Aufgabe 1

$$\emptyset \implies ((\neg A \to B) \to C) \to ((\neg B \to A) \to C)$$

$$(\neg A \to B) \to C \implies (\neg B \to A) \to C$$

$$\neg B \to A,\ (\neg A \to B) \to C \implies C$$

$$\neg B \to A \implies \neg A \to B,\ C \qquad\qquad C,\ \neg B \to A \implies C$$

$$\neg A,\ \neg B \to A \implies B,\ C$$

$$\neg A \implies \neg B,\ B,\ C \qquad\qquad A,\ \neg A \implies B,\ C$$

$$B,\ \neg A \implies B,\ C \qquad\qquad A \implies A,\ B,\ C$$

Aufgabe 2

Transformation der Negation von F in konjunktive Normalform

$$\begin{aligned}
\neg F &= \neg(((\neg A \to B) \to C) \to ((\neg B \to A) \to C)) \\
&\equiv \neg(\neg(\neg(A \lor B) \lor C) \lor (\neg(B \lor A) \lor C)) \\
&\equiv ((\neg A \land \neg B) \lor C) \land (B \lor A) \land \neg C \\
&\equiv (\neg A \lor C) \land (\neg B \lor C) \land (B \lor A) \land \neg C
\end{aligned}$$

S die Menge dieser Klauseln

Aufgabe 2 (Fortsetzung)

$$S = \{\neg A \lor C, \quad \neg B \lor C, \quad B \lor A, \quad \neg C\}$$

`widerlege`(S):
- wähle Einerklausel $\{\neg C\}$
- `reduziere`$(\{\neg C\}, S) \to$ Ergebnis: $S' = \{\neg A, \quad \neg B, \quad B \lor A\}$

`widerlege`(S')
- wähle Einerklausel $\{\neg A\}$
- `reduziere`$(\{\neg A\}, S') \to$ Ergebnis: $S'' = \{\neg B, \quad B\}$

`widerlege`(S'')
- wähle Einerklausel $\{\neg B\}$
- `reduziere`$(\{\neg B\}, S'') \to$ Ergebnis: $S'' = \{\Box\}$

`widerlege`$(\{\Box\})$:
- fertig, da leere Klausel enthalten.

Aufgabe 3

Klauseln in S:

$$\begin{array}{rcrcr@{\qquad\qquad}rcrcr}
P_1 & \lor & P_2 & \lor & P_3 & P_1 & \lor & P_2 & \lor & \neg P_3 \\
\neg P_1 & \lor & P_2 & \lor & P_3 & \neg P_1 & \lor & P_2 & \lor & \neg P_3 \\
P_1 & \lor & \neg P_2 & \lor & P_3 & P_1 & \lor & \neg P_2 & \lor & \neg P_3 \\
\neg P_1 & \lor & \neg P_2 & \lor & P_3 & \neg P_1 & \lor & \neg P_2 & \lor & \neg P_3
\end{array}$$

`widerlege`(S):
- keine Einerklausel $\to$ wähle Variable P_3
- `widerlege`$(S \cup \{P_3\})$
- `widerlege`$(S \cup \{\neg P_3\})$

Aufgabe 3 (Fortsetzung)

`widerlege`$(S \cup \{P_3\})$:
- wähle Einerklausel $\{P_3\}$
- `reduziere`$(\{P_3\}, S \cup \{P_3\}) \to$ Ergebnis: S'
- `widerlege`(S')

Klauseln in S':

$P_1 \vee \neg P_2$	$P_1 \vee P_2$
$\neg P_1 \vee \neg P_2$	$\neg P_1 \vee P_2$

`widerlege`$(S \cup \{\neg P_3\})$:
- wähle Einerklausel $\{\neg P_3\}$
- `reduziere`$(\{\neg P_3\}, S \cup \{\neg P_3\}) \to$ Ergebnis: S'
- `widerlege`(S')

Aufgabe 3 (Fortsetzung)

Klauseln in S':

$P_1 \vee \neg P_2$	$P_1 \vee P_2$
$\neg P_1 \vee \neg P_2$	$\neg P_1 \vee P_2$

`widerlege`(S'):
- keine Einerklausel $\to$ wähle Variable P_2
- `widerlege`$(S' \cup \{P_2\})$
- `widerlege`$(S' \cup \{\neg P_2\})$

`widerlege`$(S' \cup \{P_2\})$:
- wähle Einerklausel $\{P_2\}$
- `reduziere`$(\{P_2\}, S' \cup \{P_2\}) \to$ Ergebnis: $S'' = \{P_1, \neg P_1\}$
- `widerlege`(S'')

`widerlege`$(S' \cup \{\neg P_2\})$:
- wähle Einerklausel $\{\neg P_2\}$
- `reduziere`$(\{\neg P_2\}, S' \cup \{\neg P_2\}) \to$ Ergebnis: S''
- `widerlege`(S'')

Aufgabe 3 (Fortsetzung)

`widerlege`(S''):
- wähle Einerklausel $\{P_1\}$
- `reduziere`$(\{P_1\} \to$ Ergebnis: $\{\Box\}$
- `widerlege`$(\{\Box\})$

`widerlege`$(\{\Box\})$:
- fertig, da leere Klausel enthalten.

