1. Übungsblatt (mit Lösungen)

3.0 VU Formale Modellierung

Gerald Berger, Gernot Salzer

29. Oktober 2015

Aufgabe 1 (0.3 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) Alle Säugetiere sind sterblich. Manche Tiere sind Säugetiere. Daher sind manche Tiere sterblich.

(b) Kein Rechteck ist ein Kreis. Alle Quadrate sind Rechtecke. Daher ist kein Quadrat ein Kreis.

(c) Alle Fische atmen durch Kiemen. Kein Säugetier ist ein Fisch. Daher atmet kein Säugetier durch Kiemen.

Lösung

(a)
$$\begin{array}{l} \text{Säugetiere sind sterblich.} \\ \text{Manche Tiere sind Säugetiere.} \\ \hline \text{Manche Tiere sind sterblich.} \end{array} \qquad \text{Inferenzregel: } \begin{array}{l} \text{Alle } x \text{ sind } y. \\ \text{Manche } z \text{ sind } x. \\ \hline \text{Manche } z \text{ sind } y. \end{array}$$

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

$$\begin{array}{l} \text{Alle manipulierten Autos sind zurückzurufen.} \\ \text{Manche VW-Autos sind manipuliert.} \\ \hline \text{Manche VW-Autos sind zurückzurufen.} \end{array}$$

(b)
$$\begin{array}{l} \text{Kein Rechteck ist ein Kreis.} \\ \text{Alle Quadrate sind Rechtecke.} \\ \hline \text{Kein Quadrat ist ein Kreis.} \end{array} \qquad \text{Inferenzregel: } \begin{array}{l} \text{Kein } x \text{ ist ein } y. \\ \text{Alle } z \text{ sind } x. \\ \hline \text{Kein } z \text{ ist ein } y. \end{array}$$

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

Kein Säugetier schlüpft aus dem Ei.
Alle Katzen sind Säugetiere.

Keine Katze schlüpft aus dem Ei.

(c) Alle Fische atmen durch Kiemen.
Kein Säugetier ist ein Fisch.

Kein Säugetier atmet durch Kiemen.

Inferenzregel: Alle x haben Eigenschaft y.
Kein z ist ein x.

Kein z hat Eigenschaft y.

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

Alle x haben Eigenschaft y.
Kein z hat Eigenschaft y.

Kein z ist ein x.

Ein anderes Beispiel, dem dieselbe Inferenzregel zugrunde liegt:

Alle Professorinnen sind habilitiert.
Keine Assistentin ist habilitiert.

Keine Assistentin ist eine Professorin.

Aufgabe 2 (0.4 Punkte)

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

(a) Die Erdbeeren sind süß.

(b) Ich gehe ins Kino oder ich bleibe daheim.

(c) Wenn ich mich nicht beeile, werde ich die Vorlesung versäumen.

(d) Nur wenn ich jetzt losfahre, komme ich rechtzeitig zum Flughafen.

(e) Entweder fahre ich im Juli oder im August auf Urlaub. Beides geht sich zeitlich nicht aus.

(f) Ich koche heute nicht, lasse mir jedoch eine Pizza liefern.

(g) Wenn der Bus nicht rechtzeitig kommt, so werde ich nicht pünktlich sein.

(h) Ich putze nur dann die Fenster, wenn es nicht regnet.

Lösung

(a) *Die Erdbeeren sind süß.*
A ... Die Erdbeeren sind süß.
Struktur: A.
Formel: A

(b) *Ich gehe ins Kino oder ich bleibe daheim.*
A ... Ich gehe ins Kino.
B ... Ich bleibe daheim.
Struktur: A oder B.
Formel: $A \vee B$

Sofern Heim und Kino nicht identisch sind, schließen sich die beiden Aufenthaltsorte aus. Das führt zur stärkeren Formalisierung $A \not\equiv B$.

(c) *Wenn ich mich nicht beeile, werde ich die Vorlesung versäumen.*
A ... Ich beeile mich.
B ... Ich versäume die Vorlesung.
Struktur: Wenn nicht A, dann B.
Formel: $\neg A \supset B$

Man kann dieses „wenn – dann“ auch als Äquivalenz auffassen. Erstens wird in der Alltagssprache generell wenn-dann meist als Äquivalenz verstanden. Zweitens geht man implizit davon aus, dass, wenn man sich beeilt, es so tut, dass man sein Ziel auch erreicht (hier also die Vorlesung nicht versäumt); es schwingt also auch die andere Richtung der Implikation mit: Wenn ich mich beeile, werde ich die Vorlesung nicht versäumen.
Struktur: Nicht A genau dann, wenn B.
Formel: $\neg A \equiv B$

(d) *Nur wenn ich jetzt losfahre, komme ich rechtzeitig zum Flughafen.*
A ... Ich fahre jetzt los.
B ... Ich komme rechtzeitig zum Flughafen.
Struktur: Nur wenn A, dann B.
Formel: $A \subset B$

Beachten Sie, dass sich durch das Wort „nur“ die Richtung der Implikation umdreht. Die Aussage ist äquivalent zu den folgenden Umformulierungen:

Ich bin jetzt losgefahren, falls/wenn ich rechtzeitig zum Flughafen gekommen sein werde.

Wenn ich jetzt nicht losfahre, komme ich nicht rechtzeitig zum Flughafen.

Die Formulierung „nur wenn – dann“ ist ziemlich eindeutig als Implikation und nicht als Äquivalenz zu verstehen. Die Abfahrt zum jetzigen Zeitpunkt ist notwendig, aber nicht hinreichend: Auch ein Stau auf der Autobahn oder ein Ausfall des CAT kann die rechtzeitige Ankunft verhindern.

