Notizen zur Aussagenlogik

Inhaltsverzeichnis

Notizen zur Aussagenlogik

Aussagenlogische Formeln sprechen über Tupel von Booleschen Wahrheitswerten. Wahrheitswerte (0 = falsch, 1 = wahr) werden als Belegungen (Zuweisungen, Interpretationen) aussagenlogischer Variablen betrachtet; Formeln beschreiben Boolesche Verknüpfungen dieser Variablen. Aussagenlogik AL ist die Logik dieser Formeln.

Kontext: jede bit-weise Repräsentation von Information kann erfasst werden; jede Eigenschaft von so kodierten endlichen Informationsinstanzen wird durch aussagenlogische Formeln erfasst (i.d.S.: universelles Format für die Spezifikation).

Fragen nach logischen Eigenschaften von Formeln und Formelmengen (z.B. Erfüllbarkeit), Beziehungen zwischen Formeln (z.B. Implikationen, Äquivalenzen), können anhand von formalen Kalkülen (auch algorithmisch) analysiert werden.

Wir behandeln zwei derartige Kalküle für die Aussagenlogik: Resolution (ein Kalkül für Widerlegungsbeweise) und einen Sequenzenkalkül (ein Deduktionskalkül für allg. formale Beweise).

1 Syntax und Semantik der AL

Symbole

$0, 1$	Wahrheitswerte, aussagenlogische Konstanten [1]
$p, q, r, \ldots, p_1, p_2, \ldots\ldots$	Aussagenvariablen
$\neg$	Junktor für Negation ("nicht")
$\wedge, \vee$	Junktoren für Konjunktion ("und"), Disjunktion ("oder")
$\rightarrow, \leftrightarrow, \ldots$	weitere, definierte Junktoren
$(,)$	

Definition 1.1 [Syntax]
Zu einer gegebenen Menge $\mathcal{V}$ von aussagenlogischen Variablen ist die Menge der *aussagenlogischen Formeln* über Variablen aus $\mathcal{V}$, $\mathrm{AL}(\mathcal{V})$, induktiv erzeugt wie folgt:

- (atomare Formeln): 0, 1, p (für $p \in \mathcal{V}$).
- (Negation): für $\varphi \in \mathrm{AL}(\mathcal{V})$ ist auch $\neg\varphi \in \mathrm{AL}(\mathcal{V})$.
- (Konjunktion, Disjunktion): für $\varphi, \psi \in \mathrm{AL}(\mathcal{V})$ sind auch $(\varphi \wedge \psi), (\varphi \vee \psi) \in \mathrm{AL}(\mathcal{V})$.

Bem.: Allgemeinere AL-Formeln, die auch $\rightarrow, \leftrightarrow, \ldots$ als Junktoren zulassen, werden als definierte Abkürzungen eingeführt, z.B.

$$\begin{array}{rcl} (\varphi \rightarrow \psi) & := & (\neg\varphi \vee \psi), \\ (\varphi \leftrightarrow \psi) & := & \big((\neg\varphi \wedge \neg\psi) \vee (\varphi \wedge \psi)\big). \end{array}$$

Bem.: Äussere Klammern wie in $(\varphi \wedge \psi)$ werden i.d.R. weggelassen.

Mit Standardvariablenmengen $\mathcal{V} = \{p_i \colon i \geqslant 1\}$ bzw. $\mathcal{V}_n = \{p_1, \ldots, p_n\}$ für $n \in \mathbb{N}$, sei

$$\begin{array}{rcl} \mathrm{AL} & := & \mathrm{AL}(\mathcal{V}), \\ \mathrm{AL}_n & := & \mathrm{AL}(\mathcal{V}_n). \end{array}$$

[1] Alternativ oft auch F,W oder, vor allem in Formeln, $\bot$,$\top$.

Übung 1.2 Geben Sie für festes $n \in \mathbb{N}$ eine kontextfreie Grammatik für AL_n an, über $\Sigma = \{0, 1, (,), \neg, \wedge, \vee\} \cup \{p_1, \ldots, p_n\}$. Wie könnte man die Syntax (und die zugehörige Grammatik) modifizieren um z.B. redundante äussere Klammerungen zu vermeiden, aber eindeutige Lesbarkeit zu erhalten?

Formeln $\varphi \in \mathrm{AL}(\mathcal{V})$ sprechen über *Interpretationen*, die die Variablen in $\mathcal{V}$ durch Boolesche Wahrheitswerte in $\mathbb{B} = \{0, 1\}$ belegen (daher auch: *Belegungen*).

Definition 1.3 Eine $\mathcal{V}$-*Interpretation* (*Belegung*) ist eine Funktion

$$\begin{array}{rcl} \mathfrak{I}\colon \mathcal{V} & \longrightarrow & \mathbb{B} \\ p & \longmapsto & \mathfrak{I}(p) \end{array} \qquad \mathfrak{I} \text{ interpretiert } p \text{ als } \begin{cases} \text{“wahr”} & \text{wenn } \mathfrak{I}(p) = 1, \\ \text{“falsch”} & \text{wenn } \mathfrak{I}(p) = 0. \end{cases}$$

Zur Definition der Semantik weisen wir jeder Formel $\varphi \in \mathrm{AL}(\mathcal{V})$ zu einer gegebenen $\mathcal{V}$-Interpretation einen *Wahrheitswert* $\varphi^{\mathfrak{I}} \in \mathbb{B}$ zu. Die Funktion

$$\begin{array}{rcl} {}^{\mathfrak{I}}\colon \mathrm{AL}(\mathcal{V}) & \longrightarrow & \mathbb{B} \\ \varphi & \longmapsto & \varphi^{\mathfrak{I}}, \end{array}$$

wird induktiv über den Aufbau der Formel φ definiert:

- (atomare Formeln): $0^{\mathfrak{I}} := 0$; $1^{\mathfrak{I}} := 1$; $p^{\mathfrak{I}} := \mathfrak{I}(p)$.
- (Negation): $(\neg\varphi)^{\mathfrak{I}} := 1 - \varphi^{\mathfrak{I}}$.
- (Konjunktion): $(\varphi \wedge \psi)^{\mathfrak{I}} := \min(\varphi^{\mathfrak{I}}, \psi^{\mathfrak{I}}) \quad \big(= \varphi^{\mathfrak{I}} \cdot \psi^{\mathfrak{I}}\big)$.
- (Disjunktion): $(\varphi \vee \psi)^{\mathfrak{I}} := \max(\varphi^{\mathfrak{I}}, \psi^{\mathfrak{I}}) \quad \big(= \varphi^{\mathfrak{I}} + \psi^{\mathfrak{I}} - \varphi^{\mathfrak{I}} \cdot \psi^{\mathfrak{I}}\big)$.

Definition 1.4 [Semantik]
Für $\mathcal{V}$-Interpretation $\mathfrak{I}$ und $\varphi \in \mathrm{AL}(\mathcal{V})$:
$\mathfrak{I}$ *erfüllt* φ gdw. $\varphi^{\mathfrak{I}} = 1$. Schreibweise: $\mathfrak{I} \models \varphi$.
Für Formelmengen $\Phi \subseteq \mathrm{AL}(\mathcal{V})$ entsprechend: $\mathfrak{I} \models \Phi$ gdw. $\mathfrak{I} \models \varphi$ für alle $\varphi \in \Phi$.

Sprechweisen für $\mathfrak{I} \models \varphi$: “$\mathfrak{I}$ *erfüllt* φ”,
“$\mathfrak{I}$ *ist Modell von* φ”,
“φ *ist wahr unter* $\mathfrak{I}$”.
Die Relation $\models$ heißt *Modellbeziehung*.

Schreibweisen: für $\varphi \in \mathrm{AL}_n$ schreiben wir auch $\varphi = \varphi(p_1, \ldots, p_n)$.
Für $(b_1, \ldots, b_n) \in \mathbb{B}^n$ ist dann

$$\varphi[b_1, \ldots, b_n] := \varphi^{\mathfrak{I}} \text{ für die Interpretation } (\mathfrak{I}(p_i) = b_i)_{i=1,\ldots,n}.$$

Die Semantik von $\varphi \in \mathrm{AL}_n$ wird vollständig durch die Funktion wiedergegeben, die $(b_1, \ldots, b_n)$ auf $\varphi[b_1, \ldots, b_n]$ abbildet (eine n-stellige Boolesche Funktion). Eine nützliche Beschreibung der Semantik von φ ist eine Wertetabelle für diese Funktion. Die Semantik einer Formel wird durch ihre *Wahrheitstafel* eindeutig beschrieben.

Die Wahrheitstafeln für die Standardjunktoren:

p	$\neg p$
0	1
1	0

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

Für $\to$ und $\leftrightarrow$ ergeben sich aus ihren Definitionen:

p	q	$p \to q$
0	0	1
0	1	1
1	0	0
1	1	1

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

2 Grundlegende semantische Begriffe

2.1 Semantische Folgerung und Allgemeingültigkeit

Eine Folgerungsbeziehung liegt vor, wenn die Wahrheit der Voraussetzung die Wahrheit der Konklusion erzwingt. Ist eine Formel (ohne Voraussetzungen) immer wahr, so heißt sie allgemeingültig (auch: Tautologie).

Definition 2.1 [Folgerungsbeziehung, $\varphi \models \psi$ bzw. $\Phi \models \psi$.]
Für $\varphi, \psi \in \mathrm{AL}(\mathcal{V})$, bzw. $\Phi \subseteq \mathrm{AL}(\mathcal{V})$ und $\psi \in \mathrm{AL}(\mathcal{V})$:
ψ ist eine *logische Folgerung* von φ, oder ψ *folgt aus* φ, in Symbolen $\varphi \models \psi$, gdw. für alle $\mathcal{V}$-Interpretationen $\mathfrak{I}$ gilt: $\mathfrak{I} \models \varphi \;\Rightarrow\; \mathfrak{I} \models \psi$.
Entsprechend ist $\Phi \models \psi$ für Formelmengen Φ definiert (ψ *folgt aus* Φ).

