

# 1. Übungsblatt (mit Lösungen)

## 3.0 VU Formale Modellierung

Marion Scholz, Gernot Salzer

16. April 2014

### 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) Tweety ist ein Pinguin. Vögel können fliegen. Pinguine sind Vögel. Daher kann Tweety fliegen.
- (b) Schingals können liffen. Kein Bloggef ist ein Schingal. Daher können Bloggefs nicht liffen.
- (c) Güterzüge haben keinen Speisewagen. Alle Züge sind Güterzüge. Daher haben Züge keinen Speisewagen.

### Lösung

- (a) Tweety ist ein Pinguin. Inferenzregel:  $x$  ist ein  $y$ .  
Vögel können fliegen. Alle  $z$  können  $a$ .  
Pinguine sind Vögel. Alle  $y$  sind  $z$ . / Jedes  $y$  ist ein  $z$ .  

---

---

  
Tweety kann fliegen.  $x$  kann  $a$ .

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

Spindelegger ist Politiker.  
Menschen können schwindeln.  
Politiker sind Menschen.  

---

  
Spindelegger kann schwindeln.

- (b) Schingals können liffen. Inferenzregel: Alle  $x$  können  $y$ .  
Kein Bloggef ist ein Schingal. Kein  $z$  ist ein  $x$ .  
Bloggefs können nicht liffen. Alle  $z$  können nicht  $y$ . / Kein  $z$  kann  $y$ .

Diese Inferenzregel ist **nicht gültig**. Gegenbeispiel:

Hunde können laufen. (wahr)  
Keine Katze ist ein Hund. (wahr)  
Katzen können nicht laufen. (falsch)

Man erhält eine gültige Inferenzregel zum Beispiel durch Vertauschung von zweiter Prämisse und Konklusion:

Schingals können liffen.	Inferenzregel: Alle $x$ können $y$ .
<u>Blogefs können nicht liffen.</u>	<u>Alle <math>z</math> können nicht <math>y</math>.</u>
Kein Blogef ist ein Schingal.	Kein $z$ ist ein $x$ .

Andere Schlussfolgerungen mit dieser Inferenzregel:

Hunde können laufen. (wahr)	Hunde können bellen. (wahr)
<u>Katzen können nicht laufen. (falsch)</u>	<u>Katzen können nicht bellen. (wahr)</u>
Keine Katze ist ein Hund. (wahr)	Keine Katze ist ein Hund. (wahr)

(c) Güterzüge haben keinen Speisewagen.	Inferenzregel: Alle $x$ haben nicht $y$ .
<u>Alle Züge sind Güterzüge.</u>	<u>Alle <math>z</math> sind <math>x</math>.</u>
Züge haben keinen Speisewagen.	Alle $z$ haben nicht $y$ .

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

Menschen haben keine Flügel.  
Kinder sind Menschen.  
Kinder haben keine Flügel.

## Aufgabe 2 (0.4 Punkte)

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

- (a) Mir ist sehr kalt.
- (b) Ich werde den Zug verpassen, wenn ich mich nicht beeile.
- (c) Der Zirkusartist ist Hochseil- oder Trapezkünstler.
- (d) Es regnet und schneit nie gleichzeitig.
- (e) Eine Teilnahme an der Lehrveranstaltung ist nur möglich, wenn man in TISS angemeldet ist und das Studieneingangsgespräch absolviert hat.
- (f) Mir ist schlecht, der Kopf tut mir weh, aber ich bin nicht unglücklich.
- (g) Ich darf nur dann fernsehen, wenn ich meine Hausübungen gemacht habe.
- (h) Florian trinkt zum Essen entweder Bier oder Wein, aber nie beides gleichzeitig.

## Lösung

- (a) *Mir ist sehr kalt.*  
A ... Mir ist sehr kalt.  
Struktur: A  
Formel: A
- (b) *Ich werde den Zug verpassen, wenn ich mich nicht beeile.*  
A ... Ich verpasse den Zug.  
B ... Ich beeile mich.  
Struktur: A, wenn nicht B.  
Formel:  $A \supset \neg B$
- (c) *Der Zirkusartist ist Hochseil- oder Trapezkünstler.*  
A ... Der Zirkusartist ist Hochseilkünstler.  
B ... Der Zirkusartist ist Trapezkünstler.  
Struktur: A oder B  
Formel:  $A \vee B$
- (d) *Es regnet und schneit nie gleichzeitig.*  
A ... Es regnet.  
B ... Es schneit.  
Struktur: A nicht gleichzeitig mit B. Oder: Entweder nicht A oder nicht B.  
Formel:  $\neg(A \wedge B)$  oder  $\neg A \vee \neg B$  oder  $A \uparrow B$
- (e) *Eine Teilnahme an der Lehrveranstaltung ist nur möglich, wenn man in TISS angemeldet ist und das Studieneingangsgespräch absolviert hat.*  
A ... Eine Teilnahme an der Lehrveranstaltung ist möglich.  
B ... Man ist in TISS angemeldet.  
C ... Man hat das Studieneingangsgespräch absolviert.  
Struktur: A nur dann, wenn B und C.  
Formel:  $A \supset (B \wedge C)$
- (f) *Mir ist schlecht, der Kopf tut mir weh, aber ich bin nicht unglücklich.*  
A ... Mir ist schlecht.  
B ... Mir tut der Kopf weh.  
C ... Ich bin unglücklich.  
Struktur: A und B und nicht C.  
Formel:  $A \wedge B \wedge \neg C$

Anmerkung: „aber“ drückt nicht nur eine Konjunktion wie „und“ aus, sondern hebt außerdem einen Gegensatz zwischen den beiden Aussagen hervor. Dieser Aspekt ist in der Aussagenlogik nicht direkt ausdrückbar. Vergleichen Sie etwa die beiden Aussagen „Ich bin Bundespräsident, verdiene aber gut“ (klingt falsch) und „Ich bin Kindergärtnerin, verdiene aber gut“; eine Formalisierung mittels Konjunktion ist für einfache Analysen ausreichend, wenn nur der Beruf und das Einkommen von

Personen eine Rolle spielt, offenbar geht bei dieser Übersetzung aber Information über den Gegensatz der Aussagen verloren.

- (g) *Ich darf nur dann fernsehen, wenn ich meine Hausübungen gemacht habe.*

$A$  ... Ich darf fernsehen.

$B$  ... Ich habe meine Hausübungen gemacht.

Struktur:  $A$  nur dann, wenn  $B$ .

Formel:  $A \supset B$

- (h) *Florian trinkt zum Essen entweder Bier oder Wein, aber nie beides.*

$A$  ... Florian trinkt zum Essen Bier.

$B$  ... Florian trinkt zum Essen Wein.