(e) *Entweder fahre ich im Juli oder im August auf Urlaub. Beides geht sich zeitlich nicht aus.*
A ... Ich fahre im Juli auf Urlaub.
B ... Ich fahre im August auf Urlaub.
Struktur: Entweder A oder B (ausschließendes „oder“).
Formel: $A \not\equiv B$

(f) *Ich koche heute nicht, lasse mir jedoch eine Pizza liefern.*
A ... Ich koche heute.
B ... Ich lasse mir eine Pizza liefern.
Struktur: nicht A, aber (und) B.
Formel: $\neg A \wedge B$

Bei der Formalisierung von „jedoch“/„aber“ durch den und-Operator geht ein Teil der Bedeutung verloren, das lässt sich aber aufgrund der beschränkten Ausdrucksmöglichkeiten in der Aussagenlogik nicht vermeiden.

(g) *Wenn der Bus nicht rechtzeitig kommt, so werde ich nicht pünktlich sein.*
A ... Der Bus kommt rechtzeitig.
B ... Ich bin pünktlich.
Struktur: Wenn nicht A, dann nicht B.
Formel: $\neg A \supset \neg B$

Ist aus dem Zusammenhang klar, dass die pünktliche Ankunft ausschließlich vom Bus abhängt, kann man auch den Äquivalenzoperator verwenden. Wenn es noch andere Faktoren gibt, die die Ankunft verzögern können, ist die Äquivalenz falsch.

(h) *Ich putze nur dann die Fenster, wenn es nicht regnet.*
A ... Ich putze die Fenster.
B ... Es regnet.
Struktur: A nur dann, wenn nicht B.
Formel: $A \supset \neg B$

Eine Äquivalenz statt der Implikation ist hier falsch, denn diese würde ja bedeuten, dass die Fenster *immer* geputzt werden, sobald es nicht regnet.

Aufgabe 3 (0.4 Punkte)

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

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

Anmerkung: Die Begründung, dass in jedem Ausdruck immer mindestens eine Variable vorkommen muss und es daher nicht möglich ist, nullstellige Funktionen (also Konstanten wie false) darzustellen, ist nicht ausreichend, da das nur ein Problem der gewählten Darstellung ist. Wenn Sie aber zeigen können, dass beispielsweise die einstellige konstante Funktion definiert durch $\mathsf{false}(x) = 0$ nicht darstellbar ist, ist das schlüssig.

Lösung

(a) Aus der Vorlesung wissen wir, dass die Menge $\{\mathsf{nand}\}$ funktional vollständig ist. Für die funktionale Vollständigkeit von $\{\mathsf{implies}, \mathsf{false}\}$ genügt es daher zu zeigen, dass nand durch die beiden Funktionen implies und false dargestellt werden kann. Um diese Darstellung zu finden, greifen wir auf die in der Vorlesung besprochenen Äquivalenzen zurück:

$$A \uparrow B = \neg A \vee \neg B = A \supset \neg B = A \supset (B \supset \bot)$$

Es gilt also

$$x \ \mathsf{nand} \ y = x \ \mathsf{implies} \ (y \ \mathsf{implies} \ \mathsf{false}) \ ,$$

was sich auch unabhängig von den Äquivalenzen durch Überprüfung aller Wahrheitsbelegungen zeigen lässt:

x	y	x nand y	$=$	x implies	(y implies	false)
0	0	**1**	✓	**1**	1	0
0	1	**1**	✓	**1**	0	0
1	0	**1**	✓	**1**	1	0
1	1	**0**	✓	**0**	0	0

Anstelle von {nand} kann jede andere als funktional vollständig bekannte Menge herangezogen werden.

(b) Es genügt von einer einzigen Funktion zu zeigen, dass sie nicht durch implies darstellbar ist. Wir untersuchen, welche einstelligen Funktionen darstellbar sind, indem wir vom einzigen Argument x ausgehend implies mehrfach anwenden. Wir stellen zunächst fest, dass x implies $x = 1$ gilt. Unter Berücksichtigung von 1 als weiterem Argument erhalten wir zusätzlich noch 1 implies $x = x$, x implies $1 = 1$ und 1 implies $1 = 1$. Wir stellen daher folgende Behauptung auf.

Induktionsbehauptung: Jeder Ausdruck bestehend aus implies und x ist äquivalent zu x oder 1.

Der Beweis erfolgt induktiv nach der Anzahl n der Anwendungen der Funktion implies.

Induktionsanfang $n = 0$: Der einzige Ausdruck ohne Anwendung von implies ist x selber, daher gilt unsere Behauptung für $n = 0$.

Induktionshypothese: Unsere Behauptung gelte für alle Ausdrücke mit n oder weniger Anwendungen von implies.

Induktionsschritt: Wir zeigen, dass unter der Annahme, dass die Induktionshypothese zutrifft, unsere Behauptung auch für Ausdrücke mit $n+1$ Anwendungen von implies gilt. Wir betrachten also einen Ausdruck implies$(f(x), g(x))$, bei dem sowohl f als auch g mit n oder weniger Anwendungen von implies definiert sind. Laut Hypothese ist jeder der beiden Ausdrücke äquivalent zu x oder 1. Wie wir oben aber festgestellt haben, liefert implies mit den Argumenten x bzw. 1 wieder nur x oder 1. Damit gilt die Behauptung auch für den Fall $n + 1$.

Da somit die einzigen darstellbaren einstelligen Funktionen äquivalent zu x bzw. 1 sind, ist z.B. die einstellige Funktion not nicht darstellbar. Die Menge {implies} ist daher nicht funktional vollständig.

Aufgabe 4 **(0.3 Punkte)**

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

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

(m2) Wenn $x \in \mathcal{M}$, dann $\texttt{*}x\texttt{*} \in \mathcal{M}$.

(m3) Wenn $x, y \in \mathcal{M}$, dann $x\texttt{+}y \in \mathcal{M}$.

(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, zehn typische Elemente anzugeben.

