1. Übungsblatt (mit Lösungen)

3.0 VU Formale Modellierung

Marion Brandsteidl, Gernot Salzer

19. April 2013

Aufgabe 1 (0.3 Punkte)

Gegeben sei folgender Sachverhalt:

(A) Kein Murks ist ein Klax.
(B) Mindestens ein Troll ist ein Klax.

Nur eine der folgenden Aussagen ist eine zulässige Konklusion. Welche?

(C1) Kein Troll ist ein Murks.
(C2) Mindestens ein Troll ist ein Murks.
(C3) Mindestens ein Troll ist kein Murks.
(C4) Jeder Troll ist ein Murks.
(C5) Jeder Troll ist kein Murks.

Ersetzen Sie in der entsprechenden zulässigen Inferenz die Begriffe Murks, Klax und Troll durch alltägliche Begriffe, sodass

(a) … sowohl die Prämissen als auch die Konklusion wahr werden;

(b) … mindestens eine der Prämissen falsch und die Konklusion wahr wird;

(c) … alle Prämissen wahr werden und die Konklusion falsch wird;

(d) … sowohl mindestens eine der Prämissen als auch die Konklusion falsch wird.

Eine dieser vier Teilaufgaben wird Ihnen schwer fallen zu lösen. Welche? Und warum?

Lösung

A besagt, dass die Schnittmenge von Murksen und Klaxen leer ist.
B besagt, dass die Mengen der Trolle und Klaxe mindestens ein gemeinsames Element e besitzen.

Ob die Mengen der Trolle und Murkse überlappen (C2) oder nicht (C1, C5), wird durch A und B nicht eindeutig festgelegt, keine der drei Aussagen C1, C2 und C5 ist daher eine zulässige (zwingende) Konklusion.
C4 steht im Widerspruch zu den Annahmen A und B: e ist ein Troll und ein Klax (B), aber kein Murks (A). Daher widerlegt die Existenz von e die Behauptung, jeder Troll sei auch ein Murks.
C3 hingegen ist eine zulässige Konklusion: e ist ein Beispiel für einen Troll, der kein Murks ist. Die Inferenz besitzt somit folgenden Aufbau:

$$\begin{array}{l} \text{Kein } x \text{ ist ein } y. \\ \text{Mindestens ein } z \text{ ist ein } y. \\ \hline \text{Mindestens ein } z \text{ ist kein } x. \end{array}$$

(a) x = Kreis, y = Quadrat, z = Rechteck:

Kein Kreis ist ein Quadrat.	wahr
Mindestens ein Rechteck ist ein Quadrat.	wahr
Mindestens ein Rechteck ist kein Kreis.	wahr

(b) x = Quadrat, y = Kreis, z = Rechteck:

Kein Quadrat ist ein Kreis.	wahr
Mindestens ein Rechteck ist ein Kreis.	falsch
Mindestens ein Rechteck ist kein Quadrat.	wahr

(c) Diese Teilaufgabe ist nicht lösbar. Könnte man durch geeignete Wahl von Begriffen für x, y und z die Prämissen wahr und die Konklusion falsch machen, wäre es keine zulässige Schlussfolgerung.

(d) x = Rechteck, y = Kreis, z = Quadrat:

Kein Rechteck ist ein Kreis.	wahr
Mindestens ein Quadrat ist ein Kreis.	falsch
Mindestens ein Quadrat ist kein Rechteck.	falsch

Aufgabe 2 (0.3 Punkte)

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

(a) Ich habe gute Laune.

(b) Ich habe Hunger und Durst.

(c) Wir bekommen nur dann frei, wenn die Temperatur über 35 Grad steigt.

(d) Ich verbringe das Wochenende in Rom oder Paris.

(e) Mir wird kalt werden, wenn ich meine Jacke vergesse.

(f) Ich kann an der Lehrveranstaltung nicht teilnehmen, wenn ich den Eingangstest nicht positiv absolviere.

(g) Der Direktor und sein Stellvertreter gehen nicht gleichzeitig auf Urlaub.

(h) Wenn die Sonne scheint, ist es nie der Fall, dass Susi und Bernd gemeinsam in die Vorlesung gehen.

Lösung

(a) A ... Ich habe gute Laune.
Struktur: A
Formel: A

(b) A ... Ich habe Hunger.
B ... Ich habe Durst.
Struktur: A und B
Formel: $A \wedge B$

(c) A ... Wir bekommen frei.
B ... Die Temperatur steigt über 35 Grad.
Struktur: A nur dann, wenn B. Oder: Wenn A, dann B.
Formel: $A \supset B$.

(d) A ... Ich verbringe das Wochenende in Rom.
B ... Ich verbringe das Wochenende in Paris.
Struktur: A oder B
Formel: $A \vee B$

(e) A ... Mir wird kalt werden.
B ... Ich vergesse meine Jacke.
Struktur: A, wenn B.
Formel: $A \subset B$

(f) A ... Ich kann an der Lehrveranstaltung teilnehmen.
B ... Ich absolviere den Eingangstest positiv.
Struktur: Nicht A, wenn nicht B.
Formel: $\neg A \subset \neg B$

(g) A ... Der Direktor geht auf Urlaub.
B ... Der Stellvertreter geht auf Urlaub.
Struktur: A nicht gleichzeitig mit B. Oder: Entweder nicht A oder nicht B.
Formel: $\neg(A \wedge B)$ oder $\neg A \vee \neg B$

(h) A ... Die Sonne scheint.
B ... Susi geht in die Vorlesung.
C ... Bernd geht in die Vorlesung.
Struktur: Wenn A, dann nicht gleichzeitig B und C.
Formel: $A \supset \neg(B \wedge C)$

Aufgabe 3 (0.4 Punkte)

(a) Zeigen Sie, dass die Menge $\{\mathsf{nor}\}$ vollständig ist für die Klasse der aussagenlogischen Funktionen.

