligned}
Q &= \{i, 0, 1, 2, 3, 4, f\}, \\
\Sigma &= \{\texttt{0}, \texttt{1}, \texttt{2}, \texttt{3}, \texttt{4}, \texttt{5}, \texttt{6}, \texttt{7}\}
\end{aligned}
$$

und die Übergangsfunktion $\delta$ definiert ist durch folgende Tabelle:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $i$ | 0 | $f$ | $f$ | $f$ | $f$ | $f$ | $f$ | $f$ |
| 0 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 |
| 1 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 |
| 2 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 |
| 3 | 4 | 0 | 1 | 2 | 3 | 4 | 0 | 1 |
| 4 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 |
| $f$ | $f$ | $f$ | $f$ | $f$ | $f$ | $f$ | $f$ | $f$ |

Graphisch lässt sich der Automat folgendermaßen darstellen:




## Aufgabe 15 (4 Punkte)

Ein elektronisches Tresorschloss besteht aus einem dreistelligen Display sowie den Tasten `1`, `0`, `Next`, `Ok` und `Reset`. Jede Stelle des Displays kann 0 oder 1 anzeigen. Ein- oder mehrmaliges Drücken der `0`- bzw. `1`-Taste setzt die aktive Stelle auf 0 bzw. 1. Welche der drei Stellen aktiv ist, lässt sich durch die `Next`-Taste kontrollieren: Mit jedem Druck dieser Taste wandert die Aktivierung um eine Stelle weiter nach rechts bzw. springt von der dritten Stelle zurück zur ersten. Im Anfangszustand zeigt das Display die Zahl 000 an und die erste Stelle (ganz links) ist aktiviert. Wird die Zahl 101 eingestellt und anschließend die `Ok`-Taste gedrückt, öffnet das Schloss; bei jeder anderen Zahl geht das Schloss in einen Fehlerzustand. Sowohl im geöffneten Zustand als auch im Fehlerzustand werden alle weiteren Tasten ausgenommen `Reset` ignoriert, d.h., sie beeinflussen den Zustand des Schlosses nicht. Wird zu einem beliebigen Zeitpunkt die `Reset`-Taste gedrückt, geht das Schloss wieder in den Anfangszustand über. Das Schloss lässt sich also z.B. mit folgenden Tastenfolgen öffnen:

`1` – `Next` – `Next` – `1` – `Ok`
`1` – `Reset` – `Next` – `0` – `Next` – `1` – `1` – `Next` – `1` – `Ok` – `Ok`

- Überlegen Sie, welche Informationen notwendig sind, um den Zustand des Schlosses zu beschreiben. Wieviele Zustände kann das Schloss annehmen? Wieviele Zustände sind es im Allgemeinen, wenn das Schloss $n$ Ziffern (statt 2) pro Stelle sowie $k$ Stellen (statt 3) besitzt?
- Legen Sie die möglichen Aktionen fest, die zu einem Zustandswechsel führen.
- Geben Sie einen endlichen Automaten an, der das Verhalten des Schlosses vollständig beschreibt. Der Endzustand ist erreicht, wenn das Schloss öffnet. Spezifizieren Sie die Übergangsfunktion des Automaten in tabellarischer oder graphischer Form.

## Lösung

Der Zustand des Schlosses wird durch die angezeigten Ziffern sowie durch die Position der Aktivierung eindeutig festgelegt. Daher besitzt das Schloss $2^3 \cdot 3 + 2 = 26$ Zustände, im Allgemeinen sind es $n^k \cdot k + 2$ Zustände. (Die beiden Extrazustände sind der Fehlerzustand und das geöffnete Schloss.) Die Aktionen, die zu Zustandswechseln führen (können), sind die möglichen Tastendrücke, also `1`, `0`, `Next`, `Ok` und `Reset`. Als Zustandsbezeichnung wählen wir $\underline{a}bc$, $a\underline{b}c$ bzw. $ab\underline{c}$, wobei $abc$ den angezeigten Ziffern entspricht und die Unterstreichung die aktivierte Stelle markiert.
Das Verhalten des Schlosses wird durch den Automaten

$$\langle\{\underline{0}00, \ldots, 11\underline{1}, \mathit{Fehler}, \mathit{Offen}\}, \{\mathtt{1}, \mathtt{0}, \mathtt{Next}, \mathtt{Ok}, \mathtt{Reset}\}, \delta, \underline{0}00, \{\mathit{Offen}\}\rangle$$

beschrieben, wobei die Übergangsfunktion $\delta$ durch folgende Tabelle definiert wird.