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

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

Lösung

(a)
$$
\begin{aligned}
\mathcal{M}_0 &= \{\texttt{a}, \texttt{b}\},\\
\mathcal{M}_1 &= \mathcal{M}_0 \cup \{\texttt{*a*, *b*, a+a, a+b, b+a, b+b}\},\\
\mathcal{M}_2 &= \mathcal{M}_1 \cup \{\texttt{**a**, **b**, *a+a*, *a+b*, *b+a*, *b+b*,}\\
&\qquad \texttt{a+*a*, a+*b*, a+a+a, a+a+b, a+b+a, a+b+b,}\\
&\qquad \texttt{b+*a*, b+*b*, b+a+a, b+a+b, b+b+a, b+b+b,}\\
&\qquad \texttt{*a*+a, *a*+b, *a*+*a*, *a*+*b*,}\\
&\qquad \texttt{*a*+a+a, *a*+a+b, *a*+b+a, *a*+b+b,}\\
&\qquad \texttt{*b*+a, *b*+b, *b*+*a*, *b*+*b*,}\\
&\qquad \texttt{*b*+a+a, *b*+a+b, *b*+b+a, *b*+b+b,}\\
&\qquad \texttt{a+a+*a*, a+a+*b*, a+a+a+a, a+a+a+b, a+a+b+a, a+a+b+b,}\\
&\qquad \texttt{a+b+*a*, a+b+*b*, a+b+a+a, a+b+a+b, a+b+b+a, a+b+b+b,}\\
&\qquad \texttt{b+a+*a*, b+a+*b*, b+a+a+a, b+a+a+b, b+a+b+a, b+a+b+b,}\\
&\qquad \texttt{b+b+*a*, b+b+*b*, b+b+a+a, b+b+a+b, b+b+b+a, b+b+b+b}\}
\end{aligned}
$$

(b) Um zu zeigen, dass ein Wort in einer induktiv definierten Menge liegt, muss man das Wort schrittweise mit Hilfe der Eigenschaften (Konstruktionsregeln) erzeugen. Wir ordnen wenn-dann Schlüsse vertikal mit einer Trennlinie an. Oberhalb der Linie stehen die Voraussetzungen, unterhalb das Ergebnis der Regelanwendung, und daneben steht die Bezeichnung der verwendeten Eigenschaft.

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

In Worten bedeutet das:

(1) Wegen Eigenschaft m1 gilt $\texttt{a} \in \mathcal{M}$.

(2) Wegen Eigenschaft m1 gilt $\texttt{b} \in \mathcal{M}$.

(3) Aus Punkt (1) und Eigenschaft m2 folgt $\texttt{*a*} \in \mathcal{M}$.

(4) Aus Punkt (2) und Eigenschaft m2 folgt $\texttt{*b*} \in \mathcal{M}$.

(5) Aus den Punkten (3) und (4) sowie Eigenschaft m3 folgt $\texttt{*a*+*b*} \in \mathcal{M}$.

(6) Aus den Punkten (1) und (5) sowie Eigenschaft m3 folgt $\texttt{a+*a*+*b*} \in \mathcal{M}$.

(7) Aus Punkt (6) und Eigenschaft m2 folgt $\texttt{*a+*a*+*b**} \in \mathcal{M}$.

Die Herleitung des Wortes $\texttt{*a+*a*+*b**}$ ist nicht eindeutig. Eine andere, gleichwertige Argumentation sieht so aus:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\texttt{a}\in\mathcal{M}}\,\text{m1} \qquad \dfrac{\dfrac{}{\texttt{a}\in\mathcal{M}}\,\text{m1}}{\texttt{*a*}\in\mathcal{M}}\,\text{m2}}{\texttt{a+*a*}\in\mathcal{M}}\,\text{m3} \qquad \dfrac{\dfrac{}{\texttt{b}\in\mathcal{M}}\,\text{m1}}{\texttt{*b*}\in\mathcal{M}}\,\text{m2}}{\texttt{a+*a*+*b*}\in\mathcal{M}}\,\text{m3}}{\texttt{*a+*a*+*b**}\in\mathcal{M}}\,\text{m2}$$

(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 $2i + 1$; das lässt sich mit einem kurzen Induktionsbeweis zeigen. Da das Wort $\texttt{**a+b*a+b}$ die Länge 9 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 9 541 122 Elemente.

Argumentation über die Anzahl der Vorkommnisse des Symbols $\texttt{*}$. Es lässt sich leicht feststellen, dass jedes Wort in der Menge $\mathcal{M}$ die Eigenschaft besitzt, dass die Anzahl der vorkommenden Sterne gerade ist, etwa so:

- Die beiden Wörter, die durch Eigenschaft m1 eingeführt werden, besitzen keine Sterne, also eine gerade Anzahl.
- Besitzt x n Sterne, dann besitzt das durch Eigenschaft m2 konstruierte Wort $\texttt{*}x\texttt{*}$ $n + 2$ Sterne. Falls n gerade ist, ist auch $n + 2$ gerade.
- Besitzen x und y jeweils m bzw. n Sterne, dann besitzt das durch Eigenschaft m3 konstruierte Wort $x\texttt{+}y$ $m + n$ Sterne. Falls m und n gerade sind, ist auch $m + n$ gerade.

Da das untersuchte Wort `**a+b*a+b` drei Sterne, also eine ungerade Anzahl, enthält, kann es nicht in der Menge $\mathcal{M}$ liegen.

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

Aufgabe 5 (0.3 Punkte)

Sei F die Formel $(((A \land B) \supset C) \equiv (A \supset (B \supset C)))$.

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

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

(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 $* \in \{\land, \uparrow, \lor, \downarrow, \equiv, \not\equiv, \supset, \subset\}$.

$\mathcal{V} = \{A, B, C, \dots, A_0, A_1, \dots\}$ (aussagenlogische Variablen)

Wir zeigen, dass $(((A \land B) \supset C) \equiv (A \supset (B \supset C)))$ 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 $(A \land B)$ eine Formel (a4).

(3) Da $(A \land B)$ und C Formeln sind (Punkt 2 bzw. 1), ist auch $((A \land B) \supset C)$ eine Formel (a4).