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

Lösung

(a) Aus der Vorlesung wissen wir, dass die Menge $\{\mathsf{not}, \mathsf{and}\}$ funktional vollständig ist. Wir zeigen, dass die Funktionen not und and durch nor dargestellt werden können. Denn es gilt

$$\begin{aligned} \mathsf{not}\, x &= x \;\mathsf{nor}\; x \\ x \;\mathsf{and}\; y &= (x \;\mathsf{nor}\; x) \;\mathsf{nor}\; (y \;\mathsf{nor}\; y) \end{aligned}$$

wie sich durch Überprüfung aller Wahrheitsbelegungen nachweisen lässt:

x	y	x nor y	not x	$=$	x nor x	x and y	$=$	(x nor x)	nor	(y nor y)
1	1	0				**1**	✓	0	**1**	0
1	0	0	**0**	✓	**0**	**0**	✓	0	**0**	1
0	1	0	**1**	✓	**1**	**0**	✓	1	**0**	0
0	0	1				**0**	✓	1	**0**	1

(b) Es genügt von einer einzigen Funktion zu zeigen, dass sie nicht durch xor darstellbar ist. Wir untersuchen, welche einstelligen Funktionen darstellbar sind, indem wir vom einzigen Argument x ausgehend xor mehrfach anwenden. Wir stellen zunächst fest, dass $x \;\mathsf{xor}\; x = 0$ gilt. Unter Berücksichtigung von 0 als weiterem Argument erhalten wir zusätzlich noch $x \;\mathsf{xor}\; 0 = x$, $0 \;\mathsf{xor}\; x = x$ und $0 \;\mathsf{xor}\; 0 = 0$. Wir stellen daher folgende Behauptung auf.

Induktionsbehauptung: Jeder Ausdruck bestehend aus xor und x ist äquivalent zu x oder 0.

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

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

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

Induktionsschritt: Wir zeigen, dass unter der Annahme, dass die Induktionshypothese zutrifft, unsere Behauptung auch für Ausdrücke mit $n+1$ Anwendungen von xor gilt. Wir haben also einen Ausdruck $\mathsf{xor}(f(x), g(x))$ vor uns, bei dem sowohl f als auch g mit n oder weniger Anwendungen von xor definiert sind. Laut Hypothese ist jeder der beiden Ausdrücke äquivalent zu x oder 0. Wie wir oben aber festgestellt haben, liefert xor mit den Argumenten x bzw. 0 wieder nur x oder 0. Damit gilt die Behauptung auch für den Fall $n+1$.

Da somit die einzigen darstellbaren einstelligen Funktionen äquivalent zu x bzw. 0 sind, ist z.B. die einstellige Funktion not nicht darstellbar. Die Menge $\{\mathsf{xor}\}$ ist daher nicht funktional vollständig.

Anmerkung für Puristen: Die Argumentation ist ein wenig schlampig, da sie Funktionen und Werte mischt. Genau genommen wollen wir nachweisen, dass ausgehend von der identischen Abbildung id (definiert durch $\mathsf{id}(x) = x$ für alle x) durch Anwendung von xor wieder nur id oder die einstellige Nullfunktion zero (definiert durch $\mathsf{zero}(x) = 0$ für alle x) entstehen kann. Mehr dazu unter dem Stichwort „Klon“ (engl. *clone*) im Internet oder in Büchern zur Algebra.

Aufgabe 4 (0.3 Punkte)

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

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

(m2) Wenn $x \in \mathcal{M}$, dann auch $x\mathtt{[}x\mathtt{]} \in \mathcal{M}$.

(m3) Wenn $x, y \in \mathcal{M}$, dann auch $x\mathtt{!}y \in \mathcal{M}$.

(a) Geben Sie die Mengen $\mathcal{M}_0$, $\mathcal{M}_1$ und $\mathcal{M}_2$ der stufenweise Konstruktion von $\mathcal{M}$ an. (Siehe die Vorlesungspräsentation vom 12.3.2013, Seite 23.) Die Menge $\mathcal{M}_2$ enthält bereits mehr als 200 Elemente; es genügt, 10 typische Elemente anzugeben.

(b) Zeigen Sie, dass die Zeichenkette `b!c[b!c]!a[a][a[a]]` in der Menge $\mathcal{M}$ liegt.

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

Lösung

(a) $\mathcal{M}_0 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$
$\mathcal{M}_1 = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{a[a]}, \mathtt{b[b]}, \mathtt{c[c]}, \mathtt{a!a}, \mathtt{a!b}, \mathtt{a!c}, \mathtt{b!a}, \mathtt{b!b}, \mathtt{b!c}, \mathtt{c!a}, \mathtt{c!b}, \mathtt{c!c}\}$
$\mathcal{M}_2 = \mathcal{M}_1 \cup \{\mathtt{a[a][a[a]]}, \mathtt{b[b][b[b]]}, \ldots, \mathtt{a!a[a!a]}, \mathtt{a!b[a!b]}, \ldots,$
$\mathtt{a!a[a]}, \mathtt{a!b[b]}, \ldots, \mathtt{a[a]!a}, \mathtt{a[a]!b}, \ldots,$
$\mathtt{a[a]!a[a]}, \mathtt{a[a]!b[b]}, \ldots, \mathtt{a!a!a[a]}, \ldots, \mathtt{a!a!a!a}, \ldots\}$

(b) Wir verwenden die Notation $A \overset{r}{\Longrightarrow} B$ um auszudrücken, dass Aussage B aus Aussage A mit Hilfe der Regel r folgt, wobei r entweder m2 oder m3 sein kann.