Struktur: Entweder  $A$  oder  $B$  (ausschließendes „oder“)

Formel:  $A \neq B$

Diese Formalisierung bedeutet, dass Florian immer eines der beiden Getränke zum Essen trinkt. Soll die Aussage nur ausdrücken, dass er Bier und Wein nicht gleichzeitig trinkt, sonst aber die Möglichkeit offen lassen, dass Florian gar nichts oder andere Getränke konsumiert, ist folgende Formalisierung die richtige:

Struktur:  $A$  nicht gleichzeitig mit  $B$ .

Formel:  $\neg(A \wedge B)$  oder  $A \uparrow B$  oder  $\neg A \vee \neg B$

### Aufgabe 3 (0.2 Punkte)

Wieviele unterschiedliche drei-, vier- und fünfstellige aussagenlogische Funktionen gibt es? Wieviele  $n$ -stellige Funktionen gibt es, wenn der Wertebereich nicht zwei Elemente wie in der Aussagenlogik, sondern mehr Elemente (aber endlich viele) besitzt? Begründen Sie Ihre Antwort.

### Lösung

Die Anzahl der  $n$ -stelligen aussagenlogischen Funktionen wurde in der Vorlesung mit  $2^{2^n}$  angegeben, es gibt daher  $2^{2^3} = 256$  dreistellige,  $2^{2^4} = 65\,536$  vierstellige und  $2^{2^5} = 4\,294\,967\,296$  fünfstellige aussagenlogische Funktionen.

Eine totale  $n$ -stellige Funktion über einem endlichen Wertebereich mit  $k$  Elementen ordnet jeder Wertebelegung der  $n$  Argumente eines der  $k$  Elemente als Ergebnis zu. Da es  $k^n$  Wertebelegungen gibt, gibt es  $k^{k^n}$  totale Funktionen. Bei partiellen Funktionen kann das Ergebnis für einzelne Wertebelegungen undefiniert sein; „undefiniert“ entspricht also einem weiteren Ergebniswert, es gibt somit  $(k+1)^{k^n}$  partielle Funktionen.

### Aufgabe 4 (0.4 Punkte)

- (a) Zeigen Sie, dass die Menge {not, or} vollständig ist für die Klasse der aussagenlogischen Funktionen.

- (b) Zeigen Sie, dass die Menge  $\{\text{iff}\}$  nicht vollständig ist.

## Lösung

- (a) Aus der Vorlesung wissen wir, dass die Menge  $\{\text{nand}\}$  funktional vollständig ist. Für die funktionale Vollständigkeit von  $\{\text{not}, \text{or}\}$  genügt es daher zu zeigen, dass **nand** durch die beiden Funktionen **not** und **or** dargestellt werden kann. In der Vorlesung wurde die Äquivalenz  $A \uparrow B = \neg A \vee \neg B$  behauptet. Wir zeigen daher mittels Auswertung in allen Wahrheitsbelegungen, dass  $x \text{ nand } y$  und  $(\text{not } x) \text{ or } (\text{not } y)$  identisch sind.

$x$	$y$	$x \text{ nand } y$	$=$	$(\text{not } x)$	$\text{or}$	$(\text{not } y)$
0	0	<b>1</b>	✓	1	<b>1</b>	1
0	1	<b>1</b>	✓	1	<b>1</b>	0
1	0	<b>1</b>	✓	0	<b>1</b>	1
1	1	<b>0</b>	✓	0	<b>0</b>	0

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

- (b) Es genügt von einer einzigen Funktion zu zeigen, dass sie nicht durch entsprechend zusammengesetzte **iff**'s darstellbar ist. Wir untersuchen zunächst, welche einstelligen Funktionen durch **iff** darstellbar sind in der Hoffnung, dass es nicht alle sind. Zu diesem Zweck müssen wir analysieren, welche Funktionen man ausgehend vom einzigen Argument  $x$  durch mehrfache Anwendung von **iff** erhalten kann. Wir stellen fest, dass  $\text{iff}(x, x) = 1$  gilt. Unter Berücksichtigung von 1 als weiterem Argument erhalten wir zusätzlich nur noch  $\text{iff}(1, x) = \text{iff}(x, 1) = x$  und  $\text{iff}(1, 1) = 1$ . Wenn wir also beweisen können, dass auch weitere Anwendungen der Funktion **iff** keine anderen einstelligen Funktionen liefern, haben wir gezeigt, dass die Menge  $\{\text{iff}\}$  nicht funktional vollständig ist, da etwa die einstellige Funktion **not** nicht dabei ist.

Wir beweisen daher folgende Behauptung mittels vollständiger Induktion.

*Induktionsbehauptung:* Jeder Ausdruck bestehend aus **iff** und  $x$  ist äquivalent zu  $x$  oder 1.

Der Beweis erfolgt induktiv nach der Anzahl  $n$  der Anwendungen der Funktion **iff**.

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

*Induktionshypothese:* Unsere Behauptung gelte für alle Ausdrücke mit  $n$  oder weniger Anwendungen von **iff**.

*Induktionsschritt:* Wir zeigen, dass unter der Annahme, dass die Induktionshypothese zutrifft, unsere Behauptung auch für Ausdrücke mit  $n + 1$  Anwendungen von **iff** gilt. Wir betrachten also einen Ausdruck  $\text{iff}(f(x), g(x))$ , der insgesamt  $n + 1$  Anwendungen von **iff** enthält. Die beiden Funktionen  $f$  und  $g$  müssen dann mit  $n$  oder weniger Anwendungen von **iff** definiert sein. Laut Hypothese ist jeder der beiden Ausdrücke äquivalent zu  $x$  oder 1. Wie wir oben aber festgestellt haben, liefert **iff**

mit den Argumenten  $x$  bzw. 1 wieder nur  $x$  oder 1. Damit gilt die Behauptung auch für den Fall  $n + 1$ .

Schlussfolgerung: Die einstellige Funktion  $\text{not}$  (ebenso übrigens wie die einstellige Funktion, die immer 0 liefert) ist nicht durch  $\text{iff}$  darstellbar, die Menge  $\{\text{iff}\}$  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  $\text{id}$  (definiert durch  $\text{id}(x) = x$  für alle  $x$ ) durch Anwendung von  $\text{iff}$  wieder nur  $\text{id}$  oder die einstellige Funktion  $\text{one}$  (definiert durch  $\text{one}(x) = 1$  für alle  $x$ ) entstehen kann. Mehr dazu unter dem Stichwort „Klon“ (engl. *clone*) im Internet oder in Büchern zur Algebra.

## Aufgabe 5 (0.4 Punkte)

- (a) Definieren Sie die Syntax arithmetischer Ausdrücke mit Hilfe einer induktiven Definition. Nehmen Sie an, dass die Ausdrücke nur aus den Variablen  $x, y$  und  $z$ , den Konstanten 0 bis 9, den Operatoren  $+$  und  $-$  (ein- und zweistellig), dem Operator  $*$  (zweistellig) sowie aus Klammern bestehen dürfen.