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

(5) Da A und $(B \supset C)$ Formeln sind (Punkt 1 bzw. 4), ist auch $(A \supset (B \supset C))$ eine Formel (a4).

(6) Da $((A \land B) \supset C)$ und $(A \supset (B \supset C))$ Formeln sind (Punkt 3 bzw. 6), ist auch $(((A \land B) \supset C) \equiv (A \supset (B \supset C)))$ eine Formel (a4).

Gemäß Vereinbarung in der Vorlesung kann man als Schreibvereinfachung die äußeren Klammern weglassen, womit wir auch $((A \land B) \supset C) \equiv (A \supset (B \supset C))$ als korrekte Formel auffassen.

Dieselbe Argumentation in Form eines Ableitungsbaumes:

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{A \in \mathcal{V}}{A \in \mathcal{A}}\,\text{a1}
      \qquad
      \dfrac{B \in \mathcal{V}}{B \in \mathcal{A}}\,\text{a1}
    }{(A \wedge B) \in \mathcal{A}}\,\text{a4}
    \qquad
    \dfrac{C \in \mathcal{V}}{C \in \mathcal{A}}\,\text{a1}
  }{((A \wedge B) \supset C) \in \mathcal{A}}\,\text{a4}
  \qquad
  \dfrac{
    \dfrac{A \in \mathcal{V}}{A \in \mathcal{A}}\,\text{a1}
    \qquad
    \dfrac{
      \dfrac{B \in \mathcal{V}}{B \in \mathcal{A}}\,\text{a1}
      \qquad
      \dfrac{C \in \mathcal{V}}{C \in \mathcal{A}}\,\text{a1}
    }{(B \supset C) \in \mathcal{A}}\,\text{a4}
  }{(A \supset (B \supset C)) \in \mathcal{A}}\,\text{a4}
}{(((A \wedge B) \supset C) \equiv (A \supset (B \supset C))) \in \mathcal{A}}\,\text{a4}
$$

(b) $\mathsf{val}_I(((A \wedge B) \supset C) \equiv (A \supset (B \supset C)))$
$= \mathsf{val}_I((A \wedge B) \supset C) \text{ iff } \mathsf{val}_I(A \supset (B \supset C))$
$= (\mathsf{val}_I(A \wedge B) \text{ implies } \mathsf{val}_I(C)) \text{ iff } (\mathsf{val}_I(A) \text{ implies } \mathsf{val}_I(B \supset C))$
$= ((\mathsf{val}_I(A) \text{ and } \mathsf{val}_I(B)) \text{ implies } 1) \text{ iff } (0 \text{ implies } (\mathsf{val}_I(B) \text{ implies } \mathsf{val}_I(C)))$
$= ((0 \text{ and } 1) \text{ implies } 1) \text{ iff } (0 \text{ implies } (1 \text{ implies } 1))$
$= (0 \text{ implies } 1) \text{ iff } (0 \text{ implies } 1)$
$= 1 \text{ iff } 1$
$= 1$

(c) Die Formel F ist erfüllbar, da es eine Interpretation I gibt, in der sie wahr ist, etwa $I(A) = I(B) = 1, I(C) = 0$. Sie ist nicht widerlegbar, da es keine Interpretation I gibt, in der sie falsch ist. Es folgt unmittelbar, dass die Formel gültig ist. Dies lässt sich systematisch mittels einer Wahrheitstafel prüfen:

A	B	C	$((A \wedge$	$B) \supset C)$	$\equiv$	$(A \supset$	$(B \supset C))$
1	1	1	1	1	**1**	1	1
1	1	0	1	0	**1**	0	0
1	0	1	0	1	**1**	1	1
1	0	0	0	1	**1**	1	1
0	1	1	0	1	**1**	1	1
0	1	0	0	1	**1**	1	0
0	0	1	0	1	**1**	1	1
0	0	0	0	1	**1**	1	1

Aufgabe 6 (0.3 Punkte)

Zeigen Sie, dass die beiden Formeln $((A \supset B) \not\equiv C) \wedge A \wedge ((B \vee C) \supset \bot)$ und $(A \supset (\neg A \supset (B \not\equiv C))) \supset \bot$ äquivalent sind.

(a) mithilfe einer Wahrheitstafel;

(b) durch algebraische Umformungen.

Lösung

(a) Wahrheitstafel:

A	B	C	$((A \supset B) \not\equiv C) \wedge (A \wedge ((B \vee C) \supset \bot)) = (A \supset (\neg A \supset (B \not\equiv C))) \supset \bot$											
0	0	0	1	1	**0**	0	0	1	✓	1	1	0	0	**0**
0	0	1	1	0	**0**	0	1	0	✓	1	1	1	1	**0**
0	1	0	1	1	**0**	0	1	0	✓	1	1	1	1	**0**
0	1	1	1	0	**0**	0	1	0	✓	1	1	0	0	**0**
1	0	0	0	0	**0**	1	0	1	✓	1	0	1	0	**0**
1	0	1	0	1	**0**	0	1	0	✓	1	0	1	1	**0**
1	1	0	1	1	**0**	0	1	0	✓	1	0	1	1	**0**
1	1	1	1	0	**0**	0	1	0	✓	1	0	1	0	**0**