| $\delta$ | `1` | `0` | `Next` | `Ok` | `Reset` |
|---|---|---|---|---|---|
| $\underline{0}00$ | $\underline{1}00$ | $\underline{0}00$ | $0\underline{0}0$ | *Fehler* | $\underline{0}00$ |
| $\underline{1}00$ | $\underline{1}00$ | $\underline{0}00$ | $1\underline{0}0$ | *Fehler* | $\underline{0}00$ |
| $\underline{0}10$ | $\underline{1}10$ | $\underline{0}10$ | $0\underline{1}0$ | *Fehler* | $\underline{0}00$ |
| $\underline{1}10$ | $\underline{1}10$ | $\underline{0}10$ | $1\underline{1}0$ | *Fehler* | $\underline{0}00$ |
| $\underline{0}01$ | $\underline{1}01$ | $\underline{0}01$ | $0\underline{0}1$ | *Fehler* | $\underline{0}00$ |
| $\underline{1}01$ | $\underline{1}01$ | $\underline{0}01$ | $1\underline{0}1$ | *Offen* | $\underline{0}00$ |
| $\underline{0}11$ | $\underline{1}11$ | $\underline{0}11$ | $0\underline{1}1$ | *Fehler* | $\underline{0}00$ |
| $\underline{1}11$ | $\underline{1}11$ | $\underline{0}11$ | $1\underline{1}1$ | *Fehler* | $\underline{0}00$ |
| $0\underline{0}0$ | $0\underline{1}0$ | $0\underline{0}0$ | $00\underline{0}$ | *Fehler* | $\underline{0}00$ |
| $1\underline{0}0$ | $1\underline{1}0$ | $1\underline{0}0$ | $10\underline{0}$ | *Fehler* | $\underline{0}00$ |
| $0\underline{1}0$ | $0\underline{1}0$ | $0\underline{0}0$ | $01\underline{0}$ | *Fehler* | $\underline{0}00$ |
| $1\underline{1}0$ | $1\underline{1}0$ | $1\underline{0}0$ | $11\underline{0}$ | *Fehler* | $\underline{0}00$ |
| $0\underline{0}1$ | $0\underline{1}1$ | $0\underline{0}1$ | $00\underline{1}$ | *Fehler* | $\underline{0}00$ |
| $1\underline{0}1$ | $1\underline{1}1$ | $1\underline{0}1$ | $10\underline{1}$ | *Offen* | $\underline{0}00$ |
| $0\underline{1}1$ | $0\underline{1}1$ | $0\underline{0}1$ | $01\underline{1}$ | *Fehler* | $\underline{0}00$ |
| $1\underline{1}1$ | $1\underline{1}1$ | $1\underline{0}1$ | $11\underline{1}$ | *Fehler* | $\underline{0}00$ |
| $00\underline{0}$ | $00\underline{1}$ | $00\underline{0}$ | $\underline{0}00$ | *Fehler* | $\underline{0}00$ |
| $10\underline{0}$ | $10\underline{1}$ | $10\underline{0}$ | $\underline{1}00$ | *Fehler* | $\underline{0}00$ |
| $01\underline{0}$ | $01\underline{1}$ | $01\underline{0}$ | $\underline{0}10$ | *Fehler* | $\underline{0}00$ |
| $11\underline{0}$ | $11\underline{1}$ | $11\underline{0}$ | $\underline{1}10$ | *Fehler* | $\underline{0}00$ |
| $00\underline{1}$ | $00\underline{1}$ | $00\underline{0}$ | $\underline{0}01$ | *Fehler* | $\underline{0}00$ |
| $10\underline{1}$ | $10\underline{1}$ | $10\underline{0}$ | $\underline{1}01$ | *Offen* | $\underline{0}00$ |
| $01\underline{1}$ | $01\underline{1}$ | $01\underline{0}$ | $\underline{0}11$ | *Fehler* | $\underline{0}00$ |
| $11\underline{1}$ | $11\underline{1}$ | $11\underline{0}$ | $\underline{1}11$ | *Fehler* | $\underline{0}00$ |
| *Fehler* | *Fehler* | *Fehler* | *Fehler* | *Fehler* | $\underline{0}00$ |
| *Offen* | *Offen* | *Offen* | *Offen* | *Offen* | $\underline{0}00$ |