Beispiele für arithmetische Ausdrücke:  $((-x+2)*y)$ ,  $-(((1+2)-x)*(y*z))$

- (b) Definieren Sie die Semantik arithmetischer Ausdrücke, indem Sie eine Auswertungsfunktion  $\text{val}_I$  festlegen, die ausgehend von einer Interpretation  $I$  (weist den Variablen ganze Zahlen zu) jedem arithmetischen Ausdruck eine ganze Zahl als Wert zuordnet.

Beispiel: Für  $I(x) = 10$  und  $I(y) = 2$  liefert  $\text{val}_I((( -x+2)*y))$  den Wert  $-16$ .

## Lösung

Am einfachsten lassen sich Syntax und Semantik der arithmetischen Ausdrücke definieren, wenn wir wie bei der Aussagenlogik in der Vorlesung strikte Klammerung binärer Ausdrücke vorsehen. Sei  $I: \{x, y, z\} \mapsto \mathbb{Z}$  eine arithmetische Interpretation (Variablenbelegung), d.h., eine Abbildung der Variablen auf ganze Zahlen, und sei  $\mathcal{I}$  die Menge aller solchen Interpretationen.

- (a) Die Menge  $\mathcal{A}$  der arithmetischen Ausdrücke ist die kleinste Menge mit den folgenden Eigenschaften.

- $\{x, y, z\} \subseteq \mathcal{A}$
- $\{0, \dots, 9\} \subseteq \mathcal{A}$
- $+a, -a \in \mathcal{A}$  falls  $a \in \mathcal{A}$
- $(a + b), (a - b), (a * b) \in \mathcal{A}$  falls  $a, b \in \mathcal{A}$

- (b) Der Wert eines Ausdrucks in einer Interpretation  $I$  wird durch die Funktion  $\text{val}: \mathcal{I} \times \mathcal{A} \mapsto \mathbb{Z}$  festgelegt, die definiert ist durch:

- $\text{val}_I(a) = I(a)$ , falls  $a \in \{x, y, z\}$
- $\text{val}_I(0) = 0, \dots, \text{val}_I(9) = 9$
- $\text{val}_I(+a) = \text{val}_I(a)$   
 $\text{val}_I(-a) = -\text{val}_I(a)$
- $\text{val}_I((a + b)) = \text{val}_I(a) + \text{val}_I(b)$   
 $\text{val}_I((a - b)) = \text{val}_I(a) - \text{val}_I(b)$   
 $\text{val}_I((a * b)) = \text{val}_I(a) \cdot \text{val}_I(b)$

Will man auch Ausdrücke mit weniger oder mehr Klammern zulassen, kann man auf der Meta-Ebene Schreibvereinfachungen durch Festlegung von Prioritäten (etwa Punkt-vor-Strichrechnung) vereinbaren. Eine andere Möglichkeit besteht darin, die liberale Klammersetzung bereits in die Definition einzubauen, etwa so.

Die Menge  $\mathcal{A}'$  der arithmetischen Ausdrücke ist die kleinste Menge mit den folgenden Eigenschaften.

- $\{x, y, z\} \subseteq \mathcal{A}'$
- $\{0, \dots, 9\} \subseteq \mathcal{A}'$
- $+a, -a \in \mathcal{A}'$  falls  $a \in \mathcal{A}'$
- $a + b, a - b, a * b \in \mathcal{A}'$  falls  $a, b \in \mathcal{A}'$
- $(a) \in \mathcal{A}'$  falls  $a \in \mathcal{A}'$

Der Wert eines Ausdrucks in einer Interpretation  $I$  wird durch die Funktion  $\text{val}: \mathcal{I} \times \mathcal{A}' \mapsto \mathbb{Z}$  festgelegt, die definiert ist durch:

- $\text{val}_I(a) = I(a)$ , falls  $a \in \{x, y, z\}$
- $\text{val}_I(0) = 0, \dots, \text{val}_I(9) = 9$
- $\text{val}_I(+a) = \text{val}_I(a)$   
 $\text{val}_I(-a) = -\text{val}_I(a)$
- $\text{val}_I(a + b) = \text{val}_I(a) + \text{val}_I(b)$   
 $\text{val}_I(a - b) = \text{val}_I(a) - \text{val}_I(b)$   
 $\text{val}_I(a * b) = \text{val}_I(a) \cdot \text{val}_I(b)$
- $\text{val}_I((a)) = \text{val}_I(a)$

Die Menge  $\mathcal{A}'$  ist eine echte Übermenge von  $\mathcal{A}$  und enthält auch Ausdrücke der Form  $1+2*3$ ,  $(1+2)*3$ ,  $1+(2*3)$  oder  $((x))$ , was ja beabsichtigt war. Eine Eigenschaft dieser Definition von  $\mathcal{A}'$  ist, dass manche Elemente darin auf verschiedene Arten hergeleitet werden können. So kann etwa der Ausdruck  $1+2*3$  aus 1 und  $2*3$  mit Hilfe von  $+$  aufgebaut werden, aber auch aus  $1+2$  und 3 mit Hilfe von  $*$ . Diese Mehrdeutigkeit stellt kein Problem für die Definition der Syntax dar, sehr wohl aber für jene der Semantik, denn: **val ist keine wohldefinierte Funktion!** Das sieht man beispielsweise, wenn man versucht, den Wert von  $1+2*3$  zu bestimmen. Einerseits gilt

$$\text{val}_I(1+2*3) = \text{val}_I(1) + \text{val}_I(2*3) = \text{val}_I(1) + \text{val}_I(2) \cdot \text{val}_I(3) = 1 + 2 \cdot 3 = 7$$

andererseits gilt aber auch

$$\text{val}_I(1+2*3) = \text{val}_I(1+2) \cdot \text{val}_I(3) = (\text{val}_I(1) + \text{val}_I(2)) \cdot \text{val}_I(3) = (1 + 2) \cdot 3 = 9.$$

$\mathcal{A}'$  kann allerdings etwas aufwändiger mit Hilfsmengen  $\mathcal{T}$  (Terme) und  $\mathcal{F}$  (Faktoren) so definiert werden, dass es keine Mehrdeutigkeiten mehr gibt. Dann ist auch die davon abgeleitete Auswertungsfunktion wieder wohldefiniert. In den folgenden Definitionen bezeichnet  $a$  einen Ausdruck ( $a \in \mathcal{A}'$ ),  $t$  einen Term ( $t \in \mathcal{T}$ ) und  $f$  einen Faktor ( $f \in \mathcal{F}$ ).