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

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

$$
\begin{array}{ll}
((A \supset B) \not\equiv C) \wedge A \wedge ((B \vee C) \supset \bot) & \\
= ((\neg A \vee B) \not\equiv C) \wedge A \wedge (\neg(B \vee C) \vee \bot) & \text{da } F \supset G = \neg F \vee G \\
= ((\neg A \vee B) \not\equiv C) \wedge A \wedge \neg(B \vee C) & \text{da } F \vee \bot = F \\
= ((\neg A \vee B) \not\equiv C) \wedge A \wedge \neg B \wedge \neg C & \text{De Morgan} \\
= (\neg A \vee B \vee C) \wedge (\neg(\neg A \vee B) \vee \neg C) \wedge A \wedge \neg B \wedge \neg C & \text{da } F \not\equiv G = (F \vee G) \wedge (\neg F \vee \neg G) \\
= \neg(A \wedge \neg B \wedge \neg C) \wedge ((\neg\neg A \wedge \neg B) \vee \neg C) \wedge (A \wedge \neg B \wedge \neg C) & \text{De Morgan} \\
= \bot \wedge ((\neg\neg A \wedge \neg B) \vee \neg C) & \text{da } \neg F \wedge F = \bot \\
= \bot & \text{da } \bot \wedge F = \bot
\end{array}
$$

$$
\begin{array}{ll}
(A \supset (\neg A \supset (B \not\equiv C))) \supset \bot & \\
= \neg(\neg A \vee (\neg\neg A \vee (B \not\equiv C))) \vee \bot & \text{da } F \supset G = \neg F \vee G \\
= \neg(\neg A \vee (\neg\neg A \vee (B \not\equiv C))) & \text{da } F \vee \bot = F \\
= \neg\neg A \wedge \neg(\neg\neg A \vee (B \not\equiv C)) & \text{De Morgan} \\
= A \wedge \neg(A \vee (B \not\equiv C)) & \neg\neg F = F \\
= A \wedge \neg A \wedge \neg(B \not\equiv C) & \text{De Morgan} \\
= \bot \wedge \neg(B \not\equiv C) & \text{da } \neg F \wedge F = \bot \\
= \bot & \text{da } \bot \wedge F = \bot
\end{array}
$$

Aufgabe 7 (0.3 Punkte)

Ist die Formel B eine logische Konsequenz der drei Formeln $A \supset B$, $B \vee (A \not\equiv \neg C)$ und $A \wedge C$? Wie sieht die Formel aus, deren Gültigkeit/Nichtgültigkeit zeigen würde, dass die Konsequenzbeziehung gilt/nicht gilt?

Lösung

$I(A)$	$I(B)$	$I(C)$	$A \supset B,$	$B \vee (A \not\equiv \neg C),$	$A \wedge C$	$\models_I$	B
0	0	0	1	1	0	✓	0
0	0	1	1	0	0	✓	0
0	1	0	1	1	0	✓	1
0	1	1	1	1	0	✓	1
1	0	0	0	0	0	✓	0
1	0	1	0	1	1	✓	0
1	1	0	1	1	0	✓	1
1	1	1	1	1	1	✓	1

Die Formel B ist somit eine logische Konsequenz der Prämissen.
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. Wertet man in diesem Beispiel die Formeln von rechts nach links aus, ergibt sich folgende vereinfachte Tabelle:

$I(A)$	$I(B)$	$I(C)$	$A \supset B,$	$B \vee (A \not\equiv \neg C),$	$A \wedge C$	$\models_I$	B
0	0	0			0	✓	0
0	0	1			0	✓	0
0	1	0				✓	1
0	1	1				✓	1
1	0	0			0	✓	0
1	0	1	0	1	1	✓	0
1	1	0				✓	1
1	1	1				✓	1

Formel zur Konsequenzbeziehung: B ist genau dann eine logische Konsequenz der drei Formeln $A \supset B$, $B \vee (A \not\equiv \neg C)$ und $A \wedge C$, wenn die Formel

$$((A \supset B) \wedge (B \vee (A \not\equiv \neg C)) \wedge (A \wedge C)) \supset B$$

gültig ist.

Aufgabe 8 (0.2 Punkte)

Sei f folgende dreistellige Funktion.

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

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

Stellen Sie f durch eine Formel in

(a) disjunktiver

(b) konjunktiver

Normalform dar.

Lösung

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

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

Aufgabe 9 (0.2 Punkte)

Sei F die Formel $(A \subset (B \uparrow C)) \vee (C \not\equiv (B \downarrow A))$.

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

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

Lösung

(a) KNF mittels semantischer Methode:

A	B	C	$(A \subset$	$(B \uparrow C))$	$\vee$	$(C \not\equiv$	$(B \downarrow A))$
0	0	0	0	1	**1**	1	1
0	0	1	0	1	**0**	0	1
0	1	0	0	1	**0**	0	0
0	1	1	1	0	**1**	1	0
1	0	0	1	1	**1**	0	0
1	0	1	1	1	**1**	1	0
1	1	0	1	1	**1**	0	0
1	1	1	1	0	**1**	1	0

Aus dieser Tafel lässt sich folgende KNF ablesen:

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

(b) DNF mittels algebraischer Methode:

$$\begin{aligned}
&(A \subset (B \uparrow C)) \vee (C \not\equiv (B \downarrow A)) \\
&= (A \vee \neg\neg(B \wedge C)) \vee (C \wedge \neg\neg(B \vee A)) \vee (\neg C \wedge \neg(B \vee A)) \\
&= (A \vee (B \wedge C)) \vee (C \wedge (B \vee A)) \vee (\neg C \wedge \neg B \wedge \neg A) \\
&= A \vee (B \wedge C) \vee (C \wedge B) \vee (C \wedge A) \vee (\neg C \wedge \neg B \wedge \neg A) \\
&= A \vee (B \wedge C) \vee (\neg A \wedge \neg B \wedge \neg C)
\end{aligned}$$

Aufgabe 10 **(0.4 Punkte)**

Hexe Berta versucht einen neuen Zaubertrank zu kreieren, der ihr ewige Jugend garantieren soll. Sie stellt folgende Überlegungen an:

- Ich brauche auf jeden Fall Flubberwurmschleim, damit der Trank dickflüssig wird.
- Ich sollte auch noch Aalaugen oder Belladonnaessenz dazugeben, aber nur eines davon.
- Wenn ich Belladonnaessenz oder Einhornhaare verwende, dann brauche ich keinen Wolfswurz mehr.
- Ich nehme Aalaugen oder Einhornhaare, vielleicht sogar beide.
- Wenn ich Aalaugen nehme, dann auch Belladonnaessenz und Flubberwurmschleim.