Beispiele: Man weist anhand von Wahrheitstafeln nach, dass z.B. für beliebige φ, ψ gilt: $\varphi \wedge \psi \models \varphi \vee \psi$, oder dass $\psi \models (\psi \wedge \varphi) \vee (\psi \wedge \neg\varphi)$.

Definition 2.2 [Allgemeingültigkeit]
Eine Formel $\varphi \in \mathrm{AL}(\mathcal{V})$ heißt *allgemeingültig* gdw. für alle $\mathcal{V}$-Interpretationen $\mathfrak{I}$ gilt: $\mathfrak{I} \models \varphi$. In Symbolen: $\models \varphi$ (anstelle von $\emptyset \models \varphi$).

Z.B. gilt für jedes φ: $\models \varphi \vee \neg\varphi$.
Man prüft mittels Wahrheitstafeln nach, dass $\varphi \models \psi$ gdw. $\models \varphi \to \psi$.

2.2 Logische Äquivalenz

Formeln mit derselben Semantik heißen logisch äquivalent.

Definition 2.3 [logische Äquivalenz]
Zwei Formeln $\varphi, \psi \in \mathrm{AL}(\mathcal{V})$ heißen *(logisch) äquivalent*, gdw. für alle $\mathcal{V}$-Interpretationen $\mathfrak{I}$ gilt: $\mathfrak{I} \models \varphi$ gdw. $\mathfrak{I} \models \psi$. In Symbolen: $\varphi \equiv \psi$.

Für $\varphi, \psi \in \mathrm{AL}_n$ gilt demnach:

$$\varphi \equiv \psi \quad \text{gdw.} \quad \text{für alle } (b_1, \ldots, b_n) \in \mathbb{B}^n \text{ ist } \varphi[b_1, \ldots, b_n] = \psi[b_1, \ldots, b_n].$$

Logische Äquivalenz lässt sich also durch Vergleich der Wahrheitstafeln nachweisen.
Man prüft anhand der Definitionen und Wahrheitstafeln auch nach, dass

$$\varphi \equiv \psi \quad \text{gdw} \quad \models \varphi \leftrightarrow \psi.$$

Beispiel 2.4 (Dualität von $\vee$ und $\wedge$) Es gilt $p \vee q \equiv \neg(\neg p \wedge \neg q)$ und $p \wedge q \equiv \neg(\neg p \vee \neg q)$.

Inbesondere könnte man in der Syntax von AL auf einen der Junktoren $\wedge$ oder $\vee$ verzichten, ohne semantisch etwas zu verlieren.

Übung 2.5 Geben Sie eine Übersetzung von AL_n-Formeln in logisch äquivalente $\wedge$-freie AL_n-Formeln an. Dazu soll die Übersetzung induktiv über den Aufbau der Formeln definiert werden, und die Äquivalenz induktiv nachgewiesen werden.

Weitere Äquivalenzen, die man nachprüfen sollte (hier: Kommutativität, Assoziativität, Distributivität; vergleiche auch weitere Identitäten in Booleschen Algebren, FG I):

$$\begin{array}{lll} p \vee q \equiv q \vee p & (p \vee q) \vee r \equiv p \vee (q \vee r) & p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r) \\ p \wedge q \equiv q \wedge p & (p \wedge q) \wedge r \equiv p \wedge (q \wedge r) & p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r) \end{array}$$

Folgerungen: Reihenfolge und Klammerung innerhalb mehrfacher Konjunktionen oder Disjunktionen sind logisch unerheblich. Wir schreiben daher z.B. für eine endliche Menge $\Phi = \{\varphi_1, \ldots, \varphi_k\} \subseteq \mathrm{AL}_n$ auch einfach $\bigwedge \Phi$ für (irgendeine der untereinander äquivalenten) Konjunktionsketten aller aufgeführten φ_i; ebenso $\bigwedge_{i=1}^k \varphi_i$ usw.

2.3 Erfüllbarkeit

Erfüllbarkeit betrifft die Existenz mindestens einer Interpretation, die eine Formel oder Formelmenge erfüllt.

Definition 2.6 [Erfüllbarkeit]
$\varphi \in \mathrm{AL}(\mathcal{V})$ heißt *erfüllbar* gdw. eine Interpretation existiert mit $\mathfrak{I} \models \varphi$.
Analog für Formelmengen $\Phi \subseteq \mathrm{AL}(\mathcal{V})$: Φ ist *erfüllbar* gdw. es eine Interpretation $\mathfrak{I}$ gibt, die Φ erfüllt, d.h. mit $\mathfrak{I} \models \varphi$ für alle $\varphi \in \Phi$.

Schreibweisen: SAT für die Menge der erfüllbaren Formeln (englisch: *satisfiability*).

Das *Erfüllbarkeitsproblem* ist das Entscheidungsproblem, für vorgelegte Formel φ zu entscheiden, ob φ erfüllbar ist.

Erfüllbarkeit ist das zentrale algorithmische Problem für AL (und andere Logiken). Andere Probleme, wie Allgemeingültigkeit oder die Folgerungbeziehung lassen sich oft direkt auf SAT reduzieren:

Beobachtung 2.7 Erfüllbarkeit ist dual zur Allgemeingültigkeit:
φ allgemeingültig gdw. $\neg\varphi$ nicht erfüllbar.

Beobachtung 2.8 Die Folgerungsbeziehung lässt sich auf Erfüllbarkeit reduzieren:
$\Phi \models \psi$ gdw. $\Phi \cup \{\neg\psi\}$ unerfüllbar ist.

Ein naiver Algorithmus zur Entscheidung von SAT(AL) würde für $\varphi \in AL_n$ auf vollständiger Suche durch alle 2^n Belegungen basieren, mit exponentiellem Zeitaufwand. Die Frage, ob es einen Algorithmus mit polynomialer Laufzeit gibt, ist äquivalent zum großen offenen Problem der Komplexitätstheorie ob "NP=P" ist, d.h. ob deterministische Turingmaschinen mit polynomial beschränkter Laufzeit dasselbe leisten können wie nichtdeterministische Turingmaschinen mit polynomial beschränkter Laufzeit. (Stichwort: NP-Vollständigkeit von SAT(AL).)

3 AL und Boolesche Funktionen

Allgemein ist eine n-stellige *Boolesche Funktion* eine Funktion des Typs

$$\begin{array}{rcl} f\colon \mathbb{B}^n & \longrightarrow & \mathbb{B} \\ (b_1, \ldots, b_n) & \longmapsto & f(b_1, \ldots, b_n). \end{array}$$

Definition 3.1 Für $n \in \mathbb{N}$ sei $\mathcal{B}_n$ die Menge aller n-stelligen Booleschen Funktionen.

Grenzfall für $n = 0$: $\mathbb{B}^0 = \{\Box\}$, wo $\Box$ für das leere Tupel steht. Eine Boolesche Funktion in $\mathcal{B}_0$ kann $\Box$ auf 0 oder 1 abbilden; also gibt es genau zwei Funktionen in $\mathcal{B}_0$.

Mit einer Formel $\varphi \in \mathrm{AL}_n$ assoziieren wir die Boolesche Funktion $f_\varphi \in \mathcal{B}_n$:

$$\begin{array}{rcl} f_\varphi \colon \mathbb{B}^n & \longrightarrow & \mathbb{B} \\ (b_1, \ldots, b_n) & \longmapsto & \varphi[b_1, \ldots, b_n]. \end{array}$$

Kann man jede Boolesche Funktion so darstellen?

3.1 Funktionale Vollständigkeit

Satz 3.2 *Zu jeder Funktion $f \in \mathcal{B}_n$ gibt es $\varphi \in \mathrm{AL}_n$ mit $f = f_\varphi$.*

Damit wissen wir nun auch: $\varphi \mapsto f_\varphi$ liefert eine bijektive Korrespondenz zwischen den Äquivalenzklassen von AL_n-Formeln bezüglich $\equiv$ und den n-stelligen Booleschen Funktionen.

Beweis (des Satzes) Sei $f \in \mathcal{B}_n$ durch eine Wertetabelle für $f(\mathbf{b})$, $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{B}^n$ gegeben. Für $\mathbf{b} = (b_1, \ldots, b_n) \in \mathbb{B}^n$ sei

$$\varphi_{\mathbf{b}} = \varphi_{\mathbf{b}}(p_1, \ldots, p_n) := \bigwedge\{p_i \colon b_i = 1\} \wedge \bigwedge\{\neg p_i \colon b_i = 0\}.$$

Dann ist $\varphi_{\mathbf{b}}[b'_1, \ldots, b'_n] = 1$ gdw. $(b'_1, \ldots, b'_n) = \mathbf{b}$.

Sei

$$\varphi_f := \bigvee\big\{\varphi_{\mathbf{b}} \colon \mathbf{b} \in \mathbb{B}^n, f[\mathbf{b}] = 1\big\}.$$

Man prüft durch Fallunterscheidung nach, dass φ_f die Funktion f darstellt. $\Box$

Oben hatten wir $|\mathcal{B}_0| = 2$ gesehen. Allgemein ist $|\mathcal{B}_n| = 2^{2^n}$.

Demnach gibt es genau 2^{2^n} viele *bis auf logische Äquivalenz* verschiedene Formeln in AL_n.

Begründung zur Anzahlaussage für $\mathcal{B}_n$: Für jedes Tupel im Definitionsbereich kann der Funktionswert aus $\mathbb{B} = \{0, 1\}$ gewählt werden. Also ist $|\mathcal{B}_n| = 2^{|\mathbb{B}^n|}$ ($|\mathbb{B}^n|$ viele binäre Auswahlen) und da $|\mathbb{B}^n| = 2^n$, folgt die Behauptung.