Aufgabe 4 a

Signatur: F: Flugreise V: Vollpension M: Meer P: Pool

Aussagen

Falls sie nicht mit dem Flugzeug fliegen,
besteht der Vater auf Vollpension am Meer.
$\neg F \to (V \wedge M)$

Die Mutter möchte mindestens einen ihrer drei Wünsche erfüllt sehen:
ans Meer fliegen, oder am Meer aber ohne Pool, oder Vollpension und Pool.
$(M \wedge F) \vee (M \wedge \neg P) \vee (V \wedge P)$

Gibt es keinen Pool, so besteht Tochter Lisa auf einer Flugreise und Urlaub am Meer und darauf daß keine Vollpension gebucht wird.
$\neg P \to (F \wedge M \wedge \neg V)$

Auch dem Baby soll einer seiner Wünsche erfüllt werden: erstens einen Pool und nicht fliegen oder zweitens Vollpension, dann aber ohne Pool.
$(P \wedge \neg F) \vee (V \wedge \neg P)$

Aufgabe 4 b

$\neg F \to (V \wedge M)$	(1)	$F \vee V$
	(2)	$F \vee M$
$(M \wedge F) \vee (M \wedge \neg P) \vee (V \wedge P)$	(3)	$M \vee V$
	(4)	$M \vee P$
	(5)	$M \vee \neg P \vee V$
	(6)	$F \vee M \vee V$
	(7)	$F \vee M \vee P$
	(8)	$F \vee \neg P \vee V$
$\neg P \to (F \wedge M \wedge \neg V)$	(9)	$P \vee F$
	(10)	$P \vee M$
	(11)	$P \vee \neg V$
$(P \wedge \neg F) \vee (V \wedge \neg P)$	(12)	$P \vee V$
	(13)	$\neg F \vee V$
	(14)	$\neg F \vee \neg P$

Aufgabe 4 c

Behauptung: $M \wedge V \wedge P \wedge \neg F$

Negation der Behauptung (Klausel): $\neg M \vee \neg V \vee \neg P \vee F$

Klauseltableau:

- Aufspaltung einer Klausel in einem Schritt
- Keine Vorzeichen

Heuristiken: Einschränkung des Suchraums bei der Konstruktion von Beweisen

- Ein Literal der Klausel, mit der erweitert wird, muß zum Blatt des Pfades komplementär sein (Konnektionsbedingung).
- Kein Literal darf auf einem Pfad zweimal vorkommen (Regularität).

Frage

Beobachtung:

Mit der Heuristik zur Beweissuche (Konnektionsbedingung und Regularität) kann die Beweiskonstruktion in Sackgassen laufen.

Frage:

Ist der Tableaukalkül für aussagenlogische Formeln in KNF mit diesen Einschränkungen dennoch vollständig?

Aufgabe 5

k:	Karl	$v(x)$:	Vater von x	$verw(x,y)$:	x, y verwandt
l:	Lisa	$m(x)$:	Mutter von x	$verh(x,y)$:	x, y verheiratet

(a) Karl ist ein Großvater.
$\exists x(k \doteq v(v(x)) \vee k \doteq v(m(x)))$

(b) Karl und Lisa sind verheiratet.
$verh(k, l)$

(c) Niemandes Großmutter ist irgend jemandes Vater.
$\neg\exists x \exists y(m(m(x)) \doteq v(y) \vee m(v(x)) \doteq v(y))$

Aufgabe 5 (Fortsetzung)

k:	Karl	$v(x)$:	Vater von x	$verw(x,y)$:	x, y verwandt
l:	Lisa	$m(x)$:	Mutter von x	$verh(x,y)$:	x, y verheiratet

(d) Cousins und Cousinen sind verwandt.

$$\forall x \forall y (\exists z ((z \doteq m(m(x)) \vee z \doteq m(v(x)) \vee z \doteq v(m(x)) \vee z \doteq v(v(x))) \wedge \\ (z \doteq m(m(y)) \vee z \doteq m(v(y)) \vee z \doteq v(m(y)) \vee z \doteq v(v(y))) \\ ) \to verw(x,y))$$

(e) Verwandte zweiten Grades sind nicht verheiratet
(nach § 1589 BGB ist der Grad der Verwandschaft
gleich der Zahl der die Verwandschaft vermittelnden Geburten).

$$\forall x \forall y (\exists z (\neg (x \doteq y) \wedge (z \doteq m(x) \vee z \doteq v(x)) \wedge \\ (y \doteq m(z) \vee y \doteq v(z) \vee z \doteq m(y) \vee z \doteq v(y)) \\ ) \to \neg verh(x,y))$$

Aufgabe 6

$p(x,y)$: Knoten x, y sind durch Kante verbunden.

(a) Der Graph ist symmetrisch.

$$\forall x \forall y (p(x,y) \to p(y,x))$$

(b) Der Graph enthält keinen Zyklus der Länge drei.

$$\neg \exists x \exists y \exists z (p(x,y) \wedge p(y,z) \wedge p(z,x))$$

(c) Der Graph enthält einen 3-Clique
(ein n-Clique ist eine Menge von n verschiedenen Knoten, so daß,
wenn k_1 und k_2 zwei verschiedene dieser Knoten sind,
es eine Kante von k_1 nach k_2 gibt).

$$\exists x \exists y \exists z (\neg x \doteq y \wedge \neg x \doteq z \wedge \neg y \doteq z \wedge \\ p(x,y) \wedge p(y,x) \wedge p(x,z) \wedge p(z,x) \wedge p(y,z) \wedge p(z,y))$$