

Übung zu Logik

Bearbeiten Sie diese Aufgaben bis zur nächsten Übung am Freitag, 2. November, um 12 Uhr. Die Lösungen werden in der Übung besprochen.

Aufgabe 1 Folgerung und Äquivalenz

Untenstehend sehen Sie drei Paare von Formeln F_1, F_2 . Überlegen Sie jeweils, ob $F_1 \models F_2$, $F_2 \models F_1$ oder beides gilt.

(a) $F_1 = (A \rightarrow B) \wedge (B \rightarrow C)$
 $F_2 = A \rightarrow C$

(b) $F_1 = (A \rightarrow B) \wedge (B \rightarrow \neg A)$
 $F_2 = A \vee \neg A$

(c) $F_1 = (A \rightarrow B) \rightarrow (C \vee \neg D)$
 $F_2 = (A \wedge \neg B) \vee (D \rightarrow C)$

Begründen Sie Ihre Antworten mit den Mitteln der Aussagenlogik, z.B. mit Wahrheitstafeln, den in der Vorlesung vorgestellten Äquivalenzgesetzen oder dem Ersetzbarkeitstheorem.

Aufgabe 1 (Lösungsvorschlag) Folgerung und Äquivalenz

Alle drei Teilaufgaben lassen sich natürlich mithilfe von Wahrheitstabellen lösen. Wenn in jeder Zeile der Wahrheitstabelle, in der sich für F_1 eine 1 ergibt, auch für F_2 eine 1 steht, dann gilt $F_1 \models F_2$, umgekehrt analog.

Außerdem kann man Limboole zur Lösung der Aufgabe heranziehen: $F_1 \models F_2$ gilt genau dann, wenn $F_1 \rightarrow F_2$ von Limboole als gültige Formel erkannt wird.

Die Vorschläge unten beschränken sich deshalb auf Beweise mittels Äquivalenzumformungen (d.h. Anwendung des Ersetzbarkeitstheorems).

(a) Es gilt $F_1 \models F_2$:

$$\begin{aligned} F_1 &= (A \rightarrow B) \wedge (B \rightarrow C) \\ &\equiv (\neg A \vee B) \wedge (\neg B \vee C) && \text{(Def. von } \rightarrow \text{)} \\ &\equiv (\neg A \wedge (\neg B \vee C)) \vee (B \wedge \neg B) \vee (B \wedge C) && \text{(zweimal Distributivgesetz)} \end{aligned}$$

Man sieht sofort, dass $B \wedge \neg B$ unerfüllbar ist und daher aus der Disjunktion rausgelassen werden kann. (Nach dem Gesetz $F \vee G \equiv G$, falls F unerfüllbar, siehe Vorlesung.) Außerdem gilt $F \wedge G \models F$ für beliebige Formeln F und G . Wendet man dies geschickt auf die übrigen Formelbestandteile an, so erhält man:

$$F_1 \equiv (\neg A \wedge (\neg B \vee C)) \vee (B \wedge C) \equiv \neg A \vee C \equiv A \rightarrow C = F_2$$

Die Umkehrung $F_2 \models F_1$ gilt nicht, da z.B. für die Belegung \mathcal{A} mit $\mathcal{A}(A) = \mathcal{A}(C) = 0$ und $\mathcal{A}(B) = 1$ gilt: $\mathcal{A}(F_2) = 1$, aber $\mathcal{A}(F_1) = 0$.

Die obigen Sachverhalte lassen sich auch leicht folgern, wenn man sich klar macht, dass die Wahrheitstabelle der Implikation der “Kleiner-Gleich”-Relation (eingeschränkt auf 0 und 1) entspricht. Wenn $A \leq B$ und $B \leq C$, dann auch $A \leq C$, aber nicht notwendigerweise umgekehrt.

- (b) $F_1 \models F_2$ gilt trivialerweise, da F_2 eine Tautologie ist. $F_2 \models F_1$ könnte in diesem Fall nur gelten, wenn auch F_1 eine Tautologie wäre, was aber nicht der Fall ist:

$$\begin{aligned} F_1 &= (A \rightarrow B) \wedge (B \rightarrow \neg A) \\ &\equiv (\neg A \vee B) \wedge (\neg B \vee \neg A) \\ &\equiv \neg A \vee (B \wedge \neg B) && \text{(Distributivgesetz)} \\ &\equiv \neg A && (B \wedge \neg B \text{ unerfüllbar}) \end{aligned}$$

- (c) $F_1 \wedge F_2$ sind äquivalent, d.h. es gilt $F_1 \models F_2$ und $F_2 \models F_1$. Zum Beweis macht man sich klar, dass $F \rightarrow G \equiv \neg F \vee G$ (Definition) und $\neg(F \rightarrow G) \equiv F \wedge \neg B$ (De Morgan!) gelten. Durch mehrfache Anwendung dieser Äquivalenzen erhält man:

$$\begin{aligned} F_1 &= (A \rightarrow B) \rightarrow (C \vee \neg D) \\ &\equiv (A \rightarrow B) \rightarrow (D \rightarrow C) \\ &\equiv \neg(A \rightarrow B) \vee (D \rightarrow C) \\ &\equiv (A \wedge \neg B) \vee (D \rightarrow C) = F_2 \end{aligned}$$

Aufgabe 2 Modellierung von Sudoku

Inspiziert durch seine Vorlesung versucht Herr Esparza, Sudoku-Puzzles mit Hilfe der Aussagenlogik zu lösen. Leider muss er viel Zeit damit verbringen, die Folien vorzubereiten und dann auch noch einen Beamer für die Vorlesung aufzutreiben, so dass er nicht so recht vorankommt.

Er überlegt sich aber noch, dass er Variablen der Form $v_{x,y,z}$ benutzen kann, um auszudrücken, dass das Feld in Zeile x und Spalte y die Zahl z enthält, und dass die folgende Formel F dann bedeutet, dass die erste Zeile alle Zahlen von 1 bis 9 enthält (in einem üblichen Sudoku-Feld der Größe 9x9):

$$F = \bigwedge_{z=1}^9 \left(\bigvee_{y=1}^9 v_{1,y,z} \right)$$

- (a) Helfen Sie Herrn Esparza, indem Sie seine Formel F zu einer Formel F' vervollständigen, die *alle* regelkonform ausgefüllten Sudoku-Puzzles der Größe 9x9 beschreibt. Benutzen Sie Variablen der Form $v_{x,y,z}$ mit der Bedeutung wie oben.
- (b) Herr Esparza hat unter <http://www.forbeginners.info/sudoku-puzzles/> eine Sudoku-Webseite entdeckt und sich das unten abgebildete Puzzle heruntergeladen, das dort als “extrem schwer” bezeichnet wird. Er stellt fest, dass es tatsächlich zu schwer für ihn ist und will es deshalb mit Hilfe seines Computers lösen. Dazu überlegt er sich, dass die Lösung eindeutig durch eine erfüllende Belegung der Formel $F' \wedge G$ charakterisiert wird, wobei F' die Formel aus (a) ist und $G = v_{1,1,6} \wedge v_{1,8,4} \wedge v_{2,3,5} \wedge \dots$ die vorgelegten Felder beschreibt. Ein SAT-Solver, d.h. ein Programm, welches das Erfüllbarkeitsproblem für Formeln der Aussagenlogik löst, kann dann auf $F' \wedge G$ angesetzt werden, um eine solche Belegung zu finden.

	1	2	3	4	5	6	7	8	9
1	6							4	
2			5			2			7
3	7	2	9						3
4		9			4				1
5					6				
6	4				8			7	
7	3						1	6	5
8	2			4			8		
9		5							4

Auf der Vorlesungs-Webseite finden Sie unter dem Stichpunkt “Übungen” ein Programm, das ein Sudoku-Puzzle einliest, daraus eine Formel generiert und dann mit Hilfe des SAT-Solvers Limboole eine Lösung findet. Leider generiert das Programm bislang nur die Formel $F \wedge G$. Verbessern Sie es so, dass es die Formel $F' \wedge G$ erzeugt, und wenden Sie es auf das unten dargestellte Sudoku-Puzzle an. (Weitere Einzelheiten sind auf der o.g. Webseite erläutert.)

- (c) Überlegen Sie sich, wie man einen SAT-Solver einsetzen könnte, um Minesweeper zu lösen. Welche Probleme treten auf, und wie kann man sie umgehen?

Aufgabe 2 (Lösungsvorschlag) *Modellierung von Sudoku*

- (a) Die Formel F' muss ausdrücken, dass in jeder Zeile, jeder Spalte und jedem 3x3-Unterfeld jede Zahl vorkommen muss, und dass keine Zelle mehr als eine Zahl enthalten kann. Sie lässt sich also darstellen als $F' = F_1 \wedge F_2 \wedge F_3 \wedge F_4$. Dabei

erhält man die Zeilen- und Spaltenbedingungen ganz einfach so:

$$F_1 = \bigwedge_{x=1}^9 \bigwedge_{z=1}^9 \bigvee_{y=1}^9 v_{x,y,z}$$

$$F_2 = \bigwedge_{y=1}^9 \bigwedge_{z=1}^9 \bigvee_{x=1}^9 v_{x,y,z}$$

F_1 und F_2 unterscheiden sich nur dadurch, dass die Rolle der Zeilen- und der Spaltenvariablen vertauscht wurden. F_3 (die Feldbedingung) folgt demselben Prinzip: In der ersten Konjunktion werden die 3x3-Felder aufgezählt, in der Disjunktion die einzelnen Zellen innerhalb des jeweiligen Felds. Nur die Art und Weise, wie diese Aufzählung geschieht, ist etwas komplizierter:

$$F_3 = \bigwedge_{s=0}^2 \bigwedge_{t=0}^2 \bigwedge_{z=1}^9 \bigvee_{u=1}^3 \bigvee_{v=1}^3 v_{s*3+u,t*3+v,z}$$

Keine Zelle kann mehr als eine Zahl enthalten:

$$F_4 = \bigwedge_{x=1}^9 \bigwedge_{y=1}^9 \bigwedge_{z_1=1}^8 \bigwedge_{z_2=z_1+1}^9 \neg(v_{x,y,z_1} \wedge v_{x,y,z_2})$$

Bemerkungen: Aus Symmetriegründen reicht es bei F_4 aus, nur die Paare mit $z_1 < z_2$ zu betrachten. Die Bedingung, dass in jeder Zelle auch *mindestens* eine Zahl stehen muss, wird durch die übrigen Bedingungen bereits impliziert (Schubfachprinzip).

- (b) Das Programm, das die obige Formel F' für Limboole erzeugt, finden Sie auf der Vorlesungs-Webseite. Für das in der Aufgabenstellung gezeigte Puzzle findet Limboole innerhalb von 6 s eine Lösung (auf einem Laptop mit Pentium 4, 1.8 GHz), für das zweite mitgelieferte Puzzle sogar in 0,6 s.
- (c) Auch Minesweeper lässt sich prinzipiell mithilfe eines SAT-Solvers lösen. Für die Zelle in Zeile x und Spalte y benutzt man eine Variable $v_{x,y}$, die 1 sein soll, wenn das Feld (x, y) eine Mine enthält und 0 sonst. Die Information, die durch jedes einzelne bekannte Feld geliefert wird, ergibt eine Teilformel. Im Beispiel aus Aufgabe 1 (c) vom ersten Übungsblatt erhält man aus der "1" auf Feld A4 die Teilformel $(v_{A5} \vee v_{B5}) \wedge \neg(v_{A5} \wedge v_{B5})$. Aus der Tatsache, dass Feld F6 mit einem Stern markiert ist, erhält man v_{F6} . Die Konjunktion F aller dieser Teilformeln beschreibt alle möglichen Situationen.

Problematisch ist, dass man nicht für jedes Feld genug Informationen hat, um sicher festzustellen, ob es eine Mine enthält oder nicht. Wenn der SAT-Solver *irgendeine* erfüllende Belegung findet, wird er diese ausgeben. Im o.g. Beispiel könnte er feststellen, dass $v_{A5} = 0$ und $v_{B5} = 1$ Teil einer Lösung ist und diese

ausgeben, auch wenn es eine weitere Lösung mit der umgekehrten Belegung gibt. Verlässt man sich aber auf diese “zufällig” gefundene, nicht eindeutige Lösung, so verliert man evtl. das Spiel.

Umgehen kann man dies, indem man den SAT-Solver fragt, ob $F \rightarrow v_{A5}$ eine gültige Formel ist. Wenn ja, liegt auf A5 eine Mine. Wenn nein, fragt man, ob $F \rightarrow \neg v_{A5}$ eine gültige Formel ist. Wenn ja, kann A5 aufgedeckt werden. Wenn beide Formeln nicht gültig sind, kann über A5 keine sichere Aussage gemacht werden. Dieser Test muss aber für jedes einzelne Feld wiederholt werden.

Ein weiterer Unterschied ist, dass der SAT-Solver bei Sudoku die Lösung des gesamten Spiels in einem Zug berechnen kann. In Minesweeper gewinnt man während des Spiels zusätzliche Information und müsste den SAT-Solver nach jedem Aufdecken eines Felds neu aufrufen. Insgesamt ergibt sich, dass der Einsatz eines SAT-Solvers für Minesweeper weniger geeignet ist.

Aufgabe 3 Formelmengen

(a) Geben Sie aussagenlogische Formeln F_1, F_2, F_3 mit folgenden Eigenschaften an:

- (i) Die Menge $\{F_1, F_2, F_3\}$ ist unerfüllbar.
- (ii) Die Mengen $\{F_1, F_2\}, \{F_1, F_3\}, \{F_2, F_3\}$ sind alle erfüllbar.

Geben Sie für die zwei-elementigen Mengen Modelle an.

(b) Sei $n \geq 3$ beliebig. Geben Sie eine Formelmenge $M_n = \{F_1, \dots, F_n\}$ mit folgenden Eigenschaften an:

- (i) M_n ist unerfüllbar.
- (ii) Alle $(n - 1)$ -elementigen Teilmengen von M_n sind erfüllbar.

Begründen Sie kurz, warum die von Ihnen gewählte Menge M_n die Eigenschaften (i) und (ii) hat.

(c) Sei $n \geq 3$ beliebig und M_n eine Formelmenge wie in (b). Es sei v_n die Anzahl unterschiedlicher Variablen, die in M_n vorkommt. Wie groß muss v_n *mindestens* sein, damit Eigenschaften (i) und (ii) erfüllt sind?

Aufgabe 3 (Lösungsvorschlag) Formelmengen

Hinweis: Wir betrachten eine Menge M von Formeln als erfüllbar, wenn die Konjunktion aller Formeln aus M erfüllbar ist, es also eine passende Belegung gibt, die *jede einzelne* Formel in M erfüllt. Umgekehrt ist M unerfüllbar, wenn es keine Belegung gibt, die alle Formeln in M zugleich erfüllt.

- (a) Es gibt viele mögliche Lösungen. Wenn einem partout keine einfallen will, kann man einen Satz von Formeln mit Variablen A und B systematisch konstruieren, etwa so:

$\{F_1, F_2\}$ soll erfüllbar sein, also muss es eine Belegung geben, die sowohl F_1 als auch F_2 erfüllt. Legen wir uns also auf irgendeine Belegung fest, sagen wir $A = 1$ und $B = 1$, die dazu passenden Formeln konstruieren wir später. Für die Menge $\{F_1, F_3\}$ legen wir uns ebenso beliebig auf $A = 1$ und $B = 0$ fest, und für $\{F_2, F_3\}$ auf $A = 0$ und $B = 1$. Wichtig ist nur, dass wir drei verschiedene Belegungen wählen.

Jetzt konstruieren wir die passenden Formeln: F_1 muss für $A = 1$, $B = 1$ und für $A = 1$, $B = 0$ erfüllt sein, die Formel $F_1 = A$ tut genau dies. Analog ergibt sich $F_2 = B$ und $F_3 = \neg(A \leftrightarrow B)$. Man überzeuge sich, dass alle Bedingungen aus der Aufgabenstellung erfüllt sind.

Zusatzaufgabe: Man überlege sich, wie viele verschiedene Formelmengen man auf diese Weise konstruieren kann...

- (b) Die folgende Lösung ist ähnlich zu der in (a): Man benutzt Variablen A_1, \dots, A_{n-1} und setzt $F_i = A_i$ für $i = 1, \dots, n-1$, dazu $F_n = \neg(A_1 \wedge \dots \wedge A_{n-1})$.

Die Menge $M_n \setminus \{F_n\}$ ist erfüllbar, indem man alle Variablen auf 1 setzt, die Menge $M_n \setminus \{F_i\}$, $1 \leq i < n$ ist erfüllbar, indem man alle Variablen außer A_i auf 1 setzt. Die Menge M_n ist offensichtlich nicht erfüllbar.

- (c) Analog zu der Konstruktion aus (a) überlegt man sich, dass man n Belegungen braucht, um die Bedingung (ii) zu erfüllen. Wegen Bedingung (i) müssen alle diese Belegungen verschieden sein. Mit m Variablen kann man bis zu 2^m verschiedene Belegungen erzeugen, also gilt $v_n \geq \log_2 n$.