Übung 3.3 Zeige die Anzahlaussage $|\mathcal{B}_n| = 2^{2^n}$ direkt durch Induktion über $n \in \mathbb{N}$. Hinweis: Zum Induktionsschritt von n nach $n + 1$ betrachte man für eine Funktion $f \in \mathcal{B}_{n+1}$ die beiden Spezialisierungen

$$\begin{array}{rcl} f_0 \colon \mathbb{B}^n & \longrightarrow & \mathbb{B} \\ (b_1, \ldots, b_n) & \longmapsto & f(b_1, \ldots, b_n, 0) \end{array} \qquad \begin{array}{rcl} f_1 \colon \mathbb{B}^n & \longrightarrow & \mathbb{B} \\ (b_1, \ldots, b_n) & \longmapsto & f(b_1, \ldots, b_n, 1), \end{array}$$

um zu zeigen, dass $|\mathcal{B}_{n+1}| = |\mathcal{B}_n| \cdot |\mathcal{B}_n|$.

Übung 3.4 Zum doppelt exponentiellen Wachstum: Wie groß muss man n wählen, damit es mehr n-stellige Boolesche Funktionen als Elementarteilchen im 'sichtbaren Universum' gibt?

3.2 Konjunktive und disjunktive Normalformen

Die Formel φ_f aus dem Beweis von Satz 3.2 ist eine Disjunktion über Konjunktionen von atomaren Formeln und negierten atomaren Formeln: eine Formel in disjunktiver Normalform, DNF.

Aussagenvariablen und ihre Negationen werden in AL zusammenfassend als *Literale* bezeichnet; p, $\neg p$ sind Literale. Beachte, dass die Negation eines Literals zu einem Literal äquivalent ist.

Definition 3.5 Eine *DNF-Formel* ist eine Disjunktion über Konjunktionen von Literalen. Eine *KNF-Formel* ist eine Konjunktion über Disjunktionen von Literalen.

Bem.: Es ist sinnvoll, per Konvention auch die leere Konjunktion bzw. die leere Disjunktion als Baustein zuzulassen, mit der Semantik

$$\bigvee \emptyset \equiv 0 \quad \text{und} \quad \bigwedge \emptyset \equiv 1.$$

Frage: Warum sind diese semantischen Setzungen natürlich? Wozu nützlich? (Siehe weiter unten.)

Aus den Äquivalenzen $\neg \bigvee_i \psi_i \equiv \bigwedge_i \neg\psi_i$, sowie $\neg \bigwedge_i \xi_i \equiv \bigvee_i \neg\xi_i$, erhält man:

Beobachtung 3.6 Die Negation einer KNF-Formel ist zu einer DNF-Formel äquivalent, und umgekehrt.

Satz 3.7 (KNF und DNF) *Zu jeder Formel $\varphi \in \mathrm{AL}_n$ gibt es äquivalente Formeln $\varphi_1 \in \mathrm{AL}_n$ in KNF und $\varphi_2 \in \mathrm{AL}_n$ in DNF.*

Beweis Zu φ_2 in DNF: Sei $f \in \mathcal{B}_n$ die Boolesche Funktion, die von φ dargestellt wird. Die Formel φ_2 sei wie im Beweis von Satz 3.2 (als φ_f zur Funktion f) gewonnen: φ_2 ist eine DNF-Formel, und, da φ und φ_2 dieselbe Boolesche Funktion darstellen, ist $\varphi_2 \equiv \varphi$.

Zu φ_1 in KNF: betrachte die Boolesche Funktion f, die von $\neg\varphi$ dargestellt wird. Aus φ_f in DNF gewinnen wir die Formel $\neg\varphi_f \equiv \varphi$. Nach der Beobachtung ist $\neg\varphi_f$ äquivalent zu einer KNF-Formel. □

Übung 3.8 Verwende die Distributivität zwischen $\wedge$ und $\vee$, um z.B. aus einer DNF durch 'Ausmultiplizieren' eine äquivalente KNF zu gewinnen. Wie hängt die Länge der entstehenden Formel von der Länge der gegebenen Formel ab?

Prinzipiell kann man also jede Formel bis auf logische Äquivalenz sowohl in KNF als auch in DNF bringen. Insbesondere lässt sich jede KNF-Formel in eine äquivalente DNF-Formel umwandeln und umgekehrt. Bei Umwandlung von DNF in KNF oder umgekehrt kann die Länge der Formel dramatisch zunehmen.

Z.B. ist $\varphi_1 := \neg(p_1 \leftrightarrow p_2)$ äquivalent zur DNF $(p_1 \wedge \neg p_2) \vee (\neg p_1 \wedge p_2)$ und zur KNF $(\neg p_1 \vee \neg p_2) \wedge (p_1 \vee p_2)$. Aber:

Beispiel 3.9 Zu $m \geqslant 1$ betrachte die folgende Formel $\varphi_m \in \mathrm{AL}_{2m}$:

$$\varphi_m = \varphi_m(p_1, \ldots, p_{2m}) := \bigwedge_{i=1}^{m} \neg(p_{2i-1} \leftrightarrow p_{2i}).$$

φ_m verlangt, dass aus jedem Paar von Variablen $\{p_{2i-1}, p_{2i}\}$ genau eine wahr gemacht wird. Es gibt genau m solcher Paare, und demnach genau 2^m erfüllende Belegungen für die $2m$ Variablen von φ_m. Weiter gilt:

(i) φ_m ist zu einer KNF-Formel äquivalent mit $2m$ Konjunktionsgliedern mit je zwei Literalen. (KNF Darstellung von linearer Länge in m)

(ii) *Jede* DNF-Formel, die zu φ_m äquivalent ist, muss mindestens 2^m Disjunktionsglieder haben. (DNF Darstellung nur von exponentieller Länge in m)

Übung 3.10 Man beweise die Behauptungen des Beispiels. Hinweis zu (ii): Wir nehmen an, dass $\bigvee_{i\in I} \chi_i(p_1,\ldots,p_{2m}) \equiv \varphi_m$ eine DNF Darstellung ist. Zunächst überlegt man sich dass jedes Disjunktionsglied χ_i eine erfüllende Belegung von φ_m exakt festlegen muss, i.d.S.d. $\chi_i \equiv \varphi_{\mathbf{b}}$ (vgl. Beweis von Satz 3.2) für ein $\mathbf{b} \in \mathbb{B}^{2m}$ mit $\varphi_m[\mathbf{b}] = 1$. Nun überlegt man sich, dass in der Disjunktion keine der 2^m erfüllenden Belegungen von φ_m fehlen darf, d.h. es gibt zu jedem $\mathbf{b} \in \mathbb{B}^{2m}$ mit $\varphi_m[\mathbf{b}] = 1$ ein $i \in I$ derart dass $\chi_i \equiv \varphi_{\mathbf{b}}$.

Übung 3.11 Man überlege sich ein effizientes Verfahren (lineare Laufzeit in der Länge der Formel), das für DNF-Formeln entscheidet, ob sie erfüllbar sind. Warum lässt sich das nicht mittels Übersetzung von KNF in äquivalente DNF auf KNF-Formeln übertragen?

3.3 Vollständige Systeme von Junktoren

Für $n \geqslant 1$ lassen sich alle Booleschen Funktionen in $\mathcal{B}_n$ durch AL_n-Formeln darstellen, die nur die Aussagenvariablen in $\mathcal{V}_n$ und die logischen Junktoren $\wedge$ und $\neg$ benutzen. (Genauso auch mit $\vee$ und $\neg$.)

Grund: Man kann $\vee$ oder aber $\wedge$ durch Dualität eliminieren (und $0 \equiv p_1 \wedge \neg p_1$ sowie $1 \equiv p_1 \vee \neg p_1$ benutzen).

Ein solches System von Junktoren (oder zugehörigen Booleschen Funktionen) heißt *vollständig*. Es gibt sogar einzelne 2-stellige Junktoren, die für sich alleine vollständig sind. Ein Beispiel ist NAND (für "Negation von AND", $p \mid q := \neg(p \wedge q)$, auch Sheffer-Strich) mit der Wahrheitstafel:

p	q	$p \mid q$
0	0	1
0	1	1
1	0	1
1	1	0

Übung 3.12 (a) Zeigen Sie, dass $\mid$ vollständig ist.

(b) Finden Sie alle 2-stelligen Junktoren (Booleschen Funktionen), die für sich alleine ein vollständiges System ergeben.

(c) Welche 2-stelligen Junktoren bilden zusammen mit den Konstanten 0 und 1 ein vollständiges System?

4 AL Kompaktheitssatz (Endlichkeitssatz)

In einer endlichen Formelmenge $\Phi \subseteq \mathrm{AL}$ kommen nur endlich viele Aussagenvariablen vor, und bei der Erfüllbarkeit von Φ geht es um die Existenz einer geeigneten Belegung dieser endlich vielen Variablen: ein endlicher, wenn auch exponentiell großer Suchraum.

Für eine unendliche Formelmenge, in der unendlich viele Aussagenvariablen vorkommen, betrifft die Erfüllbarkeit die Existenz einer geeigneten Interpretation, die selbst schon ein unendliches Objekt ist. Demgegenüber sagt aber der folgende Satz (auch *Endlichkeitssatz*), dass es immer nur auf endlich viele Variablen (Formeln) gleichzeitig ankommt. Insbesondere folgt aus dem Satz, dass die Unerfüllbarkeit einer unendlichen Menge endlich nachweisbar ist! (Warum?)

Satz 4.1 (Kompaktheitssatz) *Sei* $\mathcal{V} = \{p_i : 1 \leqslant i \in \mathbb{N}\}$, $\mathrm{AL} = \mathrm{AL}(\mathcal{V})$.
Für jede Formelmenge $\Phi \subseteq \mathrm{AL}$ *sind äquivalent:*

(i) Φ *ist erfüllbar.*

(ii) *Jede endliche Teilmenge* $\Phi_0 \subseteq \Phi$ *ist erfüllbar.*

Korollar 4.2 *Für jede Formelmenge* $\Phi \subseteq \mathrm{AL}$ *und Formel* $\psi \in \mathrm{AL}$ *sind äquivalent:*

(i) $\Phi \models \psi$.

(ii) *Es gibt eine endliche Teilmenge* $\Phi_0 \subseteq \Phi$ *derart, dass* $\Phi_0 \models \psi$.

Übung 4.3 Folgern Sie die Aussage des Korollars aus derjenigen des Satzes, indem Sie die Folgerungsbeziehung in eine Erfüllbarkeitsaussage übersetzen.

Beweis (des Satzes)
(i) $\Rightarrow$ (ii) ist trivial. Wir weisen (ii) $\Rightarrow$ (i) nach.

Sei Φ derart dass (ii) gilt. Nenne eine $\mathcal{V}_n$-Interpretation $\mathfrak{I}_n$ *gut* wenn jede endliche Teilmenge von Φ von einer Interpretation erfüllt wird, die auf $\mathcal{V}_n$ mit $\mathfrak{I}_n$ übereinstimmt. D.h. die Wahl der Belegung der ersten n Variablen $p_1, \ldots, p_n$ gemäß $\mathfrak{I}_n$ erhält die Erfüllbarkeit aller endlichen Teilmengen von Φ (man 'verliert nichts' indem man sich i.S.v. $\mathfrak{I}_n$ festlegt).

Wir wählen induktiv eine *Kette* von Interpretationen $\mathfrak{I}_0, \mathfrak{I}_1, \mathfrak{I}_2, \ldots$ derart, dass für jedes $n \in N$ gilt:

(a) $\mathfrak{I}_n$ ist eine gute $\mathcal{V}_n$-Belegung.

(b) $\mathfrak{I}_{n+1}$ ist eine Fortsetzung von $\mathfrak{I}_n$, d.h. $\mathfrak{I}_{n+1}$ belegt $p_1, \ldots, p_n$ genau wie $\mathfrak{I}_n$ und legt zusätzlich eine Belegung von p_{n+1} neu fest.

Dann setzen wir

$$\begin{array}{rcl} \mathfrak{I} : \mathcal{V} & \longrightarrow & \mathbb{B} \\ p_n & \longmapsto & \mathfrak{I}(p_n) := \mathfrak{I}_n(p_n). \end{array}$$

Wegen (a) und (b) stimmt dann $\mathfrak{I}$ für jedes n auf $\mathcal{V}_n$ mit $\mathfrak{I}_n$ (und allen $\mathfrak{I}_m$ mit $m > n$) überein. Daraus folgt dann weiter, dass $\mathfrak{I}$ jede endliche Teilmenge von Φ erfüllt, denn jede endliche Teilmenge von Φ ist in einem AL_n enthalten und wird demnach von $\mathfrak{I}_n$ und also auch von $\mathfrak{I}$ erfüllt. Also erfüllt $\mathfrak{I}$ jedes $\varphi \in \Phi$, und demnach ganz Φ.

Zur Existenz der Folge $(\mathfrak{I}_n)_{n\in\mathbb{N}}$, argumentieren wir induktiv:

$\mathfrak{I}_0$ sei die leere Interpretation (über der leeren Variablenmenge $\mathcal{V}_0$). Aus Voraussetzung (ii) folgt, dass $\mathfrak{I}_0$ gut ist.

Sei $\mathfrak{I}_n$ gut. Wir behaupten, dass mindestens eine der beiden Fortsetzungen von $\mathfrak{I}_n$ auf p_{n+1} wieder gut ist. Bezeichne mit $\mathfrak{I}^0_{n+1}$ die Fortsetzung von $\mathfrak{I}_n$ mit $\mathfrak{I}^0_{n+1}(p_{n+1}) = 0$, mit $\mathfrak{I}^1_{n+1}$ die Fortsetzung mit $\mathfrak{I}^1_{n+1}(p_{n+1}) = 1$.

Indirekter Beweis der Behauptung: Sonst gäbe es für $\mathfrak{I}^0_{n+1}$ eine endliche Teilmenge $\Phi^0 \subseteq \Phi$ so, dass Φ^0 von *keiner* Fortsetzung von $\mathfrak{I}^0_{n+1}$ erfüllt wird; ebenso für $\mathfrak{I}^1_{n+1}$ eine endliche Teilmenge $\Phi^1 \subseteq \Phi$ so, dass Φ^1 von *keiner* Fortsetzung von $\mathfrak{I}^1_{n+1}$ erfüllt wird. Da $\mathfrak{I}_n$ aber gut ist, gibt es eine Fortsetzung $\mathfrak{I}$ von $\mathfrak{I}_n$, die $\Phi^0 \cup \Phi^1$ erfüllt. $\mathfrak{I}$ ist eine Fortsetzung von $\mathfrak{I}^0_{n+1}$ oder von $\mathfrak{I}^1_{n+1}$, je nachdem ob $\mathfrak{I}(p_{n+1}) = 0$ oder $\mathfrak{I}(p_{n+1}) = 1$ ist. Aber $\mathfrak{I}$ erfüllt Φ^0 *und* Φ^1 – im Widerspruch zur Wahl von Φ^0 und Φ^1. □

Bem.: Der Kompaktheitssatz gilt für beliebige, auch überabzählbare Variablenmengen $\mathcal{V}$ und Formelmengen Φ. Anstelle des obigen Beweises, der nur den abzählbaren Fall erfasst (warum?), muss man dann stärkere Prinzipien aus der Kombinatorik unendlicher Mengen investieren (Auswahlaxiom, Zornsches Lemma).

Beispiele zur Illustration der Stärke des Kompaktheitssatzes für AL:

- Königs Lemma: Ein unendlicher, aber endlich verzweigter Baum muss einen unendlichen Pfad haben.
- k-Färbbarkeit von Graphen: Ein Graph ist k-färbbar, gdw. jeder endliche Teilgraph k-färbbar ist.

Königs Lemma Ein *Baum* $\mathcal{T} = (V, E, \lambda)$ besteht aus: Knotenmenge V, Kantenrelation $E \subseteq V \times V$, Wurzel $\lambda \in V$, mit der Bedingung:

Jeder Knoten $u \in V$ ist auf genau einem E-Pfad $\lambda \stackrel{E}{\to} \cdots \stackrel{E}{\to} u$ von der Wurzel aus erreichbar.

(Erinnerung: ein Pfad ist eine endliche Folge von Knoten, die jeweils durch Kanten verbunden sind: $(u_0, u_1, \ldots, u_\ell)$ ist ein Pfad der Länge ℓ wenn $(u_i, u_{i+1}) \in E$ für $0 \leqslant i < \ell$. Ein unendlicher Pfad ist eine nicht-abbrechende Folge $(u_i)_{i\in\mathbb{N}}$ dieser Art.)

Ein Baum $\mathcal{T}$ heißt *endlich verzweigt* gdw. für jeden Knoten $u \in V$ die direkte Nachfolgermenge $E[u] := \{v \in V : (u, v) \in E\}$ endlich ist.

Lemma 4.4 (Lemma von König) *Sei $\mathcal{T}$ ein unendlicher, aber endlich verzweigter Baum. Dann gibt es in $\mathcal{T}$ einen unendlichen Pfad (von der Wurzel aus).*

Bem.: Es ist klar, dass $\mathcal{T}$ *beliebig lange endliche Pfade* haben muss; *daraus* folgt aber keineswegs, dass es einen *unendlich langen Pfad* gibt. Man mache sich den Unterschied an einem Beispiel klar.

Beweis Wir zeigen die Behauptung für $\mathcal{T} = (V, E, \lambda)$ mit abzählbar unendlicher Knotenmenge V durch Kompaktheit von $\mathrm{AL}(\mathcal{V})$ für $\mathcal{V} := \{p_u : u \in V\}$.

Für $\varphi_u := p_u \to \bigvee\{p_v : v \in E[u]\}$ ([2]) setze

$$\Phi := \{\varphi_u : u \in V\} \cup \{p_\lambda\} \subseteq \mathrm{AL}(\mathcal{V}).$$

Jede endliche Teilmenge von Φ ist erfüllbar: Sei $\Phi_0 \subseteq \Phi$ endlich, n die maximale Tiefe von Knoten u, für die p_u in Φ_0 vorkommt. Dann ist $\mathfrak{I} \models \Phi_0$ für die Interpretation $\mathfrak{I}$, die gerade den p_u längs der Knoten u irgendeines Pfades der Länge n von λ aus den Wert 1 zuweist. Also ist Φ erfüllbar (Kompaktheit). Aus $\mathfrak{I} \models \Phi$ finden wir (induktiv) einen unendlichen Pfad $u_0, u_1, u_2, \ldots$ mit der Bedingung, dass $\mathfrak{I}(u_i) = 1$ für alle $i \in \mathbb{N}$:

$i = 0$: $u_0 := \lambda$ erfüllt Bedingung, da $\mathfrak{I} \models p_\lambda$.

Sei u_i mit $\mathfrak{I}(u_i)$ bereits gewählt. Dann folgt aus $\mathfrak{I} \models \varphi_u$ für $u = u_i$ und, da $\mathfrak{I}(u_i) = 1$, dass $\mathfrak{I} \models \bigvee\{p_v : v \in E[u_i]\}$, d.h. es gibt mindestens ein $v \in E[u_i]$ mit $\mathfrak{I}(v) = 1$, das wir als u_{i+1} wählen können. □

Übung 4.5 [k-Färbbarkeit*] Wir betrachten einen Graph $\mathcal{G} = (V, E)$ mit Knotenmenge V und Kantenrelation $E \subseteq V \times V$. Eine k-Färbung von V ist eine Abbildung $F : V \to \{1, \ldots, k\}$ derart, dass für alle $(u, v) \in E$ gilt: $F(u) \neq F(v)$ (benachbarte Knoten haben verschiedene Farben). Eine k-Färbung einer Teilmenge $V_0 \subseteq V$ der Knotenmenge ist genauso definiert.

Zeigen Sie: Ein Graph mit abzählbar unendlicher Knotenmenge V hat genau dann eine k-Färbung, wenn es auf jeder endlichen Teilmenge $V_0 \subseteq V$ eine k-Färbung gibt.

Hinweis: Um AL-Kompaktheit anzuwenden, kodiert man Knotenfarben durch Aussagenvariablen und beschreibt die Bedingungen an eine k-Färbung durch AL-Formeln.

[2] φ_u verlangt dass p_u falsch ist für Knoten u ohne Nachfolger (dead ends) und ebenso für Knoten u für deren Nachfolger v sämtlich p_v falsch ist.

Logikkalküle: Deduktion und Refutation

Logik*kalküle* stellen den semantischen Konzepten von Folgerungsbeziehung und Allgemeingültigkeit syntaktische Konzepte von formalen Beweisen gegenüber.

Formale Beweise sind syntaktische Zeichenketten (z.B. ihrerseits aus Formeln aufgebaut), die einfach nachprüfbaren syntaktischen Regeln gehorchen (das Regelsystem heißt hier *Kalkül*). Regelkonforme formalen Beweise bezeichnet man auch als Ableitungen (vgl. z.B. Ableitungen in Grammatiken). Die formalen Beweise sollen das inhaltlich korrekte Schließen wiedergeben, in dem Sinne dass

[**Korrektheit**] nur semantisch korrekte Sachverhalte formal beweisbar (ableitbar) sind.

[**Vollständigkeit**] jeder semantisch korrekte Sachverhalt einen formalen Beweis besitzt.

Wir behandeln für AL hier zwei ganz unterschiedliche Kalküle: einen *Widerlegungs-Kalkül* (Refutation) für den Nachweis der *Unerfüllbarkeit* von KNF-Formeln; und einen *Deduktionskalkül* für den Nachweis der Allgemeingültigeit von AL-Formeln.

5 AL Resolution

Der Resolutionskalkül ist ein Kalkül, der auf den Nachweis der *Unerfüllbarkeit* von AL-Formeln in KNF gerichtet ist. (Widerlegung/Refutation: Ableitungen im Resolutionskalkül sind formale, syntaktische Beweise für die Unerfüllbarkeit.)

5.1 KNF und Klauseln

Alternatives Format für KNF: Klauselmengen.

Erinnerung: Literale sind AL-Formeln der Form p (positives Literal) oder $\neg p$ (negatives Literal). Schreibweisen: i.d.R. L, L_i, usw. für Literale.

Ist L ein Literal, so bezeichnet $\overline{L}$ das zu $\neg L$ äquivalente Literal:

$$\overline{L} := \begin{cases} \neg p & \text{für positives Literal } L = p \\ p & \text{für negatives Literal } L = \neg p. \end{cases}$$

Definition 5.1 [Klauseln]
Eine *Klausel* ist eine endliche Menge von Literalen. $\Box$ bezeichnet die leere Klausel. Wir assoziieren mit der Klausel $C = \{L_1, \ldots, L_k\}$ die Disjunktion $\bigvee C \equiv L_1 \vee \ldots \vee L_k$.

Bsp. von Klauseln C: $\Box$ (leer) mit $\Box \equiv \bigvee \emptyset \equiv 0$ (unerfüllbar[3]); $C = \{p, q, \neg r\} \equiv (p \vee q \vee \neg r)$; $C = \{p, q, \neg p\} \equiv (p \vee q \vee \neg p) \equiv 1$.

Eine endliche *Klauselmenge* steht für die Konjunktion ihrer Klauseln. Für $K = \{C_1, \ldots, C_\ell\}$ ist $K \equiv C_1 \wedge \ldots \wedge C_\ell$ eine KNF-Formel. Zu jeder KNF-Formel gehört umgekehrt eine äquivalente endliche Klauselmenge.

Bsp. von Klauselmengen K: $K = \emptyset$ (leer) mit $K \equiv \bigwedge \emptyset \equiv 1$ (allgemeingültig[4]); $K = \{\Box, \{p, q\}\}$; $K = \{\{p, q, \neg r\}, \{r, \neg p\}, \{\neg r, q\}\}$.

Mit der Korrespondenz:

endliche Klauselmengen $\approx$ KNF-Formeln

[3] merke: leere Klausel = *keine Alternative*, unerfüllbar, immer falsch.
[4] merke: leere Klauselmenge = *keine Bedingungen*, immer wahr, allgemeingültig.

sind z.B. $\mathfrak{I} \models K$, Erfüllbarkeit von K usw. wohldefiniert für Klauselmengen K. Ebenso für eine einzelne Klausel C, die wir mit der Klauselmenge $\{C\}$ identifizieren können.

Beispiel 5.2 Für Literal L, Klauseln C, C' und Klauselmengen K, K':

(i) Ist $C \subseteq C'$ so gilt $C \models C'$;
für $K \subseteq K'$ gilt $K' \models K$.

(ii) Ist $\{L, \overline{L}\} \subseteq C$, so ist C allgemeingültig;
wenn $C \in K$, so ist K äquivalent zu $K \setminus \{C\}$.

(iii) $C = \Box$ ist unerfüllbar;
ist $\Box \in K$, so ist K unerfüllbar.

(iv) $K = \emptyset$ ist allgemeingültig.

5.2 Resolutionskalkül

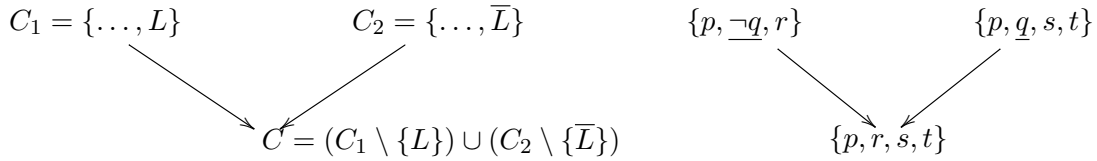
Beispiel 5.3 Betrachte Formeln $\varphi \vee p$ und $\psi \vee \neg p$, wobei p nicht in φ und ψ vorkomme. Dann ist $(\varphi \vee p) \wedge (\psi \vee \neg p)$ erfüllbar genau dann, wenn $\varphi \vee \psi$ erfüllbar ist.

Begründung: "⇒": Sei $\mathfrak{I} \models (\varphi \vee p) \wedge (\psi \vee \neg p)$, und z.B. $\mathfrak{I}(p) = 1$. Dann folgt $\mathfrak{I} \models \psi$ und demnach $\mathfrak{I} \models \varphi \vee \psi$. Der Fall $\mathfrak{I}(p) = 0$ ist analog.

"⇐": Sei $\mathfrak{I} \models \varphi \vee \psi$. Betrachte den Fall, dass $\mathfrak{I} \models \varphi$ (der Fall $\mathfrak{I} \models \psi$ ist analog). Da p in φ nicht vorkommt, kommt es auf $\mathfrak{I}(p)$ nicht an; sei $\mathfrak{I}' := \mathfrak{I}[p \mapsto 0]$ die Abwandlung von $\mathfrak{I}$ mit $\mathfrak{I}'(p) = 0$ und $\mathfrak{I}'(q) = \mathfrak{I}(q)$ für alle $q \neq p$. Dann ist auch $\mathfrak{I}' \models \varphi$ und demnach $\mathfrak{I}' \models \varphi \vee p$; aber auch $\mathfrak{I}' \models \psi \vee \neg p$, da $\mathfrak{I}' \models \neg p$. Also $\mathfrak{I}' \models (\varphi \vee p) \wedge (\psi \vee \neg p)$.

Definition 5.4 [Resolvente]
Für Klauseln C_1, C_2, C:
C ist eine *Resolvente* von C_1 und C_2, wenn für ein Literal L gilt:

$$L \in C_1, \quad \overline{L} \in C_2, \quad \text{und } C = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{\overline{L}\}).$$



Lemma 5.5 (Resolutionslemma) *Für Klauselmenge K, Klauseln $C_1, C_2 \in K$: Ist C eine Resolvente von C_1 und C_2, so ist $K \equiv K \cup \{C\}$.*

Insbesondere ändert die Hinzunahme von Resolventen nicht die Erfüllbarkeit.

Beweis Wir zeigen: $\mathfrak{I} \models K$ gdw. $\mathfrak{I} \models K \cup \{C\}$.
"⇐" ist trivial.
"⇒": sei $C = (C_1 \setminus \{L\}) \cup (C_2 \setminus \{\overline{L}\})$ und $\mathfrak{I} \models K$.
Fall 1: $\mathfrak{I} \models \overline{L}$. Dann folgt aus $\mathfrak{I} \models C_1$, dass auch $\mathfrak{I} \models C_1 \setminus \{L\}$ und also $\mathfrak{I} \models C$.
Fall 2: $\mathfrak{I} \models L$. Dann folgt aus $\mathfrak{I} \models C_2$, dass auch $\mathfrak{I} \models C_2 \setminus \{\overline{L}\}$ und also $\mathfrak{I} \models C$. □

Definition 5.6
$\mathrm{Res}(K) := K \cup \{C\colon\ C$ Resolvente von Klauseln in $K\ \}$. Eine Klausel C heißt (im Resolutionskalkül) *ableitbar* aus K, wenn sie durch iterierte Anwendung von Res auf K gewonnen wird. $\mathrm{Res}^*(K)$ ist die Menge aller aus K ableitbaren Klauseln.

$\mathrm{Res}^*(K)$ lässt sich induktiv gewinnen als

$$\begin{array}{rrcl} & \mathrm{Res}^*(K) & = & \bigcup_{n\in\mathbb{N}} \mathrm{Res}^n(K), \\ \text{wo} & \mathrm{Res}^0(K) & := & K, \\ & \mathrm{Res}^{n+1}(K) & := & \mathrm{Res}\big(\mathrm{Res}^n(K)\big). \end{array}$$

Übung 5.7 Man zeige, dass obige Menge $\bigcup_{n\in\mathbb{N}} \mathrm{Res}^n(K)$ die kleinste Menge von Klauseln ist, die K umfasst und unter Hinzunahme von Resolventen abgeschlossen ist.

Bem.: Man mache sich an Beispielen klar, dass keineswegs alle Klauseln C mit $K \models C$ im Resolutionskalkül aus K ableitbar sind.

5.3 Korrektheit und Vollständigkeit

Ziel des Resolutionskalküls ist der Nachweis der Unerfüllbarkeit einer Klauselmenge K durch Ableitung von $\Box$ aus K. Wünschenswert sind dabei:

Korrektheit: $\Box$ ist nur dann ableitbar aus K, wenn K unerfüllbar ist.