(a) Die Menge  $\mathcal{A}'$  der arithmetischen Ausdrücke ist die kleinste Menge, sodass:

- $t \in \mathcal{A}'$  falls  $t \in \mathcal{T}$  (d.h.,  $\mathcal{T} \subseteq \mathcal{A}'$ )
- $a+t, a-t \in \mathcal{A}'$  falls  $a \in \mathcal{A}'$  und  $t \in \mathcal{T}$

(b) Die Auswertungsfunktion  $\text{val}: \mathcal{I} \times \mathcal{A}' \mapsto \mathbb{Z}$  für Ausdrücke ist definiert durch:

- $\text{val}_I(t) = \text{val}_I^{\mathcal{T}}(t)$
- $\text{val}_I(a+t) = \text{val}_I(a) + \text{val}_I^{\mathcal{T}}(t)$   
 $\text{val}_I(a-t) = \text{val}_I(a) - \text{val}_I^{\mathcal{T}}(t)$

Die Menge  $\mathcal{T}$  der arithmetischen Terme ist die kleinste Menge, sodass:

- $f \in \mathcal{T}$  falls  $f \in \mathcal{F}$  (d.h.,  $\mathcal{F} \subseteq \mathcal{T}$ )
- $t * f \in \mathcal{T}$  falls  $t \in \mathcal{T}$  und  $f \in \mathcal{F}$

Die Menge  $\mathcal{F}$  der arithmetischen Faktoren ist die kleinste Menge, sodass:

- $\{x, y, z\} \subseteq \mathcal{F}$
- $\{0, \dots, 9\} \subseteq \mathcal{F}$
- $+f, -f \in \mathcal{F}$  falls  $f \in \mathcal{F}$
- $(a) \in \mathcal{F}$  falls  $a \in \mathcal{A}'$

Die Auswertungsfunktion  $\text{val}^{\mathcal{T}}: \mathcal{I} \times \mathcal{T} \mapsto \mathbb{Z}$  für Terme ist definiert durch:

- $\text{val}^{\mathcal{T}}(f) = \text{val}^{\mathcal{F}}(f)$
- $\text{val}^{\mathcal{T}}(t * f) = \text{val}^{\mathcal{T}}(t) \cdot \text{val}^{\mathcal{F}}(f)$

Die Auswertungsfunktion  $\text{val}^{\mathcal{F}}: \mathcal{I} \times \mathcal{F} \mapsto \mathbb{Z}$  für Faktoren ist definiert durch:

- $\text{val}^{\mathcal{F}}(f) = I(f)$  falls  $f \in \{x, y, z\}$
- $\text{val}^{\mathcal{F}}(0) = 0, \dots, \text{val}^{\mathcal{F}}(9) = 9$
- $\text{val}^{\mathcal{F}}(+a) = \text{val}^{\mathcal{F}}(a)$   
 $\text{val}^{\mathcal{F}}(-a) = -\text{val}^{\mathcal{F}}(a)$
- $\text{val}^{\mathcal{F}}((a)) = \text{val}_I(a)$

In der zweiten Definition von  $\mathcal{A}'$  besitzen Ausdrücke wie  $1+2*3$  nur noch eine Zerlegung:  $1+2*3$  ist ein Ausdruck, weil  $1$  ein Ausdruck und  $2*3$  ein Term ist. Der Versuch  $1+2*3$  aus  $1+2$  und  $3$  aufzubauen scheitert, da  $*$  auf der linken Seite einen Term benötigt,  $1+2$  aber keiner ist. Der Wert des Ausdrucks ist nun eindeutig definiert:

$$\begin{aligned} \text{val}_I(1+2*3) &= \text{val}_I(1) + \text{val}_I^{\mathcal{T}}(2*3) = \text{val}_I^{\mathcal{T}}(1) + \text{val}_I^{\mathcal{T}}(2) \cdot \text{val}_I^{\mathcal{F}}(3) \\ &= \text{val}_I^{\mathcal{F}}(1) + \text{val}_I^{\mathcal{F}}(2) \cdot \text{val}_I^{\mathcal{F}}(3) = 1 + 2 \cdot 3 = 7 \end{aligned}$$

Eine andere Berechnung ist nicht möglich, da laut Definition zwar

$$\text{val}_I(1+2*3) = \text{val}_I^{\mathcal{T}}(1+2*3) = \text{val}_I^{\mathcal{T}}(1+2) \cdot \text{val}_I^{\mathcal{F}}(3) = \text{val}_I^{\mathcal{F}}(1+2) \cdot 3$$

gilt,  $\text{val}_I^{\mathcal{F}}(1+2)$  aber undefiniert ist.

## Aufgabe 6 (0.3 Punkte)

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

- Zeigen Sie, dass  $F$  syntaktisch korrekt ist.
- Werten Sie  $\text{val}_I(F)$  für  $I(A) = 1$ ,  $I(B) = 0$  und  $I(C) = 0$  schrittweise aus.
- Verwenden Sie eine Wahrheitstafel um festzustellen, ob die Formel  $F$  gültig, erfüllbar, widerlegbar und/oder unerfüllbar ist.

## Lösung

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

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

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



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

(a4)  $(F * G) \in \mathcal{A}$ , wenn  $F, G \in \mathcal{A}$  und  $*$   $\in \{\wedge, \uparrow, \vee, \downarrow, \equiv, \neq, \supset, \subset\}$ .

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

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

- Die Variablen **A**, **B** und **C** sind Formeln (a1).
- Da **A** und **B** Formeln sind, ist auch  $(A \vee B)$  eine Formel (a4).
- Da  $(A \vee B)$  sowie **C** Formeln sind, ist auch  $((A \vee B) \equiv C)$  eine Formel (a4).
- $\neg A$  ist eine Formel, da **A** eine Formel ist (a3).
- Da  $((A \vee B) \equiv C)$  sowie  $\neg A$  Formeln sind, ist auch  $((A \vee B) \equiv C) \supset \neg A$  eine Formel.

$$\begin{aligned} \text{(b) } \text{val}_I(((A \vee B) \equiv C) \supset \neg A) &= \text{val}_I((A \vee B) \equiv C) \text{ implies } \text{val}_I(\neg A) \\ &= (\text{val}_I(A \vee B) \text{ iff } \text{val}_I(C)) \text{ implies not } \text{val}_I(A) \\ &= ((\text{val}_I(A) \text{ or } \text{val}_I(B)) \text{ iff } 0) \text{ implies not } 1 \\ &= ((1 \text{ or } 0) \text{ iff } 0) \text{ implies } 0 \\ &= (1 \text{ iff } 0) \text{ implies } 0 \\ &= 0 \text{ implies } 0 \\ &= 1 \end{aligned}$$

(c) Die Formel  $F$  ist erfüllbar, da es eine Interpretation  $I$  gibt, in der sie wahr ist, etwa  $I(A) = I(B) = I(C) = 0$ . Sie ist auch widerlegbar, da es eine Interpretation  $I$  gibt, in der sie falsch ist, etwa  $I(A) = I(B) = I(C) = 1$ . Diese Interpretationen lassen sich systematisch mittels der Wahrheitstafel finden:

$A$	$B$	$C$	$((A \vee B) \equiv C) \supset \neg A$		
1	1	1	1	1	<b>0</b> 0
1	1	0	1	0	<b>1</b> 0
1	0	1	1	1	<b>0</b> 0
1	0	0	1	0	<b>1</b> 0
0	1	1	1	1	<b>1</b> 1
0	1	0	1	0	<b>1</b> 1
0	0	1	0	0	<b>1</b> 1
0	0	0	0	1	<b>1</b> 1

### Aufgabe 7 (0.3 Punkte)

Zeigen Sie, dass die beiden Formeln  $(A \wedge C) \vee (A \wedge \neg B)$  und  $(A \supset B) \supset (C \wedge A)$  äquivalent sind

- mithilfe einer Wahrheitstafel;
- durch algebraische Umformungen.