1. Wegen Regel m1 gilt $\texttt{a} \in \mathcal{M}$, $\texttt{b} \in \mathcal{M}$ und $\texttt{c} \in \mathcal{M}$.
2. $\texttt{b}, \texttt{c} \in \mathcal{M} \overset{\text{m3}}{\Longrightarrow} \texttt{b!c} \in \mathcal{M} \overset{\text{m2}}{\Longrightarrow} \texttt{b!c[b!c]} \in \mathcal{M}$
3. $\texttt{a} \in \mathcal{M} \overset{\text{m2}}{\Longrightarrow} \texttt{a[a]} \in \mathcal{M} \overset{\text{m2}}{\Longrightarrow} \texttt{a[a][a[a]]} \in \mathcal{M}$
4. $\texttt{b!c[b!c]} \in \mathcal{M}$ (siehe 2.), $\texttt{a[a][a[a]]} \in \mathcal{M}$ (siehe 3.)
   $\overset{\text{m3}}{\Longrightarrow} \texttt{b!c[b!c]!a[a][a[a]]} \in \mathcal{M}$

(c) Wir zeigen, dass die Zeichenfolge `![` nicht in einem Wort aus $\mathcal{M}$ vorkommen kann. Dazu verwenden wir folgende Eigenschaft von $\mathcal{M}$: Jedes Wort in $\mathcal{M}$ beginnt mit einem der Zeichen `a`, `b` oder `c`. Diese Eigenschaft lässt sich leicht induktiv zeigen:

m1: Die Wörter `a`, `b` und `c` beginnen mit einem dieser Zeichen.

m2: Wenn x mit einem dieser Zeichen beginnt, dann auch $x\texttt{[}x\texttt{]}$.

m3: Wenn x mit einem dieser Zeichen beginnt, dann auch $x\texttt{!}y$.

Das Zeichen `!` kann nur durch die Eigenschaft m3 in ein Wort gelangen. Laut dieser Regel muss aber ein Wort $y \in \mathcal{M}$ folgen, das unserer Feststellung zufolge aber mit `a`, `b` oder `c` beginnen muss. Daher enthalten Wörter aus $\mathcal{M}$ nie die Zeichenfolge `![`, somit kann `a[a][a[a]]!b![b]` nicht in $\mathcal{M}$ vorkommen.

Alternativ lässt sich auch die Eigenschaft nutzen, dass Wörter in $\mathcal{M}$ nie mit `!` enden (ebenfalls leicht induktiv zu zeigen). Da `[` nur durch die Eigenschaft m2 in ein Wort gelangen kann, dann aber ein Wort $x \in \mathcal{M}$ vorangehen muss, enthalten Wörter aus $\mathcal{M}$ nie die Zeichenfolge `![`.

Aufgabe 5 (0.3 Punkte)

Sei F die Formel $(((A \not\equiv B) \wedge C) \supset (B \vee \neg C))$.

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

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

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

Lösung

(a) Wir verwenden die Notation $A \overset{r}{\Longrightarrow} B$ um auszudrücken, dass Aussage B aus Aussage A mit Hilfe der Regel r folgt, wobei r entweder a1, a2, a3 oder a4 sein kann (siehe die induktive Definition von $\mathcal{A}$, der Menge der aussagenlogischen Formeln).

1. $A, B, C \in \mathcal{V} \overset{\text{a1}}{\Longrightarrow} A, B, C \in \mathcal{A}$
2. $A, B \in \mathcal{A}$ (siehe 1.) $\overset{\text{a4}}{\Longrightarrow} (A \not\equiv B) \in \mathcal{A}$
3. $(A \not\equiv B) \in \mathcal{A}$ (siehe 2.), $C \in \mathcal{A}$ (siehe 1.) $\overset{\text{a4}}{\Longrightarrow} ((A \not\equiv B) \wedge C) \in \mathcal{A}$

4. $C \in \mathcal{A}$ (siehe 1.) $\overset{\text{a3}}{\Longrightarrow} \neg C \in \mathcal{A}$

5. $B \in \mathcal{A}$ (siehe 1.), $\neg C \in \mathcal{A}$ (siehe 4.) $\overset{\text{a4}}{\Longrightarrow} (B \vee \neg C) \in \mathcal{A}$

6. $((A \not\equiv B) \wedge C) \in \mathcal{A}$ (siehe 3.), $(B \vee \neg C) \in \mathcal{A}$ (siehe 5.)
$\overset{\text{a4}}{\Longrightarrow} (((A \not\equiv B) \wedge C) \supset (B \vee \neg C)) \in \mathcal{A}$

(b) $\mathsf{val}_I((((A \not\equiv B) \wedge C) \supset (B \vee \neg C)))$
$= \mathsf{val}_I(((A \not\equiv B) \wedge C))$ implies $\mathsf{val}_I((B \vee \neg C))$
$= (\mathsf{val}_I((A \not\equiv B))$ and $\mathsf{val}_I(C))$ implies $(\mathsf{val}_I(B)$ or $\mathsf{val}_I(\neg C))$
$= (\mathsf{val}_I((A \not\equiv B))$ and $I(C))$ implies $(\mathsf{val}_I(B)$ or $\mathsf{val}_I(\neg C))$
$= (\cdots$ and $0)$ implies $\cdots$
$= 0$ implies $\cdots$
$= 1$

(c)

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

Da die Formel in der Wahrheitsbelegung mit $I(A) = I(C) = 1$ und $I(B) = 0$ den Wert 0 und in einer beliebigen anderen den Wert 1 besitzt, ist die Formel widerleg- und erfüllbar, aber weder gültig noch unerfüllbar.

Anmerkung: Da wir aus der vorherigen Teilaufgabe bereits wissen, dass die Formel mindestens erfüllbar ist, hätte es auch gereicht, gezielt nach einer Wahrheitsbelegung zu suchen, die die Formel falsch macht. Auch ist es nicht notwendig, die gesamte Formel auszuwerten, wenn ihr Wert bereits nach der Auswertung eines ihrer Teile feststeht. Insofern hätte bei systematischer Suche nach der falsifizerenden Belegung auch folgende partielle Tabelle gereicht:

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

Aufgabe 6 (0.3 Punkte)

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

(a) mithilfe einer Wahrheitstafel;

(b) durch algebraische Umformungen.

Lösung

(a) Unter Verwendung der Lösung zu Aufgabe 5c, in der die zweite Formel bereits ausgewertet wurde, erhalten wir:

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

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

(b) Wir zeigen die Äquivalenz, indem wir beide Formeln zur selben DNF (bzw. KNF) vereinfachen.

$$\begin{aligned}
((A \not\equiv B) \wedge C) \supset (B \vee \neg C) &= \neg((A \not\equiv B) \wedge C) \vee B \vee \neg C \\
&= \neg(A \not\equiv B) \vee \neg C \vee B \vee \neg C \\
&= \neg(A \not\equiv B) \vee B \vee \neg C \\
&= \neg((A \vee B) \wedge (\neg A \vee \neg B)) \vee B \vee \neg C \\
&= \neg(A \vee B) \vee \neg(\neg A \vee \neg B) \vee B \vee \neg C \\
&= (\neg A \wedge \neg B) \vee (A \wedge B) \vee B \vee \neg C \\
&= (\neg A \wedge \neg B) \vee B \vee \neg C \\
&= ((\neg A \vee B) \wedge (\neg B \vee B)) \vee \neg C \\
&= ((\neg A \vee B) \wedge \top) \vee \neg C \\
&= \neg A \vee B \vee \neg C
\end{aligned}$$

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

Aufgabe 7 (0.3 Punkte)

Untersuchen Sie, ob die Formel $G := A$ eine logische Konsequenz der drei Formeln $F_1 := A \supset B$, $F_2 := \neg C \supset A$ und $F_3 := A \vee \neg(B \vee C)$ ist. Geben Sie eine Formel H an, die genau dann in einer Interpretation I wahr ist, wenn $F_1, F_2, F_3 \models_I G$ gilt.

Lösung

G ist eine Konsequenz der Formeln F_1, F_2 und F_3, wenn die Beziehung $F_1, F_2, F_3 \models_I G$ für alle Interpretationen I gilt.

$I(A)$	$I(B)$	$I(C)$	F_1, $A \supset B$	F_2, $\neg C \supset A$	F_3 $A \vee \neg(B \vee C)$	$\models_I$	G A
1	1	1	1	1	1	✓	1
1	1	0	1	1	1	✓	1
1	0	1	0			✓	
1	0	0	0			✓	
0	1	1	1	1	0	✓	
0	1	0	1	0		✓	
0	0	1	1	1	0	✓	
0	0	0	1	0		✓	

Da G immer dann wahr ist, wenn sämtliche Prämissen wahr sind, gilt die Konsequenzbeziehung. Sie entspricht der Formel $H := (F_1 \wedge F_2 \wedge F_3) \supset G$.

Aufgabe 8 (0.3 Punkte)

Sei f folgende dreistellige Funktion.

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

Stellen Sie f durch eine Formel in

(a) disjunktiver

(b) konjunktiver

Normalform dar.

Lösung

(a) $$\begin{aligned}\mathsf{DNF}_f &= K_{110} \vee K_{101} \vee K_{100} \vee K_{001} \vee K_{000} \\ &= (A_1 \wedge A_2 \wedge \neg A_3) \vee (A_1 \wedge \neg A_2 \wedge A_3) \vee (A_1 \wedge \neg A_2 \wedge \neg A_3) \\ &\quad \vee (\neg A_1 \wedge \neg A_2 \wedge A_3) \vee (\neg A_1 \wedge \neg A_2 \wedge \neg A_3)\end{aligned}$$

Eine einfachere, äquivalente DNF ist $(A_1 \wedge \neg A_3) \vee \neg A_2$.

(b) $\mathsf{KNF}_f = D_{111} \wedge D_{011} \wedge D_{010}$
$$= (\neg A_1 \vee \neg A_2 \vee \neg A_3) \wedge (A_1 \vee \neg A_2 \vee \neg A_3) \wedge (A_1 \vee \neg A_2 \vee A_3)$$
Eine einfachere, äquivalente KNF ist $(A_1 \vee \neg A_2) \wedge (\neg A_2 \vee \neg A_3)$.

Aufgabe 9 (0.3 Punkte)

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

(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) Wir erstellen die Wahrheitstafel und lesen daran eine DNF ab.

A	B	C	$((B \supset C)$	$\downarrow A)$	$\vee$	$\neg$	$(B \vee$	$(\neg\, C$	$\not\equiv A))$
1	1	1		0	**0**	0	1		
1	1	0		0	**0**	0	1		
1	0	1		0	**0**	0	1	0	1
1	0	0		0	**1**	1	0	1	0
0	1	1	1	0	**0**	0	1		
0	1	0	0	1	**1**				
0	0	1	1	0	**1**	1	0	0	0
0	0	0	1	0	**0**	0	1	1	1

$\mathsf{DNF}_F = K_{100} \vee K_{010} \vee K_{001} = (A \wedge \neg B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge \neg B \wedge C)$

(b)
$$\begin{aligned}
&((B \supset C) \downarrow A) \vee \neg(B \vee (\neg C \not\equiv A)) \\
&= (\neg(\neg B \vee C) \wedge \neg A) \vee \neg(B \vee (\neg\neg C \wedge A) \vee (\neg C \wedge \neg A)) \\
&= (B \wedge \neg C \wedge \neg A) \vee (\neg B \wedge \neg(C \wedge A) \wedge \neg(\neg C \wedge \neg A)) \\
&= (B \wedge \neg C \wedge \neg A) \vee (\neg B \wedge (\neg C \vee \neg A) \wedge (C \vee A)) \\
&= (B \vee \neg B) \wedge (B \vee \neg C \vee \neg A) \wedge (B \vee C \vee A) \\
&\quad \wedge (\neg C \vee \neg B) \wedge (\neg C \vee \neg C \vee \neg A) \wedge (\neg C \vee C \vee A) \\
&\quad \wedge (\neg A \vee \neg B) \wedge (\neg A \vee \neg C \vee \neg A) \wedge (\neg A \vee C \vee A) \\
&= \top \wedge (\neg A \vee B \vee \neg C) \wedge (A \vee B \vee C) \\
&\quad \wedge (\neg B \vee \neg C) \wedge (\neg A \vee \neg C) \wedge \top \\
&\quad \wedge (\neg A \vee \neg B) \wedge (\neg A \vee \neg C) \wedge \top \\
&= (\neg A \vee \neg B) \wedge (\neg A \vee \neg C) \wedge (\neg B \vee \neg C) \wedge (A \vee B \vee C)
\end{aligned}$$

Aufgabe 10 (0.3 Punkte)

Zeigen Sie: $F_1, \ldots, F_n \models G$ gilt genau dann, wenn die Formel $F_1 \wedge \cdots \wedge F_n \wedge \neg G$ unerfüllbar ist.

Lösung

Beweisstrategie:

- Eine „A genau dann, wenn B“-Aussage kann man zeigen, indem man sie in zwei Implikationen zerlegt: Man zeigt, dass aus A die Aussage B folgt und dass aus B die Aussage A folgt. Diese Zerlegung vereinfacht oft die Argumentation.
- Die Äquivalenz von A und B lässt sich auch dadurch zeigen, dass man die Äquivalenz der Gegenteile von A und B überprüft, d.h., man zeigt, dass A genau dann *nicht* zutrifft, wenn B *nicht* zutrifft. Der Wechsel zu den negierten Aussagen ist sinnvoll, wenn sich einfach charakterisieren lässt, wann eine Aussage *nicht* gilt.

Sei H die Formel $F_1 \wedge \cdots \wedge F_n \wedge \neg G$. Wir zeigen die beiden folgenden Aussagen:

1. Wenn $F_1, \ldots, F_n \models G$ nicht gilt, dann ist H erfüllbar.

   Beweis: Wenn $F_1, \ldots, F_n \models G$ nicht gilt, gibt es eine Interpretation I, sodass $F_1, \ldots, F_n \models_I G$ nicht gilt. Das ist dann der Fall, wenn $\mathsf{val}_I(F_1) = \cdots = \mathsf{val}_I(F_n) = 1$ und $\mathsf{val}_I(G) = 0$ gilt. Aus $\mathsf{val}_I(G) = 0$ folgt $\mathsf{val}_I(\neg G) = 1$. Daher ist die Formel H in dieser Interpretation wahr, d.h., H ist erfüllbar.

2. Wenn H erfüllbar ist, dann gilt $F_1, \ldots, F_n \models G$ nicht.

   Beweis: Wenn H erfüllbar ist, gibt es eine Interpretation I, sodass $\mathsf{val}_I(H) = 1$ gilt. Das ist der Fall, wenn $\mathsf{val}_I(F_1) = \cdots = \mathsf{val}_I(F_n) = \mathsf{val}_I(\neg G) = 1$ gilt. Aus $\mathsf{val}_I(\neg G) = 1$ folgt $\mathsf{val}_I(G) = 0$, daher gilt $F_1, \ldots, F_n \models_I G$ für die Interpretation I nicht. Somit ist G keine logische Konsequenz der Formeln $F_1, \ldots, F_n$.

Aufgabe 11 (0.4 Punkte)

Ein *Baby-Sudoku* ist ein Rätsel, das aus vier 2×2-Quadraten besteht, in die die Zahlen 1 bis 4 geeignet einzusetzen sind. Es gelten Regeln analog zum regulären Sudoku[1]:

In jeder Zeile, in jeder Spalte und in jedem der markierten 2×2-Quadrate muss jede der Zahlen 1 bis 4 genau einmal vorkommen.

3			1
			3
		4	

Modellieren Sie diese Regeln durch aussagenlogische Formeln, sodass die erfüllenden Interpretationen den Lösungen entsprechen. Fügen Sie weitere Formeln hinzu, um das konkrete Baby-Sudoku oben zu modellieren.
Hinweis: Verwenden Sie Variablen der Form A_{ijk}, die für die Aussagen „In Zeile i und Spalte j steht die Zahl k“ stehen ($1 \leq i, j, k \leq 4$). Überlegen Sie sich zunächst eine Formel mit vier Variablen, die genau dann wahr ist, wenn irgendeine der Variablen wahr und die anderen falsch sind. Benennen Sie Teilformeln, die einen bestimmten Aspekt formalisieren, um die Gesamtformel zu strukturieren und lesbarer zu gestalten.

[1] `http://de.wikipedia.org/wiki/Sudoku`

Überlegen Sie, wie sich Ihr Lösungsansatz erweitern lässt, um die Regeln des regulären Sudoku (neun 3×3-Quadrate mit den Zahlen 1 bis 9) zu modellieren. Wie viele Variablen würden Sie benötigen? Wie groß wäre die entsprechende Formel?
Fleißaufgabe: Wählen Sie im Internet einen geeigneten SAT-Solver (z.B. MiniSat) aus und lösen Sie das angegebe Sudoku mittels Ihrer Formel.

Lösung

Wir definieren zunächst eine Formel mit vier Variablen, die genau dann wahr ist, wenn eine der Variablen wahr ist und die anderen falsch sind:

$$\begin{aligned}\mathsf{one}(A,B,C,D) := {}& (\neg A \vee \neg B) \wedge (\neg A \vee \neg C) \wedge (\neg A \vee \neg D) \wedge (\neg B \vee \neg C)\\ & \wedge (\neg B \vee \neg D) \wedge (\neg C \vee \neg D) \wedge (A \vee B \vee C \vee D)\end{aligned}$$

Alternativ kann man one auch durch die dazu äquivalenten Formeln

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

oder

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

definieren. Damit lassen sich die Sudoku-Bedingungen nun folgendermaßen formalisieren.

- Jedes Feld (i,j) ist mit genau einer der Zahlen 1–4 beschriftet:
  $F_{ij} := \mathsf{one}(A_{ij1}, A_{ij2}, A_{ij3}, A_{ij4})$ für alle $1 \leq i,j \leq 4$
- In jeder Zeile i kommt jede Zahl k genau einmal vor:
  $Z_{ik} := \mathsf{one}(A_{i1k}, A_{i2k}, A_{i3k}, A_{i4k})$ für alle $1 \leq i,k \leq 4$
- In jeder Spalte j kommt jede Zahl k genau einmal vor:
  $S_{jk} := \mathsf{one}(A_{1jk}, A_{2jk}, A_{3jk}, A_{4jk})$ für alle $1 \leq j,k \leq 4$
- In jedem der vier 2×2-Quadrate kommt jede Zahl k genau einmal vor:
  $Q_{1k} := \mathsf{one}(A_{11k}, A_{12k}, A_{21k}, A_{22k})$ für alle $1 \leq k \leq 4$
  $Q_{2k} := \mathsf{one}(A_{13k}, A_{14k}, A_{23k}, A_{24k})$ für alle $1 \leq k \leq 4$
  $Q_{3k} := \mathsf{one}(A_{31k}, A_{32k}, A_{41k}, A_{42k})$ für alle $1 \leq k \leq 4$
  $Q_{4k} := \mathsf{one}(A_{33k}, A_{34k}, A_{43k}, A_{44k})$ für alle $1 \leq k \leq 4$

Die zulässigen Beschriftungen von Baby-Sudokus werden durch die Konjunktion all dieser Teilformeln beschrieben:

$$B := \bigwedge_{1 \leq i,j \leq 4} F_{ij} \wedge \bigwedge_{1 \leq i,k \leq 4} Z_{ik} \wedge \bigwedge_{1 \leq j,k \leq 4} S_{jk} \wedge \bigwedge_{1 \leq k,l \leq 4} Q_{lk}$$

Die Interpretationen, die die Formel B erfüllen, entsprechen genau jenen Beschriftungen der Felder, die den Regeln folgen. Um das angegebene Beispiel-Sudoku zu formalisieren, muss man noch die Information über die bereits vorbefüllten Felder hinzufügen:

$$B \wedge A_{113} \wedge A_{141} \wedge A_{343} \wedge A_{434}$$

Zur Größenabschätzung: Die Formel B enthält 64 verschiedene Variablen und besteht aus 64 Kopien der Formel `one`. Da letztere 7 Konjunkte mit insgesamt 16 Literalen enthält, ist B eine KNF mit $64 \times 7 = 448$ Konjunkten und $64 \times 16 = 1024$ Literalen. Will man das reguläre Suduko auf dieselbe Art beschreiben, benötigt man $9{\times}9{\times}9 = 729$ Variablen. Die erweiterte Formel `one` besteht aus 37 Konjunkten mit insgesamt 81 Literalen. Die gesamte Sudoku-Formel besteht aus $4{\times}9{\times}9 = 324$ Kopien der Formel `one`, also aus 11988 Konjunkten mit 26244 Literalen.
Zur Fleißaufgabe: Man benötigt ein kleines Programm, um die oben beschriebenen Formeln zu generieren und im DIMACS Format abzuspeichern; dabei müssen auch die Variablen A_{ijk} in die Zahlen 1 bis 64 übersetzt werden. Weiters sollte das Programm die erfüllende Interpretation, die der SAT-Solver liefert, lesbar als quadratisches Sudoku darstellen. Die Archivdatei `sudoku.zip` (zu finden in TUWEL) enthält ein derartiges Programm (Programmiersprache Prolog), das für Sudokus beliebiger Größe (insbesondere 4×4 und 9×9) die entsprechenden Formeln generiert, den SAT-Solver MiniSat aufruft und die Lösung darstellt; mehr dazu in der README-Datei.

Aufgabe 12 **(0.4 Punkte)**

Schneewittchen möchte in die Stadt einkaufen gehen, benötigt dafür aber mindestens zwei Zwerge als Geleitschutz. Sneezy, Bashful und Dopey müssen Holz hacken und scheiden als Begleitung aus. Die anderen vier diskutieren, wer mit in die Stadt geht.

Grumpy: „Wir können Doc und Sleepy nicht gemeinsam gehen lassen."
Doc: „Wenn ich gehe, dann muss Happy mitkommen!"
Happy: „Ich gehe nur, wenn ich nicht gemeinsam mit Grumpy gehen muss."
Sleepy: „Ich komme dann und nur dann mit, wenn mich Happy oder Doc begleiten."

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

(b) Können sich die Zwerge entscheiden, wer Schneewittchen begleitet? Wenn ja, wer kommt als Begleitung in Frage? Begründen Sie Ihre Antwort mit Hilfe Ihrer aussagenlogischen Modellierung.

Lösung

(a) Aussagenvariablen und ihre Bedeutung:

D ... Doc geht mit in die Stadt.
G ... Grumpy geht mit in die Stadt.
H ... Happy geht mit in die Stadt.
S ... Sleepy geht mit in die Stadt.

Aussagenlogische Formeln:

$F_0 := (D \wedge G) \vee (D \wedge H) \vee (D \wedge S)$	
$\quad \vee (G \wedge H) \vee (G \wedge S) \vee (H \wedge S)$	mindestens zwei Zwerge
$F_1 := \neg(D \wedge S)$	Doc und Sleepy nicht gemeinsam
$F_2 := D \supset H$	wenn Doc, dann Happy
$F_3 := H \supset \neg G$	wenn Happy, dann nicht Grumpy
$F_4 := S \equiv (H \vee D)$	Sleepy genau dann, wenn Happy oder Doc

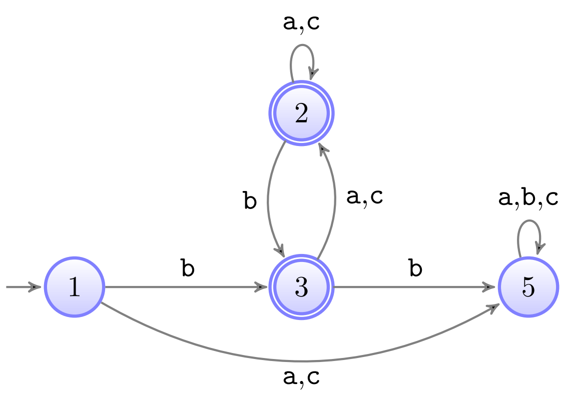
(b) Wir suchen alle Wahrheitsbelegungen für die Variablen D, G, H und S, sodass die Formeln $F_0, \ldots, F_4$ wahr werden. Wegen Formel F_0 genügt es, jene Belegungen zu betrachten, in denen mindestens zwei Variablen wahr sind.

D	G	H	S	$\neg(D \wedge S)$	$D \supset H$	$H \supset \neg G$	$S \equiv (H \vee D)$	
1	1	1	1	0				
1	1	1	0	1	1	0		
1	1	0	1	0				
1	1	0	0	1	0			
1	0	1	1	0				
1	0	1	0	1	1	1	0	
1	0	0	1	0				
0	1	1	1	1	1	0		
0	1	1	0	1	1	0		
0	1	0	1	1	1	1	0	
0	0	1	1	1	1	1	1	✓

Happy und Sleepy begleiten Schneewittchen in die Stadt.

Aufgabe 13 **(0.3 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 akzeptiert werden: ε, `b`, `babba`, `bcbac`, `bbac`.

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

(d) Beschreiben Sie $\mathcal{L}(\mathcal{A})$, die von $\mathcal{A}$ akzeptierte Sprache.

(e) 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 `b`, `ba`, `bc`, `baa` und `bab`.

(b) $\mathcal{A}$ akzeptiert `b` und `bcbac`, nicht aber ε, `babba` und `bbac`.

(c)
$$\begin{aligned}
\delta^*(1, \mathtt{babcc}) &= \delta^*(\delta(1, \mathtt{b}), \mathtt{abcc}) \\
&= \delta^*(3, \mathtt{abcc}) \\
&= \delta^*(\delta(3, \mathtt{a}), \mathtt{bcc}) \\
&= \delta^*(2, \mathtt{bcc}) \\
&= \delta^*(\delta(2, \mathtt{b}), \mathtt{cc}) \\
&= \delta^*(3, \mathtt{cc}) \\
&= \delta^*(\delta(3, \mathtt{c}), \mathtt{c}) \\
&= \delta^*(2, \mathtt{c}) \\
&= \delta^*(\delta(2, \mathtt{c}), \varepsilon) \\
&= \delta^*(2, \varepsilon) \\
&= 2
\end{aligned}$$

(d) $\mathcal{L}(\mathcal{A}) = \{\, w \in \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}^+ \mid w \text{ beginnt mit } \mathtt{b} \text{ und enthält nicht } \mathtt{bb} \,\}$

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

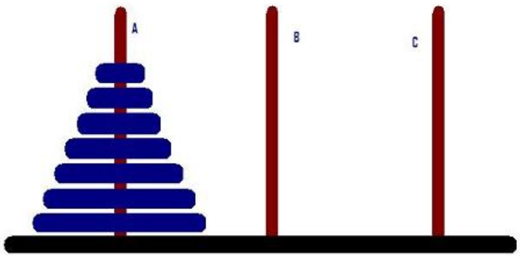
δ	a	b	c
1	5	3	5
2	2	3	2
3	2	5	2
5	5	5	5

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

Aufgabe 14 (0.4 Punkte)

Die *Türme von Hanoi* sind ein Rätsel, das aus drei senkrechten Stäben (A, B, C) und n gelochten, unterschiedlich großen Scheiben besteht. Zu Beginn befinden sich alle Scheiben nach Größe sortiert auf Stab A, mit der größten zuunterst. Ziel des Spieles ist es, den gesamten Turm zu Stab C zu verschieben.
Dabei sind folgende Regeln zu beachten:

- In jedem Zug wird die oberste Scheibe von einem der Stäbe entfernt und bei einem der anderen Stäbe zuoberst abgelegt.
- Es dürfen nur kleinere auf größere Scheiben gelegt werden.



(a) Was macht einen Zustand in diesem System aus? Welche Informationen sind notwendig, um einen Zustand eindeutig zu beschreiben? Wieviele verschiedene Zustände gibt es bei n Scheiben? Wie kann man die Zustände eindeutig aber kompakt bezeichnen?

(b) Was sind die Übergänge in diesem System? Welche und wieviele gibt es? Wie kann man sie eindeutig aber kompakt bezeichnen?

(c) Geben Sie einen endlichen Automaten an, der dieses System für $n = 2$ vollständig beschreibt. Sie können den Automaten tabellarisch oder graphisch darstellen.

Lösung

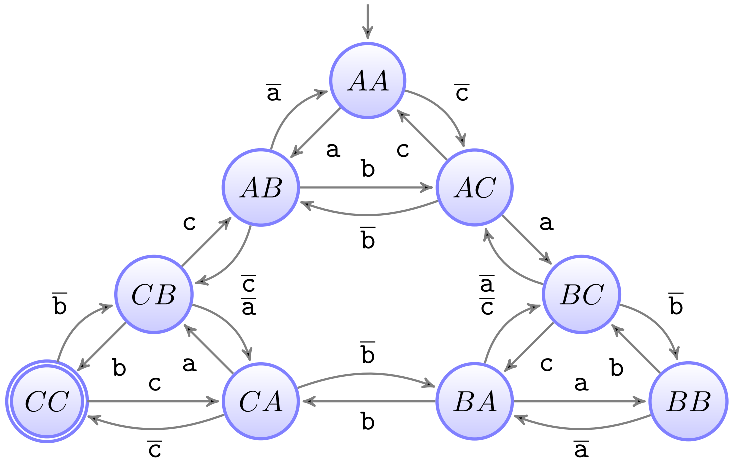
(a) Der Systemzustand wird durch die Position der Scheiben bestimmt. Um ihn eindeutig zu beschreiben, genügt es, zu jeder Scheibe den Stab festzuhalten, auf dem sie sich befindet; die Position der Scheibe auf dem Stab hingegen ergibt sich automatisch, da die Scheiben nach Größe sortiert sind. Für jede Scheibe gibt es drei mögliche Stäbe; bei n Scheiben ergeben sich 3^n verschiedene Aufteilungen der Scheiben, d.h., 3^n verschiedene Zustände. Ein Zustand wird eindeutig durch eine Zeichenkette $s_1 \cdots s_n$ mit $s_i \in \{A, B, C\}$ beschrieben, wobei $s_i = x$ bedeutet, dass sich Scheibe i auf dem Stab x befindet.

(b) Zustandsübergänge kommen durch das Verschieben einer Scheibe von einem Stab auf einen anderen zustande. Es gibt sechs verschiedene Aktionen: Verschieben der Scheibe von Stab A nach B ($\mathtt{a}$), von B nach C ($\mathtt{b}$), von C nach A ($\mathtt{c}$), und die umgekehrten Verschiebungen ($\overline{\mathtt{a}}$, $\overline{\mathtt{b}}$, $\overline{\mathtt{c}}$).

Weniger abstrakt kann man auch jeden Übergang mit einem eigenen Symbol beschriften; dieses steht dann für den Übergang von einem konkreten Zustand zum anderen. Für $n = 2$ besteht dann das Eingabealphabet aus 24 Symbolen. Diese Variante ist aber “weniger richtig”, da sie das System schlechter modelliert. Der

Unterschied wird z.B. sichtbar, wenn man die Wörter der Sprache als Handlungsanweisungen zur Lösung des Rätsels verwenden möchte, oder wenn man die Zahl der Scheiben erhöht.

(c)



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

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

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

Der Radiowecker als Transducer: Moore-Automaten sind Spezialfälle von Transducern. Der Moore-Automat oben lässt sich daher auch als Transducer $\mathcal{A} = \langle Q, \Sigma, \Gamma, \delta, I, F \rangle$ darstellen mit:

$$\begin{aligned} Q &= \{W, A, S_1, S_2, S_3, W', A', S_1', S_2', S_3'\}, \\ \Sigma &= \{\mathtt{A}, \mathtt{O}, \mathtt{S}, \mathtt{T}, \mathtt{U}\}, \\ \Gamma &= \{\mathtt{p}, \mathtt{r}, \mathtt{s}\}, \\ I &= \{W\} \\ F &= Q \end{aligned}$$

wobei die Übergangsrelation δ durch folgende Tabelle definiert ist:

δ	A	O	S	T	U
W	$\{(A,\mathtt{s})\}$	$\{(W,\mathtt{s})\}$	$\{(W,\mathtt{s})\}$	$\{(W,\mathtt{s})\}$	$\{(W',\mathtt{s})\}$
A	$\{(A,\mathtt{p})\}$	$\{(W,\mathtt{p})\}$	$\{(S_0,\mathtt{p})\}$	$\{(A,\mathtt{p})\}$	$\{(A',\mathtt{p})\}$
S_0	$\{(A,\mathtt{s})\}$	$\{(W,\mathtt{s})\}$	$\{(S_0,\mathtt{s})\}$	$\{(S_1,\mathtt{s})\}$	$\{(S_0',\mathtt{s})\}$
S_1	$\{(A,\mathtt{s})\}$	$\{(W,\mathtt{s})\}$	$\{(S_1,\mathtt{s})\}$	$\{(S_2,\mathtt{s})\}$	$\{(S_1',\mathtt{s})\}$
S_2	$\{(A,\mathtt{s})\}$	$\{(W,\mathtt{s})\}$	$\{(S_2,\mathtt{s})\}$	$\{(A,\mathtt{s})\}$	$\{(S_2',\mathtt{s})\}$
W'	$\{(A',\mathtt{s})\}$	$\{(W',\mathtt{s})\}$	$\{(W',\mathtt{s})\}$	$\{(W',\mathtt{s})\}$	$\{(W,\mathtt{s})\}$
A'	$\{(A',\mathtt{r})\}$	$\{(W',\mathtt{r})\}$	$\{(S_0',\mathtt{r})\}$	$\{(A',\mathtt{r})\}$	$\{(A,\mathtt{r})\}$
S_0'	$\{(A',\mathtt{s})\}$	$\{(W',\mathtt{s})\}$	$\{(S_0',\mathtt{s})\}$	$\{(S_1',\mathtt{s})\}$	$\{(S_0,\mathtt{s})\}$
S_1'	$\{(A',\mathtt{s})\}$	$\{(W',\mathtt{s})\}$	$\{(S_1',\mathtt{s})\}$	$\{(S_2',\mathtt{s})\}$	$\{(S_1,\mathtt{s})\}$
S_2'	$\{(A',\mathtt{s})\}$	$\{(W',\mathtt{s})\}$	$\{(S_2',\mathtt{s})\}$	$\{(A',\mathtt{s})\}$	$\{(S_2,\mathtt{s})\}$

Beim Übergang von der Darstellung als Moore-Automat zu einem Transducer werden alle Zustände zu Endzuständen. Für die Übergangsrelation gilt dann z.B.

$$\{(\mathtt{ATTO}, \mathtt{ppps}),\ (\mathtt{UTTASTTTTSO}, \mathtt{sssrsssrrss}),\ (\mathtt{TTAUTTO}, \mathtt{ssprrrs})\} \subseteq [\mathcal{A}] \ .$$