Vollständigkeit: Wenn K unerfüllbar ist, so lässt sich $\Box$ aus K ableiten.

Korrektheit des Resolutionskalküls verlangt also, dass

$$\Box \in \mathrm{Res}^*(K) \quad \Longrightarrow \quad K \text{ unerfüllbar.}$$

Das folgt direkt aus dem Lemma, denn iterierte Anwendung liefert $K \equiv \mathrm{Res}^*(K)$. Ist $\Box \in \mathrm{Res}^*(K)$, so ist $\mathrm{Res}^*(K)$ unerfüllbar ($\Box \equiv 0$), und also K ebenfalls unerfüllbar.

Vollständigkeit verlangt, dass umgekehrt

$$K \text{ unerfüllbar} \quad \Longrightarrow \quad \Box \in \mathrm{Res}^*(K).$$

Lemma 5.8 *Ist die Klauselmenge K über Aussagenvariablen $\mathcal{V}_n = \{p_1, \ldots, p_n\}$ unerfüllbar, so ist $\Box \in \mathrm{Res}^*(K)$.*

Beweis (durch Induktion über n)
Induktionsanfang: $n = 0$ ist trivial (es gibt nur die leere Klausel, also $K = \emptyset$ (erfüllbar) oder $K = \{\Box\}$ (unerfüllbar) und $\Box \in K = \mathrm{Res}^*(K)$.

Induktionsschritt von n nach $n+1$:

Sei $K = \{C_1, \ldots, C_k\}$ in Variablen $\mathcal{V}_{n+1}$ unerfüllbar. Aus K gewinne zwei Klauselmengen K_0 und K_1 in Variablen $\mathcal{V}_n$ (die der Fixierung von $p_{n+1} := 0$ bzw. $p_{n+1} := 1$ entsprechen). Sei $L = p_{n+1}$.

K_0: streiche das Literal L aus allen Klauseln,
streiche alle Klauseln, die das Literal $\overline{L}$ enthalten.

K_1: streiche das Literal $\overline{L}$ aus allen Klauseln,
streiche alle Klauseln, die das Literal L enthalten.

Man vergewissere sich, dass $K_0 \equiv K \cup \{\neg p_{n+1}\}$ und $K_1 \equiv K \cup \{p_{n+1}\}$.

Dann gilt:

(1) K_0 und K_1 sind beide unerfüllbar. Sonst erhält man z.B. aus einer $\mathcal{V}_n$-Interpretation, die K_0 erfüllt, eine $\mathcal{V}_{n+1}$-Interpretation die K erfüllt indem man $\mathfrak{I}(p_{n+1}) = 0$ setzt. Demnach

(2) (nach Induktionsannahme) $\square \in \mathrm{Res}^*(K_0)$ und $\square \in \mathrm{Res}^*(K_1)$.

In der Ableitung von $\square$ aus K_0 können Klauseln aus K verwendet worden sein, aus denen L gestrichen wurde. Wenn nicht, so wurden nur Klauseln aus K benutzt und $\square \in \mathrm{Res}^*(K)$ folgt sofort. Andernfalls fügt man L längs aller Ableitungsschritte wieder hinzu und erhält eine Ableitung der Klausel $\{L\}$ aus K, also $\{L\} \in \mathrm{Res}^*(K)$.

Genauso folgt aus $\square \in \mathrm{Res}^*(K_1)$ dass entweder sogar $\square \in \mathrm{Res}^*(K)$ oder dass $\{\overline{L}\} \in \mathrm{Res}^*(K)$.

Im verbleibenden Fall sind also $\{L\}, \{\overline{L}\} \in \mathrm{Res}^*(K)$, und damit durch einen weiteren Resolutionsschritt auch $\square \in \mathrm{Res}^*(K)$. $\square$

Bem.: Man muss die Endlichkeit von K nicht voraussetzen. Der Fall unendlicher K (mit unendlich vielen Variablen) reduziert sich durch Kompaktheit auf den endlichen Fall.

Definition 5.9 Ein Beweis im Resolutionskalkül ist ein binärer Baum, dessen Knoten mit Klauseln beschriftet sind, wobei

(i) an den inneren Knoten jeweils eine Resolvente der Klauseln an den Nachfolgerknoten steht;

(ii) an der Wurzel die leere Klausel $\square$ steht.

Ein solcher Baum ist ein Beweis der Unerfüllbarkeit der Klauselmenge, die aus den Klauseln an den Blättern besteht.

Ein Resolutionsbeweis ist ein 'Zertifikat für die Unerfüllbarkeit'.

Wir haben gesehen, dass genau die tatsächlich unerfüllbaren Klauselmengen einen solchen Unerfüllbarkeitsbeweis besitzen.

Resolutionsalgorithmus Da für eine endliche Klauselmenge K die iterative Anwendung von Res nach endlich vielen Schritten nichts neues mehr produziert, bekommt man einen terminierenden (!) Resolutionsalgorithmus (der aber im schlimmsten Fall exponentiell viele Schritte benötigt):

Eingabe: K [Klauselmenge, endlich]
$R := K$
WHILE $\big(\mathrm{Res}(R) \neq R$ and $\square \notin R\big)$ DO $R := \mathrm{Res}(R)$ OD
IF $\square \in R$ THEN output "unerfüllbar"
ELSE output "erfüllbar"

5.4 Hornklauseln und Einheitsresolution

Hornklauseln sind Klauseln mit höchstens einem positiven Literal. Dieses spezielle logische Format spielt vor allem im Kontext von Datenbanksprachen (Datalog) und Logischer Programmierung eine besondere Rolle.

Wichtig: die Klausel $C = \{\neg q_1, \ldots, \neg q_r, q\}$ ist äquivalent zur Implikation

$$\big(q_1 \wedge \ldots \wedge q_r\big) \to q, \qquad (*)$$

die verlangt, dass q wahr gemacht wird wenn sämtliche q_i wahr sind.

Definition 5.10 [Hornklauseln]
Eine Klausel mit höchstens einem positiven Literal heißt *Hornklausel*.

Zwei Spezialfälle:
C besteht nur aus einem positiven Literal: *positive Hornklausel*;
C enthält kein positives Literal: *negative Hornklausel*.

Die typische Hornklausel $(*)$ ist weder positiv noch negativ.
Wichtig: Nicht jede AL-Formel ist äquivalent zu einer Hornklauselmenge!

Übung 5.11 Man zeige, dass jede Hornklauselmenge, die keine negativen Hornklauseln enthält, erfüllbar ist; ebenso für Hornklauselmengen, die keine positiven Hornklauseln enthalten.

Effizienter Horn-Erfüllbarkeitstest Erfüllbarkeit für Hornklauselmengen ist effizient entscheidbar. Die folgenden Beobachtungen führen zu einem polynomialen Erfüllbarkeitstest.

Im ersten Schritt berechnen wir für eine Menge von nicht-negativen Hornklauseln eine eindeutig bestimmte *minimale* Belegung: jede Variable, die von dieser Belegung wahr gemacht wird, muss in *jedem* Modell der Klauselmenge wahr sein.

Lemma 5.12 *Sei H_0 eine Hornklauselmenge ohne negative Klauseln. Dann gibt es eine eindeutig bestimmte minimale Belegung $\mathfrak{I}_0$ der Variablen in H_0, die H_0 erfüllt. $\mathfrak{I}_0$ ist durch einen Algorithmus mit polynomialer Laufzeit berechenbar.*

Beweis Wir behandeln H_0 in Variablen $\mathcal{V}_n$. Durch eine Iteration mit höchstens n Schritten generieren wir die Menge $\mathcal{X} \subseteq \mathcal{V}_n$ von Variablen, die notwendig in *jeder* erfüllenden Belegung wahr gemacht werden müssen. Für $C = \{\neg q_1, \ldots, \neg q_r, q\} \in H$ und $\mathcal{X} \subseteq \mathcal{V}$ sei

$$C(\mathcal{X}) := \begin{cases} \{q\} & \text{wenn für } i = 1, \ldots, r \text{ bereits } q_i \in \mathcal{X}, \\ \emptyset & \text{sonst.} \end{cases}$$

Beachte, dass im Fall $r = 0$ (d.h. für positive C) der obere Fall gilt. Für $\mathcal{X} = \emptyset$ ist $C(\mathcal{X}) \neq \emptyset$ genau für positive C.

$$H_0(\mathcal{X}) := \mathcal{X} \cup \textstyle\bigcup_{C \in H_0} C(\mathcal{X}),$$

wobei speziell für $\mathcal{X} = \emptyset$ also herauskommt: $H_0(\emptyset) = \{q \in \mathcal{V} \colon \{q\} \in H_0\}$.

Eingabe: H_0 [Hornklauselmenge, endlich, nicht-negativ]

$\mathcal{X} := \emptyset$
WHILE $H_0(\mathcal{X}) \neq \mathcal{X}$ DO $\mathcal{X} := H_0(\mathcal{X})$ OD

Sei $\mathcal{X}_0 = \emptyset$, $\mathcal{X}_{n+1} = H_0(\mathcal{X}_n)$ das $(n+1)$-te Stadium dieser Iteration, $\mathcal{X}_\infty$ das Ergebnis der Iteration. Zu jedem $\mathcal{X}$ sei $\mathfrak{I}_\mathcal{X}$ die Interpretation mit $\mathfrak{I}_\mathcal{X}(q) = 1$ gdw. $q \in \mathcal{X}$. Dann gilt (Beweise!):

(i) $\mathcal{X}_0 \subseteq \mathcal{X}_1 \subseteq \cdots \subseteq \mathcal{X}_\infty$.
(ii) Die Iteration terminiert innerhalb von n Schritten ($\mathcal{X}_\infty = \mathcal{X}_n$), wenn H_0 nur Variablen aus $\mathcal{V}_n$ hat.
(iii) Für jedes $\mathfrak{I} \models H_0$, und alle n: $q \in \mathcal{X}_n \;\Rightarrow\; \mathfrak{I}(q) = 1$.
(iv) Für $\mathcal{X} = \mathcal{X}_\infty$: $\mathfrak{I}_\mathcal{X} \models H_0$.

Also ist $\mathfrak{I}_0 := \mathfrak{I}_\mathcal{X}$ für $\mathcal{X} = \mathcal{X}_\infty$ wie behauptet. □

Im zweiten Schritt muss nur noch getestet werden, ob auch die negativen Klauseln in H von $\mathfrak{I}_0$ erfüllt werden:

Lemma 5.13 *Für eine Hornklauselmenge H mit Menge von negativen Klauseln $H^- \subseteq H$ und nicht-negativen Klauseln $H_0 := H \setminus H^-$, sei $\mathfrak{I}_0$ die minimale Interpretation, die H_0 erfüllt. Dann ist H erfüllbar gdw. $\mathfrak{I}_0 \models H$.*

Beweis Sei H erfüllbar, $\mathfrak{I} \models H$. Dann ist $\mathfrak{I} \models H_0$, also muss $\mathfrak{I}$ alle Variablen wahr machen, die von $\mathfrak{I}_0$ wahr gemacht werden. Da $\mathfrak{I} \models H^-$, erfüllt dann aber $\mathfrak{I}_0$ erst recht alle (negativen!) Klauseln in H^-, also folgt $\mathfrak{I}_0 \models H$.

Die umgekehrte Richtung ist trivial: wenn $\mathfrak{I}_0 \models H$, so ist H erfüllbar. □

Aus den vorangehenden beiden Aussagen ergibt sich ein polynomialer Erfüllbarkeitstest für Hornklauselmengen.

Beispiel 5.14 Betrachte einen Graphen $\mathcal{G} = (V, E)$ mit abzählbarer Knotenmenge V und Kantenrelation $E \subseteq V \times V$. Zu $\mathcal{V} := \{p_u \colon u \in V\}$ und $u \in V$ sei C_u die zu

$$\big(\bigwedge\{p_v \colon v \in E[u]\}\big) \to p_u$$

äquivalente Hornklausel, und $H_0 := \big\{C_u \colon u \in V\big\}$. Beachte, dass $C_u \equiv p_u$ für Knoten u ohne E-Nachfolger.

Man überlegt sich, dass p_u von der minimalen Interpretation für H_0 wahr gemacht wird, gdw. es von u aus keine unendlichen E-Pfade gibt.

Beobachtung 5.15 [Einheitsresolution]
Die Einschränkung des Resolutionskalküls auf Resolutionsschritte, in denen eine der Eingangsklauseln aus nur einem Literal besteht (Einheitsresolution), ist vollständig (und korrekt) für Hornklauselmengen.

Übung 5.16 Zeigen Sie die Vollständigkeit von Einheitsresolution für Hornklauselmengen H über $\mathcal{V}_n$: Ist H unerfüllbar, so existiert eine Ableitung der leeren Klausel $\Box$ aus H, die nur Einheitsresolutionsschritte verwendet.

6 AL Sequenzenkalkül

Der Sequenzenkalkül ist ein Kalkül für das formale, syntaktische Beweisen von Folgerungsbeziehungen (oder allgemeingültigen Implikationen).

6.1 Sequenzen

Definition 6.1 [AL-Sequenzen]
Eine AL-Sequenz ist ein Paar von endlichen Formelmengen, (Γ, Δ), $\Gamma, \Delta \subseteq \mathrm{AL}$; auch als $\Gamma \vdash \Delta$ notiert. Eine Sequenz $\Gamma \vdash \Delta$ heißt *allgemeingültig*, wenn $\bigwedge \Gamma \models \bigvee \Delta$ gilt.

Bem.: Die linke Seiten einer Sequenz wird als Konjunktion (von Voraussetzungen) gelesen; die rechte Seite als Disjunktion (und ist eine Folgerung aus den Voraussetzungen, sofern die Sequenz allgemeingültig ist).

Bem.: Die Sequenz $\emptyset \vdash \{\varphi\}$ ist allgemeingültig, gdw. φ allgemeingültig ist.

Schreibweisen: Wir fassen die Mengen von Formeln auch als Liste auf, wobei es nicht auf die Reihenfolge ankommt. Daher schreiben wir auch Γ, φ anstelle von $\Gamma \cup \{\varphi\}$; φ anstelle von $\{\varphi\}$, usw.

6.2 Sequenzenkalkül

Der Sequenzenkalkül besteht aus Sequenzenregeln für die Erzeugung neuer Sequenzen aus bereits erzeugten Sequenzen. Eine Sequenzenregel erlaubt es, aus gegebenen Sequenzen eine weitere Sequenz zu gewinnen (abzuleiten).

Sprechweisen: gegebenen Sequenzen = Prämissen der Regel;
neue abgeleitete Sequenz = Konklusion der Regel.

Notation diagrammatisch:

$$\text{(Regel)}\quad \frac{\text{Prämissen}}{\text{Konklusion}}$$

Der AL-Sequenzenkalkül hat Regeln ohne Prämissen (deren Konklusionen heißen Axiome), Regeln mit einer Prämisse und Regeln mit zwei Prämissen. Wir bezeichnen den folgenden Sequenzenkalkül für AL mit $\mathcal{SK}$.

Die folgenden Regeln sind jeweils für beliebige $\Gamma, \Delta, \varphi, \psi$ gemeint:

$$\text{(Ax)}\quad \frac{}{\Gamma, \varphi \vdash \Delta, \varphi}$$

$$\text{(0-Ax)}\quad \frac{}{\Gamma, 0 \vdash \Delta} \qquad \text{(1-Ax)}\quad \frac{}{\Gamma \vdash \Delta, 1}$$

$$\text{(}\neg\text{L)}\quad \frac{\Gamma \vdash \Delta, \varphi}{\Gamma, \neg\varphi \vdash \Delta} \qquad \text{(}\neg\text{R)}\quad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta, \neg\varphi}$$

$$\text{(}\vee\text{L)}\quad \frac{\Gamma, \varphi \vdash \Delta \qquad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta} \qquad \text{(}\vee\text{R)}\quad \frac{\Gamma \vdash \Delta, \varphi, \psi}{\Gamma \vdash \Delta, \varphi \vee \psi}$$

$$\text{(}\wedge\text{L)}\quad \frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta} \qquad \text{(}\wedge\text{R)}\quad \frac{\Gamma \vdash \Delta, \varphi \qquad \Gamma \vdash \Delta, \psi}{\Gamma \vdash \Delta, \varphi \wedge \psi}$$

Definition 6.2 Eine Sequenz heißt *ableitbar* im Sequenzenkalkül, falls sie (in endlich vielen Schritten) durch Anwendung von Sequenzenregeln (ausgehend von Axiomen, d.h. Regeln ohne Prämissen) als Konklusion gewonnen werden kann.

Beispiel 6.3 Eine Ableitung der allgemeingültigen Sequenz $p \vdash (p \wedge q) \vee \neg q$:

$$\cfrac{\cfrac{\cfrac{}{p \vdash p, \neg q\ (\text{Ax})} \qquad \cfrac{\cfrac{}{p, q \vdash q\ (\text{Ax})}}{p \vdash q, \neg q\ (\neg\text{R})}}{p \vdash (p \wedge q), \neg q\ (\wedge\text{R})}}{p \vdash (p \wedge q) \vee \neg q\ (\vee\text{R})}$$

6.3 Korrektheit und Vollständigkeit

Für den Sequenzenkalkül bedeuten

Korrektheit: jede ableitbare Sequenz ist allgemeingültig.

Vollständigkeit: jede allgemeingültige Sequenz ist ableitbar.

Übung 6.4 Nachweis der Korrektheit. Man muss lediglich überprüfen, dass jede einzelne Regelanwendung die Allgemeingültigkeit erhält:

(a) die Axiome sind allgemeingültige Sequenzen.

(b) für alle Regeln mit Prämissen gilt: sind die Prämissen allgemeingültig, so auch die Konklusion.

Beobachtung 6.5 Für die Regeln des AL-Sequenzenkalküls mit Prämissen gilt sogar die stärkere Variante von (b): die Konklusion ist allgemeingültig *genau* dann, wenn alle Prämissen allgemeingütig sind.

Vollständigkeit folgt dann aus der folgenden Bemerkung zur "Beweissuche".

Definition 6.6 Ein *formaler Beweis* im AL-Sequenzenkalkül ist ein Baum, dessen Knoten mit Sequenzen beschriftet sind, wobei

(i) an inneren Knoten die Konklusionen von Regeln stehen, deren Prämissen an den Nachfolgerknoten stehen;

(ii) an den Blättern Axiome stehen.

Ein solcher Baum ist ein Beweis der Sequenz, die an der Wurzel steht. Eine Sequenz ist ableitbar gdw. wenn sie in diesem Sinne einen Beweis besitzt.

Ein Beweis im Sequenzenkalkül ist also ein 'Zertifikat für Allgemeingültigkeit'.

Um zu testen, ob eine gegebene Sequenz ableitbar ist, kann man eine systematische Suche nach dem Beweis (auch algorithmisch) durchführen. Im Fall des AL-Sequenzenkalküls gibt es ein terminierendes (!) Verfahren, das alle potentiellen Beweisbäume rückwärts von der Wurzel (dem Beweisziel) her konstruiert.

Dazu beachte, dass es für jede nicht-atomare Formel in einer Sequenz immer eine Regel gibt, die diese gegebene Sequenz als Konklusion hat, und die die betrachtete Formel abbaut (den führenden Junktor eliminiert). So kann eine Beweissuche von der Zielsequenz aus rückwärts fortgesetzt werden, bis nur noch Sequenzen aus atomaren Formeln an den Blättern stehen. Sind alle diese Blatt-Sequenzen Axiome, so hat man einen Beweis gefunden; ist mindestens eine davon kein Axiom, so ist sie auch nicht allgemeingültig (nachprüfen!), und mit der Beobachtung vererbt sich das durch den Baum auf die Zielsequenz an der Wurzel.