## Lösung

(a) Wahrheitstafel:

$A$	$B$	$C$	$(A \wedge C) \vee (A \wedge \neg B)$			$=$	$(A \supset B) \supset (C \wedge A)$			
0	0	0	0	0	0	1	✓	1	0	0
0	0	1	0	0	0	1	✓	1	0	0
0	1	0	0	0	0	0	✓	1	0	0
0	1	1	0	0	0	0	✓	1	0	0
1	0	0	0	1	1	1	✓	0	1	0
1	0	1	1	1	1	1	✓	0	1	1
1	1	0	0	0	0	0	✓	1	0	0
1	1	1	1	1	0	0	✓	1	1	1

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

(b) Wir bringen die zweite Formel in disjunktive Normalform (die erste ist es bereits). Da wir dabei identische Formeln erhalten, sind die ursprünglichen Formeln äquivalent.

$$\begin{aligned}
 (A \supset B) \supset (C \wedge A) &= \neg(\neg A \vee B) \vee (C \wedge A) && \text{Ersetzen von } \supset \text{ durch } \neg, \vee \\
 &= (A \wedge \neg B) \vee (C \wedge A) && \text{De Morgan} \\
 &= (A \wedge C) \vee (A \wedge \neg B) && \text{Kommutativität}
 \end{aligned}$$

## Aufgabe 8 (0.3 Punkte)

Ist die Formel  $A$  eine logische Konsequenz der drei Formeln  $A \wedge \neg B$ ,  $B \supset C$  und  $(C \vee B) \supset A$ ? Geben Sie eine einzelne Formel an, die genau dann gültig ist, wenn diese Konsequenzbeziehung gilt.

## Lösung

$I(A)$	$I(B)$	$I(C)$	$A \wedge \neg B, B \supset C, (C \vee B) \supset A$			$\models_I$	$A$
0	0	0	0	1	1	✓	0
0	0	1	0	1	0	✓	0
0	1	0	0	0	0	✓	0
0	1	1	0	1	0	✓	0
1	0	0	1	1	1	✓	1
1	0	1	1	1	1	✓	1
1	1	0	0	0	1	✓	1
1	1	1	0	1	1	✓	1

Die Formel  $A$  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 links nach rechts aus, ergibt sich folgende vereinfachte Tabelle:

$I(A)$	$I(B)$	$I(C)$	$A \wedge \neg B,$	$B \supset C,$	$(C \vee B) \supset A$	$\models_I$	$A$
0	0	0	0			✓	
0	0	1	0			✓	
0	1	0	0			✓	
0	1	1	0			✓	
1	0	0	1	1	1	✓	1
1	0	1	1	1	1	✓	1
1	1	0	0			✓	
1	1	1	0			✓	

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

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

gültig ist.

## Aufgabe 9 (0.2 Punkte)

Sei  $f$  folgende dreistellige Funktion.

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

Stellen Sie  $f$  durch eine Formel in

- (a) disjunktiver
- (b) konjunktiver

Normalform dar.

## Lösung

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

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

## Aufgabe 10 (0.3 Punkte)

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

(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:

$A$	$B$	$C$	$\neg(B \downarrow (A \wedge C)) \wedge (C \vee (\neg A \neq B))$						
0	0	0	0	1	0	<b>0</b>	1	1	1
0	0	1	0	1	0	<b>0</b>	1	1	1
0	1	0	1	0	0	<b>0</b>	0	1	0
0	1	1	1	0	0	<b>1</b>	1	1	0
1	0	0	0	1	0	<b>0</b>	0	0	0
1	0	1	1	0	1	<b>1</b>	1	0	0
1	1	0	1	0	0	<b>1</b>	1	0	1
1	1	1	1	0	1	<b>1</b>	1	0	1

Aus dieser Tafel lässt sich folgende DNF ablesen:

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

Die DNF lässt sich noch zu  $(A \wedge B) \vee (A \wedge C) \vee (B \wedge C)$  vereinfachen.

(b) KNF mittels algebraischer Methode:

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

(Jede der letzten drei Formeln ist in KNF und damit eine Lösung.)

## Aufgabe 11 (0.4 Punkte)

Der Frühling naht und Tanja möchte in ihrem Garten Blumen pflanzen. Sie stellt die folgenden Überlegungen an.

- Ich will auf jeden Fall Nelken pflanzen.
  - Meiner Mutter zuliebe brauche ich Sonnenblumen oder Veilchen; vielleicht habe ich sogar Platz für beide.
  - Rosen und Veilchen möchte ich jedenfalls nicht gleichzeitig im Garten haben, das sieht nicht hübsch aus.
  - Ich pflanze Nelken nur dann, wenn ich Tulpen gesetzt habe.
  - Wenn ich Rosen pflanze, dann besteht meine Mutter darauf, dass ich auch Tulpen und Veilchen pflanze.
  - Wenn ich Veilchen pflanze, dann auch Sonnenblumen.
- (a) Formalisieren Sie die beschriebene Situation inklusive aller Anhaltspunkte mittels aussagenlogischer Formeln. Geben Sie die Bedeutung der von Ihnen verwendeten Aussagenvariablen an.
- (b) Welche Blumen pflanzt Tanja? Begründen Sie die Antwort mit Hilfe Ihrer aussagenlogischen Modellierung.

## Lösung

- (a) Aussagenvariablen und ihre Bedeutung:

$R$  ... Tanja pflanzt Rosen.  
 $T$  ... Tanja pflanzt Tulpen.  
 $N$  ... Tanja pflanzt Nelken.  
 $S$  ... Tanja pflanzt Sonnenblumen.  
 $V$  ... Tanja pflanzt Veilchen.

Aussagenlogische Formeln:

$F_0 := N$	auf jeden Fall Nelken
$F_1 := S \vee V$	Sonnenblumen oder Veilchen
$F_2 := \neg(R \wedge V)$	Rosen und Veilchen nicht gleichzeitig
$F_3 := N \supset T$	wenn Nelken dann Tulpen; Nelken nur, wenn Tulpen
$F_4 := R \supset (T \wedge V)$	wenn Rosen dann Tulpen und Veilchen
$F_5 := V \supset S$	wenn Veilchen dann Sonnenblumen

- (b) Wir suchen alle Wahrheitsbelegungen für die Variablen  $R, T, N, S,$  und  $V$ , sodass die Formeln  $F_0, \dots, F_5$  wahr werden. Wegen Formel  $F_0$  genügt es jene Belegungen zu betrachten, in denen  $N$  wahr ist.