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

(b) Welche Zutaten nimmt Hexe Berta für den Zaubertrank? Begründen Sie die Antwort mit Hilfe Ihrer aussagenlogischen Modellierung.

Lösung

(a) Aussagenvariablen und ihre Bedeutung:

A ... Berta nimmt Aalaugen.
B ... Berta nimmt Belladonnaessenz.
E ... Berta nimmt Einhornhaare.
F ... Berta nimmt Flubberwurmschleim.
W ... Berta nimmt Wolfswurz.

Aussagenlogische Formeln:

$F_0 := F$	Flubberwurmschleim auf jeden Fall
$F_1 := A \not\equiv B$	entweder Aalaugen oder Belladonnaessenz
$F_2 := (B \vee E) \supset \neg W$	wenn Beladonnaessenz oder Einhornhaare dann kein Wolfswurz
$F_3 := A \vee E$	Aalaugen oder Einhornhaare oder beide
$F_4 := A \supset (B \wedge F)$	wenn Aalaugen dann Belladonnaessenz und Flubberwurmschleim

(b) Wir suchen alle Wahrheitsbelegungen für die Variablen A, B, C, D, und E, sodass die Formeln $F_0, \ldots, F_5$ wahr werden. Wegen der Formel F_0 genügt es, jene Belegungen zu betrachten, in denen F wahr ist.

F	A	B	E	W	$A \not\equiv B$	$(B \vee E) \supset \neg W$	$A \vee E$	$A \supset (B \wedge F)$	
1	0	0	0	0	0				
1	0	0	0	1	0				
1	0	0	1	0	0				
1	0	0	1	1	0				
1	0	1	0	0	1	1	0		
1	0	1	0	1	1	0			
1	0	1	1	0	1	1	1	1	✓
1	0	1	1	1	1	0			
1	1	0	0	0	1	1	1	0	
1	1	0	0	1	1	1	1	0	
1	1	0	1	0	1	1	1	0	
1	1	0	1	1	1	0			
1	1	1	0	0	0				
1	1	1	0	1	0				
1	1	1	1	0	0				
1	1	1	1	1	0				

Sie nimmt Flubberwurmschleim, Belladonnaessenz und Einhornhaare.

Aufgabe 11 **(0.4 Punkte)**

Nach bestandener Matura macht sich Susanne Gedanken darüber, was sie studieren könnte. Ihr Lieblingsfach in der Schule war Mathematik; Programmieren mochte sie nicht wirklich. Bei der Studienberatung erhält sie folgende Informationen.

- Mochte man in der Schule Mathematik, eignet man sich für ein Studium der Technischen Mathematik oder Informatik.
- Wenn man kein Interesse an der Programmierung hatte, sollte man Informatik meiden.
- Interesse an Mathematik und Programmieren spricht jedoch immer für ein Studium der Informatik.
- Man sollte (wegen der Arbeitsbelastung) nicht mit zwei Studien gleichzeitig beginnen.

(a) Formalisieren Sie die beschriebene Situation inklusive aller Anhaltspunkte mittels aussagenlogischer Formeln. Geben Sie die Bedeutung jeder Aussagenvariablen an.

(b) Kann Susanne aus diesen Informationen Empfehlungen ableiten? Wenn ja, welche? Begründen Sie Ihre Antwort mit Hilfe Ihrer aussagenlogischen Modellierung.

Lösung

(a) Aussagenvariablen und ihre Bedeutung:

M ... Susanne mag Mathematik.
P ... Susanne mag Programmieren.
T ... Susanne sollte Technische Mathematik studieren.
I ... Susanne sollte Informatik studieren.

Aussagenlogische Formeln:

$F_0 := M \wedge \neg P$	Susanne mag Mathematik, aber nicht Programmieren.
$F_1 := M \supset (T \vee I)$	Wenn sie MM mag, kann sie TM oder Inf studieren.
$F_2 := \neg P \supset \neg I$	Wenn sie Prog nicht mag, sollte sie nicht Inf studieren.
$F_3 := (M \wedge P) \supset I$	Bei Interesse für MM und Prog sollte sie Informatik studieren.
$F_4 := \neg(T \wedge I)$	Sie sollte nicht beide Studien gleichzeitig studieren.

(b) Wir suchen alle Wahrheitsbelegungen für die Variablen M, P, T, und I, sodass die Formeln $F_0, \ldots, F_4$ wahr werden. Die Formel F_0 bedingt, dass M wahr und P falsch sein muss. Es bleiben vier Möglichkeiten für die Belegung von T und I:

M	P	T	I	$M \wedge \neg P$	$M \supset (T \vee I)$	$\neg P \supset \neg I$	$(M \wedge P) \supset I$	$\neg(T \wedge I)$	
1	0	0	0	1	0	1	1	1	
1	0	0	1	1	1	0	1	1	
1	0	1	0	1	1	1	1	1	✓
1	0	1	1	1	1	0	1	0	

Somit ergibt sich eine eindeutige Empfehlung: Susanne sollte Mathematik studieren.

Aufgabe 12 (0.4 Punkte)

Seien $F_1, \ldots, F_n$ aussagenlogische Formeln. Zeigen Sie, dass $F_1, \ldots, F_n \models \bot$ genau dann gilt, wenn $\neg F_1 \vee \cdots \vee \neg F_n$ gültig ist.

Lösung

Wir wissen aus der Vorlesung:

> $F_1, \ldots, F_n \models G$ gilt genau dann, wenn die Formel $(F_1 \wedge \cdots \wedge F_n) \supset G$ gültig ist (Deduktionstheorem).