Damit ist auch die Vollständigkeit des AL-Sequenzenkalküls bewiesen.

Beispiel 6.7 Systematische Beweissuche zur nicht allgemeingültigen Sequenz $(p \vee q) \vdash p \wedge q$ führt auf zwei Blätter mit atomaren Sequenzen, die nicht allgemeingültig sind. Man liest aus diesen ab, dass die Interpretationen mit $p \mapsto 1, q \mapsto 0$ bzw. $q \mapsto 1, p \mapsto 0$ beide die Zielsequenz nicht erfüllen.

$$\dfrac{\dfrac{\dfrac{}{p \vdash p}\,(\mathrm{Ax}) \qquad \mathbf{p \vdash q}}{p \vdash p \wedge q}\,(\wedge\mathrm{R}) \qquad \dfrac{\mathbf{q \vdash p} \qquad \dfrac{}{q \vdash q}\,(\mathrm{Ax})}{q \vdash p \wedge q}\,(\wedge\mathrm{R})}{p \vee q \vdash p \wedge q}\,(\vee\mathrm{L})$$

Alternativ (wenn man zuerst die rechte Seite abbaut) mit demselben Ergebnis:

$$\dfrac{\dfrac{\dfrac{}{p \vdash p}\,(\mathrm{Ax}) \qquad \mathbf{q \vdash p}}{p \vee q \vdash p}\,(\vee\mathrm{L}) \qquad \dfrac{\mathbf{p \vdash q} \qquad \dfrac{}{q \vdash q}\,(\mathrm{Ax})}{p \vee q \vdash q}\,(\vee\mathrm{L})}{p \vee q \vdash p \wedge q}\,(\wedge\mathrm{R})$$

6.4 Schnittregeln und erweiterte Sequenzenkalküle

Wir diskutieren zusätzlich zum Kernkalkül $\mathcal{SK}$ eine Erweiterung $\mathcal{SK}^+$ um eine sogenannte *Schnittregel*. Die Schnittregel mit dem klassischen lateinischen Namen "*modus ponens*" (und daraus ableibare weitere Regeln) sind offensichtlich korrekt. Wie wir gesehen haben, braucht man die Schnittregel nicht für die Vollständigkeit. Andererseits erlaubt es die Verwendung der Schnittregel (und daraus abgeleiteter Schlussfiguren), typische mathematische Beweisfiguren viel natürlicher nachzugestalten. Die Regel modus ponens entspricht gerade der Verwendung von Zwischenbehauptungen (Lemmata) und Kettenschlüssen in Beweisen:

$$\text{(modus ponens)} \quad \frac{\Gamma \vdash \varphi \qquad \Gamma', \varphi \vdash \Delta}{\Gamma, \Gamma' \vdash \Delta}$$

Hier fungiert φ in der Prämisse $\Gamma', \varphi \vdash \Delta$ als "Hilfsannahme", die mittels der weiteren Prämisse $\Gamma \vdash \varphi$ zugunsten der Aufnahme von Γ auf der linken Seite eliminiert wird. In mathematischen Beweisen entspricht das der Verwendung von Kettenschlüsse: aus Beweisen von A $\Rightarrow$ B und B $\Rightarrow$ C ergibt sich ein Beweis von A $\Rightarrow$ C (Verwendung von B als Lemma). In der obigen Regel entspricht dies dem Fall dass $\Gamma' = \emptyset$ ist.

Übung 6.8 Weisen Sie die Korrektheit der modus ponens Regel nach.

Hinzunahme von modus ponens zum *schnittfreien Kalkül* $\mathcal{SK}$ verändert den Kalkül qualitativ in dem Sinne, dass systematische Beweissuche (die im Schnittfreien Kalkül $\mathcal{SK}$ quasi eindeutig rückwärts durchgeführt werden kann) nicht mehr möglich ist. Dies liegt daran, dass die Konklusion der Regel keine Information darüber enthält, welche Zwischenbehauptung φ verwandt worden sein könnte. Entsprechend gilt auch das Analogon von Beobachtung 6.5 nicht für den erweiterten Kalkül.

Aus der Vollständigkeit von $\mathcal{SK}$ folgt, dass jeder formale Beweis (in der AL), der modus ponens verwendet, alternativ auch ohne Verwendung von modus ponens durchgeführt werden könnte – das sagt aber nicht, dass es eine unmittelbare "lokale" Strategie für die Elimination von Kettenschlüssen gäbe!

Als direkt mit modus ponens und den übrigen Regeln von $\mathcal{SK}$ ableitbare Regel erhalten wir die folgende *Widerspruchsregel* (Kontradiktion).

$$\text{(Kontr)} \quad \frac{\Gamma \vdash \varphi \qquad \Gamma' \vdash \neg\varphi}{\Gamma, \Gamma' \vdash \emptyset}$$

Die Regel (Kontr), die sich ebenfalls nicht direkt in $\mathcal{SK}$ nachbilden lässt, ist direkt aus $\mathcal{SK}$ und modus ponens ableitbar wie folgt:

$$\text{(modus ponens)}\ \dfrac{\dfrac{\dfrac{\Gamma \vdash \varphi}{\Gamma, \neg\varphi \vdash \emptyset}\,(\neg\text{L})}{\Gamma \vdash \neg\neg\varphi}\,(\neg\text{R}) \qquad \dfrac{\Gamma' \vdash \neg\varphi}{\Gamma', \neg\neg\varphi \vdash \emptyset}\,(\neg\text{L})}{\Gamma, \Gamma' \vdash \emptyset}$$

Die Regel (Kontr) liefert eine Entsprechung zur Schlussfigur des *indirekten Beweises* in mathematischen Beweisen. Man weist A nach, indem man zeigt, dass $\neg A$ zum Widerspruch führt; in der Regel (Kontr) setze $\Gamma = \Gamma' := \neg A$ um aus dem Widerspruch, der sich aus $A \Rightarrow \varphi$ und $A \Rightarrow \neg\varphi$ ergibt, zu erhalten, dass $\neg A \to \bot$, also die Gültigkeit von

A. Den letztgenannten Zusammenhang liefert die ebenfalls mit modus ponens ableitbare Regel

$$\frac{\Gamma, \neg\varphi \vdash \emptyset}{\Gamma \vdash \varphi}$$

die einem Spezialfall der Umkehrung der Regel (¬L) entspricht. Eine Ableitung dieser Regel auf der Basis von $\mathcal{SK}$ und modus ponens:

$$\dfrac{\dfrac{\Gamma, \neg\varphi \vdash \emptyset}{\Gamma \vdash \neg\neg\varphi}\,(\neg\mathrm{R}) \qquad \dfrac{\dfrac{\dfrac{}{\Gamma, \varphi \vdash \varphi}\,(\mathrm{Ax})}{\Gamma \vdash \varphi, \neg\varphi}\,(\neg\mathrm{R})}{\Gamma, \neg\neg\varphi \vdash \varphi}\,(\neg\mathrm{L})}{\Gamma \vdash \varphi}\,(\text{modus ponens})$$

Insgesamt haben wir also z.B. die folgende in $\mathcal{SK}^+$ ableitbare Regel für die Beweisfigur des indirekten Beweises:

$$(\mathrm{Wid}) \quad \frac{\Gamma, \neg\varphi \vdash \psi \qquad \Gamma, \neg\varphi \vdash \neg\psi}{\Gamma \vdash \varphi}$$

Übung 6.9 Fügen Sie aus obigen Vorgaben eine Herleitung der Regel (Wid) auf der Basis von $\mathcal{SK}$ und modus ponens zusammen.

Wir können zu einem erweiterten Sequenzenkalkül $\mathcal{SK}^+$ also modus ponens, oder der Bequemlichkeit halber auch gleich noch (Kontr) und (Wid), hinzunehmen. Korrektheit bleibt erhalten (s.o.), Vollständigkeit natürlich ohnehin.

Weitere Regeln Man kann, um weitere natürliche Schlussfiguren direkt als Regeln oder als einfach ableitbare Regeln zur Verfügung zu haben, natürlich weitere korrekte Regeln einführen. Beispiele für ganz einfache, aber über die bisherigen hinausgehende Erweiterungen, liefern Abschwächnugsregeln (weakening) wie

$$\frac{\Gamma \vdash \Delta}{\Gamma, \varphi \vdash \Delta} \qquad \text{und} \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \varphi}$$

Übung 6.10 Zeigen Sie die Korrektheit folgender Doppelnegationsregeln

$$(\mathrm{NNL}) \quad \frac{\Gamma, \neg\neg\varphi \vdash \Delta}{\Gamma, \varphi \vdash \Delta} \qquad (\mathrm{NNR}) \quad \frac{\Gamma \vdash \Delta, \neg\neg\varphi}{\Gamma \vdash \Delta, \varphi}$$

$$(\mathrm{NNL}^-) \quad \frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \neg\neg\varphi \vdash \Delta} \qquad (\mathrm{NNR}^-) \quad \frac{\Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta, \neg\neg\varphi}$$

Welche dieser Regeln lassen sich durch direkte Simulation in $\mathcal{SK}$, welche in $\mathcal{SK}^+$ ($\mathcal{SK}$ mit modus ponens) simulieren, welche auch hier nicht?

Bemerkung/Ausblick: Auf dem Niveau bit-weiser Darstellung von Information kann man sich auf den Standpunkt stellen, dass "alles in AL ausdrückbar" sei (so wie jeder Computer als endlicher Automat beschreibbar ist). Für viele Zwecke ist dies aber das falsche Abstraktionsniveau. Über AL hinaus gibt es zunächst die Möglichkeit, mehrere AL-Belegungen an verschiedenen Stellen einer strukturierten Konfiguration in Beziehung zu setzen (vgl. AL-Variablen, die den Knoten von Graphen zugeordnet sind in einigen Beispielen oben). So kommt man z.B. zu Temporallogiken (zur Beschreibung zeitlicher Abläufe und Transformationen von AL-Belegungen) oder allgemeiner zu Modallogiken (z.B. zur Beschreibung von AL-Belegungen in den Zuständen von Transitionssystemen).

Index