$R$	$T$	$N$	$S$	$V$	$S \vee V$	$\neg(R \wedge V)$	$N \supset T$	$R \supset (T \wedge V)$	$V \supset S$	
0	0	1	0	0	0					
0	0	1	0	1	1	1	0			
0	0	1	1	0	1	1	0			
0	0	1	1	1	1	1	0			
0	1	1	0	0	0					
0	1	1	0	1	1	1	1	1	0	
0	1	1	1	0	1	1	1	1	1	✓
0	1	1	1	1	1	1	1	1	1	✓
1	0	1	0	0	0					
1	0	1	0	1	1	0				
1	0	1	1	0	1	1	0			
1	0	1	1	1	1	0				
1	1	1	0	0	0					
1	1	1	0	1	1	0				
1	1	1	1	0	1	1	1	0		
1	1	1	1	1	1	0				

Tanja pflanzt Tulpen, Nelken und Sonnenblumen sowie vielleicht auch noch Veilchen.

### Aufgabe 12 (0.3 Punkte)

Angenommen Sie modellieren eine Situation wie in der letzten Aufgabe mit Hilfe mehrerer aussagenlogischer Formeln  $F_1, \dots, F_m$  und ihre Kollegin beschreibt dieselbe Situation mit den aussagenlogischen Formeln  $G_1, \dots, G_n$ . (Die Anzahl der Formeln muss nicht gleich sein.) Wie können Sie mit Hilfe eines SAT-Solvers überprüfen, ob die beiden Beschreibungen gleichwertig (semantisch äquivalent) sind und dieselben Lösungen liefern? Beschreiben Sie alle erforderlichen Schritte und geben Sie ein Beispiel an, das diese Schritte illustriert. Was bedeutet es, wenn der SAT-Solver eine erfüllende Variablenbelegung findet?

### Lösung

Die Formeln  $F_1, \dots, F_m$  einerseits und die Formeln  $G_1, \dots, G_n$  andererseits sind gleichwertige Beschreibungen, wenn sie von denselben Interpretationen erfüllt bzw. nicht erfüllt werden, d.h., wenn  $F = F_1 \wedge \dots \wedge F_m$  und  $G = G_1 \wedge \dots \wedge G_n$  semantisch äquivalent sind. Das ist genau dann der Fall, wenn die Formel  $F \equiv G$  gültig bzw. die Formel  $F \not\equiv G$  unerfüllbar ist. Da SAT-Solver als Eingabe Formeln in KNF erwarten, muss  $F \not\equiv G$  in KNF umgewandelt werden; sei  $H$  diese Normalform. Stellt der SAT-Solver die

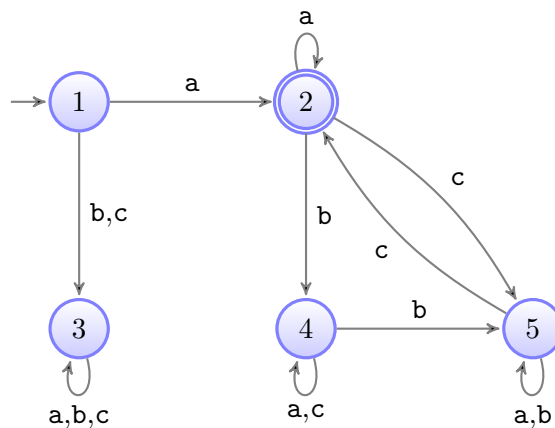
Unerfüllbarkeit von  $H$  fest, waren die beiden ursprünglichen Beschreibungen gleichwertig. Ist  $H$  erfüllbar, gibt die vom SAT-Solver berechnete erfüllende Interpretation eine Variablenbelegung an, für die die beiden Beschreibungen unterschiedliche Werte liefern. Als Beispiel betrachten wir die Formel  $F_1 = A \vee B$  einerseits und die Formeln  $G_1 = B \vee C$  und  $G_2 = C \supset A$  andererseits. Wir wandeln  $F_1 \not\equiv (G_1 \wedge G_2)$  in konjunktive Normalform um.

$$\begin{aligned}
 F_1 &\not\equiv (G_1 \wedge G_2) \\
 &= (A \vee B) \not\equiv ((B \vee C) \wedge (C \supset A)) \\
 &= ((A \vee B) \vee ((B \vee C) \wedge (\neg C \vee A))) \wedge (\neg(A \vee B) \vee \neg((B \vee C) \wedge (\neg C \vee A))) \\
 &= (A \vee B \vee C) \wedge (A \vee B \vee \neg C) \wedge ((\neg A \wedge \neg B) \vee (\neg(B \vee C) \vee \neg(\neg C \vee A))) \\
 &= (A \vee B \vee C) \wedge (A \vee B \vee \neg C) \wedge ((\neg A \wedge \neg B) \vee (\neg B \wedge \neg C) \vee (C \wedge \neg A)) \\
 &= (A \vee B \vee C) \wedge (A \vee B \vee \neg C) \wedge (\neg A \vee \neg B) \wedge (\neg A \vee \neg C) \wedge (\neg B \vee C)
 \end{aligned}$$

Ein SAT-Solver liefert für diese Formel die Antwort „erfüllbar“ zusammen mit beispielsweise der Wahrheitsbelegung  $I(A) = 1$  und  $I(B) = I(C) = 0$ . Das bedeutet, dass  $F_1$  einerseits und  $G_1, G_2$  andererseits keine gleichwertigen Beschreibungen sind und sich z.B. bei der angegebenen Wertebelegung unterscheiden. Tatsächlich ist die Formel  $F_1$  in  $I$  wahr,  $G_1$  hingegen falsch.

### Aufgabe 13 (0.4 Punkte)

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



- Geben Sie 5 Wörter an, die von  $\mathcal{A}$  akzeptiert werden.
- Geben Sie an, welche der folgenden Wörter der Automat akzeptiert:  $\varepsilon$ ,  $a$ ,  $abba$ ,  $acca$ ,  $abc$ .
- Berechnen Sie schrittweise  $\delta^*(1, acbca)$ .
- Spezifizieren Sie  $\mathcal{A}$  in tabellarischer Form. Handelt es sich bei  $\mathcal{A}$  um einen deterministischen oder indeterministischen Automaten?

## Lösung

(a)  $\mathcal{A}$  akzeptiert zum Beispiel  $a$ ,  $aa$ ,  $acc$ ,  $abbc$  und  $acca$ .