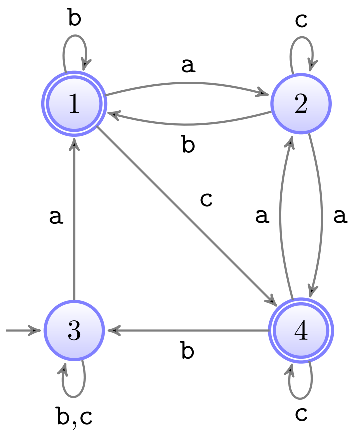
Somit gilt $F_1, \ldots, F_n \models \bot$ genau dann, wenn $(F_1 \wedge \cdots \wedge F_n) \supset \bot$ gültig ist. Umformung liefert:

$$\begin{aligned}
&(F_1 \wedge \cdots \wedge F_n) \supset \bot \\
&= \neg(F_1 \wedge \cdots \wedge F_n) \vee \bot \\
&= \neg(F_1 \wedge \cdots \wedge F_n) \\
&= \neg F_1 \vee \cdots \vee \neg F_n
\end{aligned}$$

Wir erhalten also: $F_1, \ldots, F_n \models \bot$ gilt genau dann, wenn $\neg F_1 \vee \cdots \vee \neg F_n$ gültig ist.

Aufgabe 13 **(0.4 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: ε, `ab`, `acbc`, `aa`, `abac`.

(c) Berechnen Sie schrittweise $\delta^*(3, \texttt{acbab})$.

(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 `a`, `ab`, `ac`, `ba` und `acba`.

(b) $\mathcal{A}$ akzeptiert `ab`, nicht aber ε, `acbc`, `aa` und `abac`.

(c)
$$\begin{aligned}
& \delta^*(3, \texttt{acbab}) && = \delta^*(\delta(3, \texttt{a}), \texttt{cbab}) \\
= {} & \delta^*(1, \texttt{cbab}) && = \delta^*(\delta(1, \texttt{c}), \texttt{bab}) \\
= {} & \delta^*(4, \texttt{bab}) && = \delta^*(\delta(4, \texttt{b}), \texttt{ab}) \\
= {} & \delta^*(3, \texttt{ab}) && = \delta^*(\delta(3, \texttt{a}), \texttt{b}) \\
= {} & \delta^*(1, \texttt{b}) && = \delta^*(\delta(1, \texttt{b}), \varepsilon) \\
= {} & \delta^*(1, \varepsilon) && = 1
\end{aligned}$$

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

δ	a	b	c
1	2	1	4
2	4	1	2
3	1	3	3
4	2	3	4

$\mathcal{A}$ 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 (0.3 Punkte)

Ein elektronisches Tresorschloss besteht aus einem zweistelligen Display sowie den Tasten `+`, `L`, `R`, `Ok` und `Reset`. Jede Stelle des Displays kann eine der drei Ziffern 0, 1 oder 2 anzeigen. Mit jedem Drücken der `+`-Taste ändert sich die Anzeige der aktiven Stelle von 0 auf 1, von 1 auf 2 bzw. von 2 auf 0. Welche der beiden Stellen aktiv ist, lässt sich durch die `L`- und `R`-Taste kontrollieren: Ein- oder mehrmaliges Drücken der `L`- bzw. `R`-Taste aktiviert die linke bzw. rechte Stelle. Im Anfangszustand zeigt das Display die Zahl 00 an und die linke Stelle ist aktiviert. Wird die Zahl 21 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 zum Beispiel mit jeder der beiden folgenden Tastenkombinationen öffnen:

```
+ + R + Ok
+ Reset + L R + L + 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 3) pro Stelle sowie k Stellen (statt 2) besitzt?

- Legen Sie die möglichen Aktionen fest, die zu einem Zustandswechsel führen.

- Geben Sie einen endlichen Automaten an, der das Verhalten des beschriebenen Schlosses vollständig beschreibt. Die Sprache des Automaten sollen genau jene Tastenkombinationen sein, die den Tresor öffnen. Spezifizieren Sie die Übergangsfunktion des Automaten mittels einer Tabelle.

Lösung

Der Zustand des Schlosses wird durch die angezeigten Ziffern sowie durch die Position der Aktivierung eindeutig festgelegt. Daher besitzt das Schloss $3^2 \cdot 2 + 2 = 20$ 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 `+`, `L`, `R`, `Ok` und `Reset`. Als Zustandsbezeichnung wählen wir $\underline{a}b$ bzw. $a\underline{b}$, wobei ab den angezeigten Ziffern entspricht und die Unterstreichung die aktivierte Stelle markiert.

Das Verhalten des Schlosses wird durch den Automaten

$$\langle \{\underline{0}0, \ldots, 2\underline{2}, \mathit{Fehler}, \mathit{Offen}\}, \{\texttt{+}, \texttt{L}, \texttt{R}, \texttt{Ok}, \texttt{Reset}\}, \delta, \underline{0}0, \{\mathit{Offen}\}\rangle$$

beschrieben, wobei die Übergangsfunktion δ durch folgende Tabelle definiert wird.

δ	`+`	`L`	`R`	`Ok`	`Reset`
$\underline{0}0$	$\underline{1}0$	$\underline{0}0$	$0\underline{0}$	*Fehler*	$\underline{0}0$
$\underline{1}0$	$\underline{2}0$	$\underline{1}0$	$1\underline{0}$	*Fehler*	$\underline{0}0$
$\underline{2}0$	$\underline{0}0$	$\underline{2}0$	$2\underline{0}$	*Fehler*	$\underline{0}0$
$\underline{0}1$	$\underline{1}1$	$\underline{0}1$	$0\underline{1}$	*Fehler*	$\underline{0}0$
$\underline{1}1$	$\underline{2}1$	$\underline{1}1$	$1\underline{1}$	*Fehler*	$\underline{0}0$
$\underline{2}1$	$\underline{0}1$	$\underline{2}1$	$2\underline{1}$	*Offen*	$\underline{0}0$
$\underline{0}2$	$\underline{1}1$	$\underline{0}2$	$0\underline{2}$	*Fehler*	$\underline{0}0$
$\underline{1}2$	$\underline{2}1$	$\underline{1}2$	$1\underline{2}$	*Fehler*	$\underline{0}0$
$\underline{2}2$	$\underline{0}1$	$\underline{2}2$	$2\underline{2}$	*Fehler*	$\underline{0}0$
$0\underline{0}$	$0\underline{1}$	$\underline{0}0$	$0\underline{0}$	*Fehler*	$\underline{0}0$
$1\underline{0}$	$1\underline{1}$	$\underline{1}0$	$1\underline{0}$	*Fehler*	$\underline{0}0$
$2\underline{0}$	$2\underline{1}$	$\underline{2}0$	$2\underline{0}$	*Fehler*	$\underline{0}0$
$0\underline{1}$	$0\underline{2}$	$\underline{0}1$	$0\underline{1}$	*Fehler*	$\underline{0}0$
$1\underline{1}$	$1\underline{2}$	$\underline{1}1$	$1\underline{1}$	*Fehler*	$\underline{0}0$
$2\underline{1}$	$2\underline{2}$	$\underline{2}1$	$2\underline{1}$	*Offen*	$\underline{0}0$
$0\underline{2}$	$0\underline{0}$	$\underline{0}2$	$0\underline{2}$	*Fehler*	$\underline{0}0$
$1\underline{2}$	$1\underline{0}$	$\underline{1}2$	$1\underline{2}$	*Fehler*	$\underline{0}0$
$2\underline{2}$	$2\underline{0}$	$\underline{2}2$	$2\underline{2}$	*Fehler*	$\underline{0}0$
Fehler	*Fehler*	*Fehler*	*Fehler*	*Fehler*	$\underline{0}0$
Offen	*Offen*	*Offen*	*Offen*	*Offen*	$\underline{0}0$

Aufgabe 15 **(0.4 Punkte)**

Ein Radiowecker mit Schlummertaste besitze folgendes Verhalten. Sobald die eingestellte Alarmzeit erreicht ist (`A`), beginnt er entweder laut zu piepen (`p`) oder er spielt das eingestellte Radioprogramm (`r`). Drückt man auf die Schlummertaste (`S`), dann ist der Wecker drei Minuten still (`s`), ehe er wieder zu piepen bzw. Radio zu spielen beginnt. Wird zu einem beliebigen Zeitpunkt der Alarm ausgeschaltet (`O` wie „off“), dann geht der Wecker zurück in den Wartezustand. Mittels eines Umschalters (`U`) kann zwischen Radio und Piepton gewechselt werden; zu Beginn ist der Piepton ausgewählt. Um die Zeit zu messen, erhält der Wecker von einem internen Zeitgeber jede Minute einen Tick (`T`).

Modellieren Sie den Wecker mithilfe eines Moore- oder Mealy-Automaten. Eingangssignale sind `A`, `S`, `T`, `U` und `O`, Ausgangssignale sind `p`, `r` und `s`. Sie können den Automaten tabellarisch oder graphisch darstellen.

Geben Sie die Ausgabe für folgende Eingangssignale an:

(a) `ATTO`

(b) `UTTASTTTTSO`

(c) `TTAUTTO`

Lösung

Wir geben einen Moore-Automaten mit 10 Zuständen an: Wartezustand W, Alarmzustand A sowie Schlummerzustände S_0, S_1 und S_2 (Schlummerfunktion nach 0, 1 bzw. 2 Minuten). Diese Zustände verdoppeln sich durch die Möglichkeit, dass der Wecker piepen oder Radio spielen kann; die Radio-Zustände sind mit einem Apostroph markiert.
Der Automat wird durch das Tupel

$$\langle \{W, A, S_1, S_2, S_3, W', A', S_1', S_2', S_3'\}, \{\mathtt{A}, \mathtt{O}, \mathtt{S}, \mathtt{T}, \mathtt{U}\}, \{\mathtt{p}, \mathtt{r}, \mathtt{s}\}, \delta, \gamma, W \rangle$$

beschrieben, wobei die Übergangsfunktion δ und die Ausgabefunktion γ durch folgende Tabelle definiert sind.

	δ					γ
	`A`	`O`	`S`	`T`	`U`	
W	A	W	W	W	W'	`s`
A	A	W	S_0	A	A'	`p`
S_0	A	W	S_0	S_1	S_0'	`s`
S_1	A	W	S_1	S_2	S_1'	`s`
S_2	A	W	S_2	A	S_2'	`s`
W'	A'	W'	W'	W'	W	`s`
A'	A'	W'	S_0'	A'	A	`r`
S_0'	A'	W'	S_0'	S_1'	S_0	`s`
S_1'	A'	W'	S_1'	S_2'	S_1	`s`
S_2'	A'	W'	S_2'	A'	S_2	`s`

Alternative Lösung, falls das Drücken der Schlummertaste die bereits laufende Schlummerzeit verlängert:

	δ					γ
	`A`	`O`	`S`	`T`	`U`	
W	A	W	W	W	W'	`s`
A	A	W	S_0	A	A'	`p`
S_0	A	W	S_0	S_1	S_0'	`s`
S_1	A	W	$\mathbf{S_0}$	S_2	S_1'	`s`
S_2	A	W	$\mathbf{S_0}$	A	S_2'	`s`
W'	A'	W'	W'	W'	W	`s`
A'	A'	W'	S_0'	A'	A	`r`
S_0'	A'	W'	S_0'	S_1'	S_0	`s`
S_1'	A'	W'	$\mathbf{S_0'}$	S_2'	S_1	`s`
S_2'	A'	W'	$\mathbf{S_0'}$	A'	S_2	`s`

Für die angegebenen Eingabefolgen erhalten wir:

(a) $\gamma^*(W, \mathtt{ATTO}) = $ `ppps`

(b) $\gamma^*(W, \texttt{UTTASTTTTSO}) = \texttt{sssrsssrrss}$

(c) $\gamma^*(W, \texttt{TTAUTTO}) = \texttt{ssprrrs}$