(b)  $\mathcal{A}$  akzeptiert  $a$  und  $acca$ , nicht aber  $\varepsilon$ ,  $abba$  und  $abc$ .

$$\begin{aligned} \text{(c) } \delta^*(1, acbca) &= \delta^*(\delta(1, a), cbca) \\ &= \delta^*(2, cbca) \\ &= \delta^*(\delta(2, c), bca) \\ &= \delta^*(5, bca) \\ &= \delta^*(\delta(5, b), ca) \\ &= \delta^*(5, ca) \\ &= \delta^*(\delta(5, c), a) \\ &= \delta^*(2, a) \\ &= \delta^*(\delta(2, a), \varepsilon) \\ &= \delta^*(2, \varepsilon) \\ &= 2 \end{aligned}$$

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

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

$\mathcal{A}$  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 (0.4 Punkte)

Ein Getränkeautomat bietet drei Getränkearten zum Verkauf an: Wasser um 1 €, Cola um 1,50 € und Orangensaft um 2 €. Er akzeptiert 50-Cent-, 1-Euro- und 2-Euro-Münzen; andere Münzen fallen durch. Sobald der eingeworfene Betrag 2 € oder mehr ausmacht, blockiert der Automat den Einwurfschlitz und verhindert damit den Einwurf weiterer Münzen. Der Automat besitzt vier Knöpfe. Der Stornoknopf bewirkt, dass der Automat das bereits eingeworfene Geld zurückgibt und in den Grundzustand zurückkehrt. Die anderen drei Knöpfe sind für die drei Getränkearten. Wird einer dieser Knöpfe gedrückt, ehe ausreichend Geld für das entsprechende Getränk eingeworfen wurde, passiert nichts. Andernfalls wirft der Automat das Getränk aus; Retourgeld wird nicht zurückgegeben. Modellieren Sie den Getränkeautomaten mit Hilfe eines endlichen Automaten. Die Sprache des endlichen Automaten soll aus genau jenen Aktionsfolgen bestehen, die letztendlich zur Ausgabe eines Getränkes führen. Beginnen Sie die Aufgabe damit, dass Sie sich



überlegen, welche Zustände und welche Aktionen diesen Getränkeautomaten kennzeichnen.

## Lösung

*Zustände:* Der Automat muss sich bis zu einem Betrag von 2€ merken, wieviel Geld bereits eingeworfen wurde. Wir führen daher Zustände mit den Bezeichnungen 0, 0.5, 1, 1.5 und  $\geq 2$  ein, die für einen eingeworfenen Betrag von 0€, 0.50€, 1€, 1.50€ bzw. mindestens 2€ stehen. Weiters benötigen wir einen Endzustand „Ende“, der anzeigt, dass das Getränk ausgegeben wurde. Zum Fehlerzustand „Fehler“ führen alle Übergänge, die eigentlich aufgrund der Bauweise des Automaten nicht vorkommen können. Etwa akzeptiert der Automat im Normalbetrieb kein Geld mehr, wenn bereits zwei Euro oder mehr eingeworfen wurden. Weiters führen vom Endzustand alle Übergänge in den Fehlerzustand, da der Automat laut Angabe nur jene Aktionsfolgen akzeptieren soll, die zur Ausgabe *eines* Getränkes führen. Wir erhalten somit als Zustandsmenge

$$Q = \{0, 0.5, 1, 1.5, \geq 2, \text{Ende}, \text{Fehler}\} .$$

Startzustand ist 0, die Menge der Endzustände ist  $F = \{\text{Ende}\}$ .

*Aktionen (Alphabet):* Zustandsübergänge werden durch den Einwurf einer Münze oder das Drücken einer Taste ausgelöst. Wir wählen das Alphabet  $\Sigma = \{0.5, 1, 2, S, W, C, 0\}$  mit folgender Interpretation:

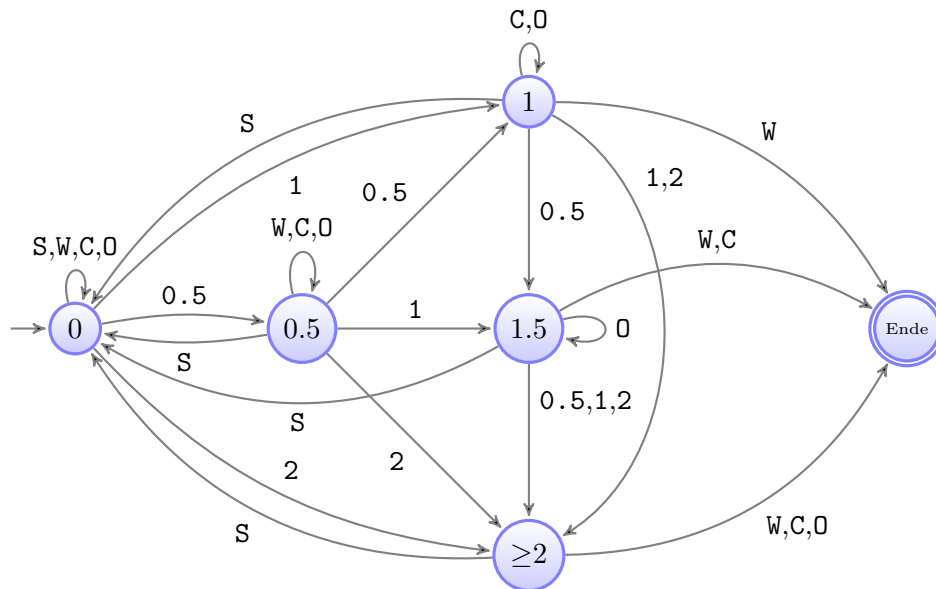
0.5/1/2 ... Eine 50 Cent-/1 Euro-/2 Euro-Münze wird eingeworfen.

S/W/C/0 ... Der Storno-/Wasser-/Cola-/Orangensaftknopf wird gedrückt.

*Automat:* Das Verhalten des Getränkeautomaten wird durch den Automaten  $\langle Q, \Sigma, \delta, 0, F \rangle$  beschrieben, wobei  $Q$ ,  $\Sigma$  und  $F$  oben definiert wurden und die Übergangsfunktion  $\delta$  durch folgende Tabelle festgelegt wird.

	0.5	1	2	S	W	C	0
0	0.5	1	$\geq 2$	0	0	0	0
0.5	1	1.5	$\geq 2$	0	0.5	0.5	0.5
1	1.5	$\geq 2$	$\geq 2$	0	Ende	1	1
1.5	$\geq 2$	$\geq 2$	$\geq 2$	0	Ende	Ende	1.5
2	Fehler	Fehler	Fehler	0	Ende	Ende	Ende
Ende	Fehler	Fehler	Fehler	Fehler	Fehler	Fehler	Fehler
Fehler	Fehler	Fehler	Fehler	Fehler	Fehler	Fehler	Fehler

Alternativ kann der Automat grafisch dargestellt werden (wir zeichnen den Fehlerzustand nicht ein):



### Aufgabe 15 (0.4 Punkte)

Ein zweistöckiges Haus (Erdgeschoß, 1. und 2. Stock) besitzt einen Aufzug. In jedem Stockwerk gibt es einen Knopf, um den Aufzug zu rufen. In der Aufzugskabine kann das gewünschte Ziel mit einem von drei Knöpfen angegeben werden. Für die Aufzugssteuerung ist es gleichbedeutend, ob im Erdgeschoß der Knopf außen oder in der Kabine der Erdgeschoß-Knopf gedrückt wird; analog für die anderen Knöpfe innen und außen. Die Aufzugssteuerung ist in der Lage sich mehrere Aufträge zu merken. Werden etwa im Erdgeschoß in der Kabine nacheinander die Knöpfe für 2. und 1. Stock (oder umgekehrt) gedrückt, steuert die Kabine zuerst den 1. und dann den 2. Stock an (die Steuerung optimiert die Aufträge). Aufträge, die das Stockwerk betreffen, in dem sich die Kabine bereits befindet, werden ignoriert. Liegt kein Auftrag vor, bleibt die Kabine im zuletzt gewählten Stockwerk. Wird die Anlage eingeschaltet, wartet die Kabine im Erdgeschoß auf den ersten Auftrag.

Zusätzlich zu den drei Signalen, die den Stockwerken entsprechen, verarbeitet die Steuerung noch das Signal „Türe schließt“. Dieses wird automatisch einige Sekunden nach Betätigen der Tasten und Freiwerden der Tür generiert und bewirkt, dass die Türe schließt und die Kabine in das nächstliegende Stockwerk fährt, für das ein Auftrag vorliegt. Gibt es keinen Auftrag, wird das Signal ignoriert. Wird nach „Türe schließt“ noch eine Stockwerktaaste betätigt, beeinflusst das die Zielwahl nicht mehr; die Taste wird so behandelt, als wäre sie nach Ankunft im Zielstockwerk gedrückt worden. Liegen in einem Stockwerk Aufträge für Ziele in entgegengesetzten Richtungen vor, wird zuerst der ältere Auftrag ausgeführt.

*Aufgabe:* Beschreiben Sie das Verhalten der Aufzugssteuerung durch einen endlichen Automaten. Der Automat befindet sich in einem Endzustand, wenn alle Aufträge aus-

geführt wurden und der Aufzug auf einen neuen wartet. Ihr Automat soll selber keine Signale generieren sondern nur das Verhalten der Steuerung beschreiben; es ist also kein Transducer gefragt.

Beginnen Sie damit zu definieren, welche Zustände das System einnehmen kann und welche Aktionen zu Übergängen führen.

*Beispiel:* Die Kabine befinde sich im Erdgeschoß. Es werden nacheinander die Tasten „2.Stock“, „Erdgeschoß“ und „1.Stock“ gedrückt (entweder in der Kabine oder außen im jeweiligen Stockwerk), ehe die „Türe schließt“. Der Auftrag „Erdgeschoß“ wird ignoriert, da sich die Kabine bereits dort befindet. Das erste Ziel ist der 1.Stock. Während des Aufenthalts dort (oder auf der Fahrt dorthin) ruft jemand im Erdgeschoß erneut den Aufzug. Die Kabine fährt dennoch zuerst in den 2.Stock, da dies der ältere Auftrag ist und das Erdgeschoß in entgegengesetzter Richtung liegt. Erst danach fährt die Kabine in das Erdgeschoß. Wenn E, 1, 2 und T die jeweiligen Signale bezeichnen, liegt somit das Wort 2E1TETT in der Sprache, die der Automat beschreibt.

## Lösung

*Zustände:* Zur Beschreibung des momentanen Zustands des Aufzugsystems sind folgende Informationen erforderlich:

- der momentane Aufenthaltsort der Kabine (Erdgeschoß, 1. oder 2. Stock),
- die vorliegenden Aufträge, die durch Drücken einer oder mehrerer Stockwerkstasten innen oder außen erteilt wurden, sowie ihre Reihenfolge.

Offenbar können Aufträge für jenes Stockwerk, in dem sich die Kabine bereits befindet, ignoriert werden. Außerdem ist die Reihenfolge der Aufträge nur im 1.Stock relevant, da nur dort eine Entscheidung zwischen Auf- und Abwärtsfahrt getroffen werden kann.

Wir bezeichnen jeden Zustand mit dem Stockwerk, in dem sich der Aufzug befindet, sowie mit der Liste der Aufträge als Index. Beispielsweise ist  $1e_2$  die Beschreibung für jenen Zustand, in dem sich die Kabine im 1.Stock befindet und als nächstes ins Erdgeschoß fahren wird; zusätzlich ist der 2.Stock als Auftrag vorgemerkt. Da im Erdgeschoß und im 2.Stock die Reihenfolge der Aufträge irrelevant ist, sind die Zustände  $e_{12}$  und  $e_{21}$  bzw.  $2e_1$  und  $2_1e$  gleichwertig, sodass wir nur jeweils einen davon benötigen. Man kann sich leicht überlegen (oder merkt es spätestens bei der Konstruktion des Automaten), dass die folgenden Zustände relevant sind:

$$Q = \{e, e_1, e_2, e_{12}, 1, 1_e, 1_{e2}, 1_2, 1_{2e}, 2, 2_e, 2_1, 2_{1e}\}$$

Startzustand ist laut Angabe jener Zustand, in dem sich die Kabine im Erdgeschoß befindet und kein Auftrag vorliegt, also der Zustand  $e$ . Endzustände sind alle Zustände, in denen keine Aufträge vorliegen:  $F = \{e, 1, 2\}$ .

*Aktionen (Alphabet):* Zustandsänderungen werden durch das Drücken eines Knopfes sowie durch das „Türe schließt“-Signal ausgelöst. Wir wählen  $\Sigma = \{E, 1, 2, T\}$ , wobei E, 1 und 2 bedeuten, dass innen oder außen die Taste für das Erdgeschoß, für den 1. bzw.

den 2. Stock gedrückt wird. T symbolisiert das Signal zum Schließen der Türen und zum Stochwerkwechsel, das nach einer gewissen Zeit nach dem Drücken einer oder mehrerer der Tasten automatisch von einer Steuereinheit generiert wird.

*Automat:* Das Verhalten des Aufzugssteuerung wird durch den Automaten  $\langle Q, \Sigma, \delta, e, F \rangle$  beschrieben ( $Q$ ,  $\Sigma$  und  $F$  siehe oben), wobei die Übergangsfunktion  $\delta$  durch folgende Tabelle definiert wird.

	E	1	2	T
$e$	$e$	$e_1$	$e_2$	$e$
$e_1$	$e_1$	$e_1$	$e_{12}$	1
$e_2$	$e_2$	$e_{12}$	$e_2$	2
$e_{12}$	$e_{12}$	$e_{12}$	$e_{12}$	$1_2$
1	$1_e$	1	$1_2$	1
$1_2$	$1_{2e}$	$1_2$	$1_2$	2
$1_{2e}$	$1_{2e}$	$1_{2e}$	$1_{2e}$	$2_e$
$1_e$	$1_e$	$1_e$	$1_{e2}$	$e$
$1_{e2}$	$1_{e2}$	$1_{e2}$	$1_{e2}$	$e_2$
2	$2_e$	$2_1$	2	2
$2_e$	$2_e$	$2_{1e}$	$2_e$	$e$
$2_1$	$2_{1e}$	$2_1$	$2_1$	1
$2_{1e}$	$2_{1e}$	$2_{1e}$	$2_{1e}$	$1_e$

Alternativ kann der Automat grafisch dargestellt werden:

