3. Logik (Teil I) – Aussagenlogik)

3.1 Wozu "Logik" im Informatik-Studium?

Logik (altgriechisch: "logos": "Vernunft")

- "die Lehre des vernünftigen Schlussfolgerns"
- Teilgebiet der – Philosophie
  – Mathematik
  – Informatik

- zentrale Frage: "Wie kann man Aussagen miteinander verknüpfen, und auf welche Weise kann man formal Schlüsse ziehen und Beweise durchführen?"

- These: Logik spielt für die Informatik eine ähnlich wichtige Rolle wie die Differentialrechnung für die Physik

- Anwendungsbereiche der Logik in der Informatik: z.B.
  - zur Repräsentation von statischem Wissen (z.B. im Bereich der künstlichen Intelligenz)
  - zur Verifikation von
    * Schaltkreisen (Ziel: beweise, dass ein Schaltkreis bzw Chip "richtig" funktioniert)
    * Programmen (Ziel: beweise, dass ein Programm gewisse wünschenswerte Eigenschaften hat)

* Protokollen (Ziel: beweise, dass die Kommunikation zwischen 2 "Agenten", die nach einem gewissen "Protokoll" abläuft, "sicher" ist – etwa gegen Abhören
Anwendungsbeispiel: Internet-Banking)

- als Grundlage für Datenbank-Anfragesprachen
- als Bestandteil von Programmiersprachen (z.B. um "Bedingungen" in "IF-Anweisungen" zu formulieren)
- zur automatischen Generierung von Beweisen (so genannte "Theorembeweiser")

3.2 Aussagenlogik

Aussagen (im Sinne der Aussagenlogik) sind sprachliche Gebilde, die entweder wahr oder falsch sind.

Aussagen können mit Junktoren wie "nicht", "und", "oder", "wenn... dann" etc. zu komplexeren Aussagen verknüpft werden.

Die Aussagenlogik beschäftigt sich mit allgemeinen Prinzipien des korrekten Argumentierens und Schließens mit Aussagen und Kombinationen von Aussagen.

Beispiel 3.1: "Geburtstagsfeier"

Fred möchte mit möglichst vielen seiner Freunde Anne, Bernd, Christine, Dirk und Eva seinen Geburtstag feiern. Er weiß, dass Eva nur dann kommt, wenn Christine und Dirk kommen. Andererseits kommt Christine nur dann, wenn auch Anne kommt; und Dirk wird auf keinen Fall kommen, wenn Bernd und Eva beide zur Feier kommen. Anne wiederum wird nur dann kommen, wenn auch Bernd oder Christine dabei sind. Wenn allerdings Bernd und Anne beide zur Party kommen, dann wird Eva auf keinen Fall dabeisein.

Frage: Wie viele Freunde (und welche) werden im besten Fall zur Party kommen?

Das Wissen, das im obigen Text wiedergegeben ist, lässt sich in "Atomare Aussagen" zerlegen, die mit Junktoren verknüpft werden können.

Die "Atomaren Aussagen", um die sich der Text dreht, kürzen wir folgendermaßen ab:

A ≙ Anne kommt zur Feier
B ≙ Bernd " " "
C ≙ Christine " " "
D ≙ Dirk " " "
E ≙ Eva " " "

Das im Text zusammengefasste "Wissen" lässt sich wie folgt repräsentieren:

(Wenn E, dann (C und D))	Eva kommt nur dann, wenn Christine und Dirk kommen
und (Wenn C, dann A)	Christine kommt nur dann, wenn auch Anne kommt
und (Wenn (B und E), dann nicht D)	Dirk wird auf keinen Fall kommen, wenn Bernd und Eva beide kommen.
und (Wenn A, dann (B oder C))	Anne kommt nur dann, wenn auch Bernd oder Christine dabei sind
und (Wenn (B und A), dann nicht E)	Wenn Bernd und Anne beide kommen, dann wird Eva auf keinen Fall dabei sein.

Die Aussagenlogik liefert einen Formalismus, mit dessen Hilfe man solches "Wissen" modellieren und Schlüsse daraus ziehen kann — insbesondere z.B. um die Frage, mit wie vielen (und welchen) Gästen Fred bei seiner Feier rechnen kann, zu beantworten.

□ Ende Bsp 3.1.

Syntax und Semantik der Aussagenlogik

Syntax: Legt fest, welche Zeichenketten (Worte) Formeln der Aussagenlogik sind

Semantik: Legt fest, welche "Bedeutung" einzelne Formeln haben.

(vgl. "Syntax" und "Semantik" von JAVA-Programmen: die Syntax legt fest, welche Zeichenketten JAVA-Programme sind; die Semantik bestimmt, was das Programm tut)

Definition 3.2 (Aussagenvariablen und Alphabet der Aussagenlogik)

(a) Eine Aussagenvariable (kurz: Variable) hat die Form V_i, für $i \in \mathbb{N}$.

Die Menge aller Variablen bezeichnen wir mit AVAR, d.h. $AVAR := \{ V_i : i \in \mathbb{N} \}$.

(b) Das Alphabet der Aussagenlogik ist

$$A_{AL} := AVAR \cup \{0, 1, \neg, \wedge, \vee, \rightarrow, \leftrightarrow, (,)\}$$

Definition 3.3 (aussagenlogische Formeln: Syntax)

Die Menge AL der aussagenlogischen Formeln (kurz: Formeln) ist die folgendermaßen rekursiv definierte Teilmenge von A_{AL}^*:

Basisregeln:
- $0 \in AL$ (B0)
- $1 \in AL$ (B1)
- für jede Variable $X \in AVAR$ gilt: $X \in AL$ (BV)

Rekursive Regeln:
- Ist $\varphi \in AL$, so ist auch $\neg\varphi \in AL$ (R1)
- Ist $\varphi \in AL$ und $\psi \in AL$, so ist auch (R2)
  - $(\varphi \wedge \psi) \in AL$,
  - $(\varphi \vee \psi) \in AL$,
  - $(\varphi \to \psi) \in AL$,
  - $(\varphi \leftrightarrow \psi) \in AL$

Notation 3.4

(a) 0, 1 und die Variablen (d.h. die Elemente aus AVAR) bezeichnen wir als atomare Formeln bzw. Atome.

(b) Die Symbole $\neg, \wedge, \vee, \to, \leftrightarrow$ heißen Junktoren

(c) Sind φ und ψ Formeln (d.h. $\varphi \in AL$ und $\psi \in AL$), so heißt

- $(\varphi \wedge \psi)$ Konjunktion (bzw. ver-UND-ung) von φ und ψ
- $(\varphi \vee \psi)$ Disjunktion (bzw ver-ODER-ung) von φ und ψ
- $\neg\varphi$ Negation (bzw ver-NEIN-ung) von φ

Beispiel 3.5

Beispiele für Formeln (d.h. Worte über A_{AL}, die zur Menge AL gehören):

- $(\neg V_0 \vee (V_5 \rightarrow V_1))$
- $\neg((V_0 \wedge 0) \leftrightarrow \neg V_3)$

Aber beispielsweise

$$V_1 \vee V_2 \wedge V_3$$

ist keine Formel (d.h. kein Element in AL) (da die Klammern fehlen). Auch $(\neg V_1)$ ist keine Formel (da die Klammern "zu viel sind").

□ Ende Bsp 3.5

Wir wissen nun, welche Zeichenketten (über dem Alphabet A_{AL}) Formeln genannt werden.
Um festlegen zu können, welche Bedeutung (d.h. Semantik) solche Formeln haben, brauchen wir folgende Definitionen:

Definition 3.6

Die Variablenmenge einer aussagenlogischen Formel φ (kurz: $Var(\varphi)$) ist die Menge aller Variablen $X \in AVAR$, die in φ vorkommen.

Beispiel:

- $Var\big((\neg V_0 \vee (V_5 \to V_1))\big) = \{V_0, V_1, V_5\}$
- $Var\big(\neg((V_0 \wedge 0) \leftrightarrow \neg V_3)\big) = \{V_0, V_3\}$
- $Var\big((0 \vee 1)\big) = \emptyset$

Definition 3.7

(a) Eine Belegung (bzw. Wahrheitsbelegung) ist eine partielle Funktion von AVAR nach $\{0,1\}$.

(b) Eine Belegung $\mathcal{B}$ ist eine Belegung für eine Formel φ (bzw passend zu φ), wenn $Def(\mathcal{B}) \supseteq Var(\varphi)$.

Intuitive Bedeutung: 1 steht für "wahr", 0 steht für "falsch".

Definition 3.8 (Semantik der Aussagenlogik)

Rekursiv über den Aufbau von AL definieren wir eine Funktion $[\![\cdot]\!]^{\cdot}$, die jeder Formel $\varphi \in AL$ und jeder Belegung $\mathcal{B}$ für φ einen Wahrheitswert (kurz: Wert) $[\![\varphi]\!]^{\mathcal{B}} \in \{0,1\}$ zuordnet:

Rekursionsanfang:
- $[\![0]\!]^{\mathcal{B}} := 0$
- $[\![1]\!]^{\mathcal{B}} := 1$
- f.a. $X \in AVAR$ gilt: $[\![X]\!]^{\mathcal{B}} := \mathcal{B}(X)$

Rekursionsschritt:
- Ist $\varphi \in AL$, so ist $[\![\neg\varphi]\!]^{\mathcal{B}} := \begin{cases} 1 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 0 \\ 0 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 1 \end{cases}$
- Ist $\varphi \in AL$ und $\psi \in AL$, so ist
  - $[\![(\varphi \wedge \psi)]\!]^{\mathcal{B}} := \begin{cases} 1 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 1 \text{ und } [\![\psi]\!]^{\mathcal{B}} = 1 \\ 0 & \text{sonst} \end{cases}$
  - $[\![(\varphi \vee \psi)]\!]^{\mathcal{B}} := \begin{cases} 1 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 1 \text{ oder } [\![\psi]\!]^{\mathcal{B}} = 1 \\ 0 & \text{sonst} \end{cases}$
  - $[\![(\varphi \to \psi)]\!]^{\mathcal{B}} := \begin{cases} 1 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 0 \text{ oder } [\![\psi]\!]^{\mathcal{B}} = 1 \\ 0 & \text{sonst} \end{cases}$
  - $[\![(\varphi \leftrightarrow \psi)]\!]^{\mathcal{B}} := \begin{cases} 1 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}} \\ 0 & \text{sonst} \end{cases}$

Intuitive Bedeutung der Semantik:

- **Atome**: $\mathbf{1}$ und $\mathbf{0}$ bedeuten einfach "wahr" und "falsch".

  Die Variablen $X \in AVAR$ stehen für irgendwelche Aussagen. Uns interessiert hier aber nur, ob diese Aussagen "wahr" oder "falsch" sind — und dies wird durch eine Belegung $\mathcal{B}$ angegeben.

- **Negation**: $\neg\varphi$ bedeutet "nicht φ".

  D.h.: $\neg\varphi$ ist wahr (unter Belegung $\mathcal{B}$) $\Leftrightarrow$ φ ist falsch (unter Belegung $\mathcal{B}$)

  Darstellung durch eine so genannte Verknüpfungstafel (bzw. Wahrheitstafel):

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\neg\varphi]\!]^{\mathcal{B}}$
0	1
1	0

- **Konjunktion**: $(\varphi \wedge \psi)$ bedeutet "φ und ψ".

  D.h.: $(\varphi \wedge \psi)$ ist wahr (unter Belegung $\mathcal{B}$) $\Leftrightarrow$ φ ist wahr und ψ ist wahr (unter Belegung $\mathcal{B}$)

  Zugehörige Verknüpfungstafel:

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\psi]\!]^{\mathcal{B}}$	$[\![(\varphi \wedge \psi)]\!]^{\mathcal{B}}$
0	0	0
0	1	0
1	0	0
1	1	1

- **Disjunktion**: $(\varphi \vee \psi)$ bedeutet "φ oder ψ".

D.h. $(\varphi \vee \psi)$ ist wahr (unter Belegung $\mathcal{B}$) $\iff$ φ ist wahr oder ψ ist wahr (unter Belegung $\mathcal{B}$)

Zugehörige Verknüpfungstafel:

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\psi]\!]^{\mathcal{B}}$	$[\![(\varphi \vee \psi)]\!]^{\mathcal{B}}$
0	0	0
0	1	1
1	0	1
1	1	1

- **Implikation**: $(\varphi \to \psi)$ bedeutet "φ impliziert ψ", d.h. "wenn φ, dann auch ψ".

Dh: $(\varphi \to \psi)$ ist wahr (unter Belegung $\mathcal{B}$) $\iff$ Wenn φ wahr ist, dann ist auch ψ wahr (unter Belegung $\mathcal{B}$)

Zugehörige Verknüpfungstafel:

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\psi]\!]^{\mathcal{B}}$	$[\![(\varphi \to \psi)]\!]^{\mathcal{B}}$
0	0	1
0	1	1
1	0	0
1	1	1

- **Biimplikation**: $(\varphi \leftrightarrow \psi)$ bedeutet "φ genau dann, wenn ψ"

D.h.: $(\varphi \leftrightarrow \psi)$ ist wahr (unter Belegung $\mathcal{B}$) $\iff$ φ ist genau dann wahr, wenn ψ wahr ist (unter Belegung $\mathcal{B}$)

Zugehörige Verknüpfungstafel:

$[\![\varphi]\!]^{\mathcal{B}}$	$[\![\psi]\!]^{\mathcal{B}}$	$[\![(\varphi \leftrightarrow \psi)]\!]^{\mathcal{B}}$
0	0	1
0	1	0
1	0	0
1	1	1

Beispiel 3.9

Betrachte die Formel $\varphi := \left(\neg V_0 \vee (V_5 \to V_1)\right)$.

Dann ist beispielsweise die Funktion $\mathcal{B} : \{V_0, V_1, V_5\} \to \{0,1\}$ mit $\mathcal{B}(V_0) := 1$, $\mathcal{B}(V_1) := 1$ und $\mathcal{B}(V_5) := 0$ eine Belegung für φ.

Der Wahrheitswert von φ unter Belegung $\mathcal{B}$ ist der Wert

$$[\![\varphi]\!]^{\mathcal{B}} \underset{\text{Def 3.8}}{=} \begin{cases} 1 & \text{falls } [\![\neg V_0]\!]^{\mathcal{B}} = 1 \text{ oder } [\![(V_5 \to V_1)]\!]^{\mathcal{B}} = 1 \\ 0 & \text{sonst} \end{cases}$$

$$\underset{\text{Def 3.8}}{=} \begin{cases} 1 & \text{falls } [\![V_0]\!]^{\mathcal{B}} = 0 \text{ oder } \left([\![V_5]\!]^{\mathcal{B}} = 0 \text{ oder } [\![V_1]\!]^{\mathcal{B}} = 1\right) \\ 0 & \text{sonst} \end{cases}$$

$$\underset{\text{Def 3.8}}{=} \begin{cases} 1 & \text{falls } \mathcal{B}(V_0) = 0 \text{ oder } \mathcal{B}(V_5) = 0 \text{ oder } \mathcal{B}(V_1) = 1 \\ 0 & \text{sonst} \end{cases}$$

$$= 1 \qquad \text{(denn gemäß obiger Wahl von } \mathcal{B} \text{ gilt } \mathcal{B}(V_5) = 0)$$

Beobachtung: Sind $\mathcal{B}$ und $\mathcal{B}'$ zwei Belegungen für eine Formel φ, die auf $\mathrm{Var}(\varphi)$ übereinstimmen (d.h. f.a. $X \in \mathrm{Var}(\varphi)$ gilt $\mathcal{B}(X) = \mathcal{B}'(X)$), so ist $[\![\varphi]\!]^{\mathcal{B}} = [\![\varphi]\!]^{\mathcal{B}'}$.

In der Literatur wird diese Beobachtung oft unter dem Namen "Koinzidenzlemma" geführt.

Intuitiv ist die Beobachtung "offensichtlich richtig", denn in der Definition von $[\![\varphi]\!]^{\mathcal{B}}$ werden ja nur diejenigen Variablen verwendet, die in φ vorkommen (also zu $Var(\varphi)$ gehören). Einen formalen Beweis der Beobachtung kann man leicht per Induktion über den Aufbau von AL führen.

Aufgrund der Beobachtung des "Koinzidenzlemmas" werden wir im Folgenden, wenn wir Belegungen $\mathcal{B}$ für eine Formel φ betrachten, uns meistens nur diejenigen Werte $\mathcal{B}(X)$ interessieren, für die $X \in Var(\varphi)$ ist.

Anmerkung: (griechische Buchstaben)
In der Literatur werden Formeln einer Logik traditionell meistens mit griechischen Buchstaben bezeichnet.
Hier eine Liste der gebräuchlichsten Buchstaben:

Buchstabe	φ	ψ	χ	θ bzw ϑ	λ	μ	ν	τ	κ	σ	ϱ	ξ	ζ
Aussprache	phi	psi	chi	theta	lambda	mü	nü	tau	kappa	Sigma	rho	xi	zeta

Buchstabe	α	β	γ	δ	ω	ε	ι	π	Δ	Γ	Σ	Π	Φ
Aussprache	alpha	beta	gamma	delta	omega	epsilon	iota	pi	Delta	Gamma	Sigma	Pi	Phi

Um umgangssprachlich formuliertes Wissen (vgl. Beispiel 3.1 "Geburtstagsfeier") durch aussagenlogische Formeln zu repräsentieren, sind folgende Konventionen bequem:

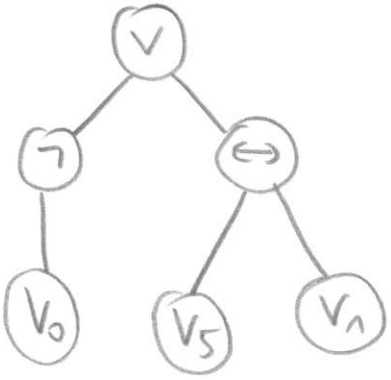
Notation 3.10:

- Statt $V_0, V_1, V_2, \ldots$ bezeichnen wir Variablen oft auch mit $A, B, C, \ldots, X, Y, Z, \ldots$ oder mit Varianten wie $X', Y_1, \ldots$.
- Wir schreiben $\bigwedge_{i=1}^{n} \varphi_i$ bzw. $(\varphi_1 \wedge \cdots \wedge \varphi_n)$ an Stelle von $\big(\cdots((\varphi_1 \wedge \varphi_2) \wedge \varphi_3) \wedge \cdots \wedge \varphi_n\big)$ (analog für "$\vee$" an Stelle von "$\wedge$").
- Die äußeren Klammern einer Formel lassen wir manchmal weg und schreiben z.B. $(A \wedge B) \to C$ an Stelle des (formal korrekten) $\big((A \wedge B) \to C\big)$.
- Ist φ eine Formel und $\mathcal{B}$ eine Belegung für φ, so sagen wir "$\mathcal{B}$ erfüllt φ" (bzw. "$\mathcal{B}$ ist eine erfüllende Belegung für φ"), falls $[\![\varphi]\!]^{\mathcal{B}} = 1$.

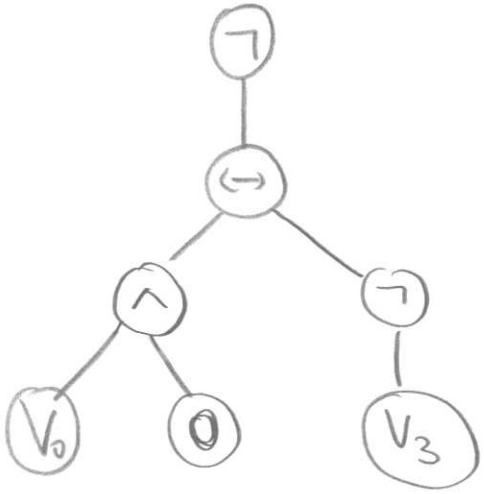
Bemerkung (Syntaxbäume zur graphischen Darstellung von Formeln)

Die Struktur einer Formel lässt sich bequem durch einen Syntaxbaum (englisch: parse tree) darstellen. Beispiele:

- Syntaxbaum der Formel $(\neg V_0 \vee (V_5 \leftrightarrow V_1))$:



- Syntaxbaum der Formel $\neg((V_0 \wedge 0) \leftrightarrow \neg V_3)$:



Umgangssprachliche Aussagen lassen sich wie folgt durch aussagenlogische Formeln repräsentieren:

Beispiel 3.11:

"Das Fluchtauto war rot oder grün und hatte weder vorne noch hinten ein Nummernschild"

Atomare Aussagen:

- X_R : das Fluchtauto war rot
- X_G : das Fluchtauto war grün
- X_V : das Fluchtauto hatte vorne ein Nummernschild
- X_H : das Fluchtauto hatte hinten ein Nummernschild

Obige Aussage wird dann durch folgende aussagenlogische Formel repräsentiert:

$$\big((X_R \vee X_G) \wedge (\neg X_V \wedge \neg X_H)\big).$$

Beispiel 3.12:

Das in Beispiel 3.1 ("Geburtstagsfeier") aufgelistete Wissen kann folgendermaßen repräsentiert werden:

Atomare Aussagen:

- A : Anne kommt zur Feier
- B : Bernd " " "
- C : Christine " " "
- D : Dirk " " "
- E : Eva " " "

Die Aussage des gesamten Textes in Bsp 3.1 wird durch folgende Formel repräsentiert:

$$\begin{aligned}\varphi := \quad & \big(E \to (C \wedge D)\big) \\ & \wedge \big(C \to A\big) \\ & \wedge \big((B \wedge E) \to \neg D\big) \\ & \wedge \big(A \to (B \vee C)\big) \\ & \wedge \big((B \wedge A) \to \neg E\big)\end{aligned}$$

Die Frage "Wie viele (und welche) Freunde werden im besten Fall zur Party kommen?" kann dann durch Lösen der folgenden Aufgabe beantwortet werden:

Finde eine Belegung $\mathcal{B}$ für φ, so dass

(1) φ von $\mathcal{B}$ erfüllt wird, d.h. $[\![\varphi]\!]^{\mathcal{B}} = 1$

und

(2) $|\{X \in \{A, B, C, D, E\} : \mathcal{B}(X) = 1\}|$ so groß wie möglich ist.

Um Aufgaben solcher Art lösen zu können, brauchen wir also eine Methode zum Finden der erfüllenden Belegungen für eine Formel.

Eine Möglichkeit dafür ist, so genannte Wahrheitstafeln zu benutzen.

Wahrheitstafeln:

Für jede Formel φ kann man die Werte unter allen möglichen Belegungen in einer Wahrheitstafel darstellen. Für jede Belegung $\mathcal{B}: Var(\varphi) \to \{0,1\}$ hat die Wahrheitstafel eine Zeile, die die Werte $\mathcal{B}(X)$, f.a. $X \in Var(\varphi)$, und den Wert $I\varphi I^{\mathcal{B}}$ enthält.
Um die Wahrheitstafel für φ auszufüllen, ist es bequem, auch Spalten für (alle oder einige) "Teilformeln" von φ einzufügen

Beispiel: Wahrheitstafel für $\varphi := (\neg V_0 \vee (V_5 \to V_1))$

V_0	V_1	V_5	$\neg V_0$	$(V_5 \to V_1)$	φ
0	0	0	1	1	1
0	0	1	1	0	1
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	0	1	1
1	0	1	0	0	0
1	1	0	0	1	1
1	1	1	0	1	1

Beispiel: Wahrheitstafel für $\varphi := (X \wedge ((1 \to 0) \to 0))$

X	1	0	$(1 \to 0)$	$((1 \to 0) \to 0)$	φ
0	1	0	0	1	0
1	1	0	0	1	1

Die erfüllenden Belegungen für eine Formel φ entsprechen also gerade denjenigen Zeilen der Wahrheitstafel für φ, in denen in der mit "φ" beschrifteten Spalte der Wert 1 steht.

Dies liefert uns ein Werkzeug, um die in Bsp. 3.12 beschriebene Aufgabe zur "Geburtstagsfeier" zu lösen

Beispiel 3.13:

Sei φ die Formel aus Beispiel 3.12.

Die Frage "Wie viele (und welche) Freunde werden bestenfalls zur Party kommen?" können wir lösen, indem wir

1) die Wahrheitstafel für φ ermitteln
2) alle Zeilen rassuchen, in denen in der mit "φ" beschrifteten Spalte der Wert 1 steht
3) aus diesen Zeilen all jene rassuchen, bei denen in den mit A, B, C, D, E beschrifteten Spalten möglichst viele Einsen stehen — jede dieser Zeilen repräsentiert dann eine größtmögliche Konstellation von gleichzeitigen Partybesuchern.

Prinzipiell führt diese Vorgehensweise zum Ziel – leider ist das Verfahren aber recht aufwendig, da die Wahrheitstafel, die man dabei aufstellen muss, sehr groß wird:

	A	B	C	D	E	$E \to (C \wedge D)$	$C \to A$	$(B \wedge E) \to \neg D$	$A \to (B \vee C)$	$(B \wedge A) \to \neg E$	φ	
→	0	0	0	0	0	1	1	1	1	1	1	←
	0	0	0	0	1	0	1	1	1	1	0	
→	0	0	0	1	0	1	1	1	1	1	1	←
	0	0	0	1	1	0	1	1	1	1	0	
	0	0	1	0	0	1	0	1	1	1	0	
	0	0	1	0	1	0	0	1	1	1	0	
	0	0	1	1	0	1	0	1	1	1	0	
	0	0	1	1	1	1	0	1	1	1	0	
→	0	1	0	0	0	1	1	1	1	1	1	←
	0	1	0	0	1	0	1	1	1	1	0	
→	0	1	0	1	0	1	1	1	1	1	1	←
	0	1	0	1	1	0	1	0	1	1	0	
	0	1	1	0	0	1	0	1	1	1	0	
	0	1	1	0	1	0	0	1	1	1	0	
	0	1	1	1	0	1	0	1	1	1	0	
	0	1	1	1	1	1	0	0	1	1	0	
	1	0	0	0	0	1	1	1	0	1	0	
	1	0	0	0	1	0	1	1	0	1	0	
	1	0	0	1	0	1	1	1	0	1	0	
	1	0	0	1	1	0	1	1	0	1	0	
→	1	0	1	0	0	1	1	1	1	1	1	←
	1	0	1	0	1	0	1	1	1	1	0	
→	1	0	1	1	0	1	1	1	1	1	1	←
→	1	0	1	1	1	1	1	1	1	1	1	←
→	1	1	0	0	0	1	1	1	1	1	1	←
	1	1	0	0	1	0	1	1	1	0	0	
→	1	1	0	1	0	1	1	1	1	1	1	←
	1	1	0	1	1	0	1	0	1	0	0	
→	1	1	1	0	0	1	1	1	1	1	1	←
	1	1	1	0	1	0	1	1	1	0	0	
→	1	1	1	1	0	1	1	1	1	1	1	←
	1	1	1	1	1	1	1	0	1	0	0	

Erfüllende Belegungen für φ werden werden hier durch Zeilen repräsentiert, die mit einem Pfeil "→" markiert sind.

In der Wahrheitstafel sieht man, dass es keine erfüllende Belegung gibt, bei der in den mit A bis E beschrifteten Spalten insgesamt 5 Einsen stehen, und dass es genau zwei erfüllende Belegungen gibt, bei denen in den mit A bis E beschrifteten Spalten insgesamt 4 Einsen stehen, nämlich die beiden Belegungen $\mathcal{B}_1$ und $\mathcal{B}_2$ mit

$$\mathcal{B}_1(A) = \mathcal{B}_1(C) = \mathcal{B}_1(D) = \mathcal{B}_1(E) = 1 \text{ und } \mathcal{B}_1(B) = 0$$

und

$$\mathcal{B}_2(A) = \mathcal{B}_2(B) = \mathcal{B}_2(C) = \mathcal{B}_2(D) = 1 \text{ und } \mathcal{B}_2(E) = 0$$

Die Antwort auf die Frage "Wie viele (und welche) Freunde werden bestenfalls zur Party kommen?" lautet also:

Bestenfalls werden 4 der 5 Freunde kommen, und dafür gibt es zwei Möglichkeiten, nämlich

1. dass alle außer Bernd kommen, und
2. dass alle außer Eva kommen.

□ Ende Bsp 3.13

Angesichts der Wahrheitstafel aus Bsp 3.13 stellt sich die Frage, wie groß die Wahrheitstafel für eine gegebene Formel φ ist. Die Antwort darauf gibt der folgende Satz.

Satz 3.14

Sei φ eine aussagenlogische Formel und sei $n := |Var(\varphi)|$ die Anzahl der in φ vorkommenden Variablen. Dann gibt es 2^n verschiedene zu φ passende Belegungen $\mathcal{B}$ mit $Def(\mathcal{B}) = Var(\varphi)$

Beweis:

$$\{\mathcal{B} : \mathcal{B} \text{ ist eine zu } \varphi \text{ passende Belegung mit } Def(\mathcal{B}) = Var(\varphi)\}$$

$$\underset{\text{Def 3.7}}{=} \{\mathcal{B} : \mathcal{B}: Var(\varphi) \to \{0,1\} \text{ ist eine Funktion}\}$$

$$\underset{\text{Notation 2.24}}{=} Abb(Var(\varphi), \{0,1\}).$$

Wir wissen, dass

$$|Abb(Var(\varphi), \{0,1\})| \underset{\substack{\text{Folgerung} \\ 2.32(a)}}{=} |\{0,1\}|^{|Var(\varphi)|} \underset{\substack{n = |Var(\varphi)| \\ \text{nach Voraussetzung}}}{=} 2^n.$$

□

Satz 3.14 besagt, dass die Wahrheitstafel einer Formel mit n Variablen genau 2^n Zeilen hat. Wie die folgende Tabelle zeigt, ergibt das bereits bei relativ kleinen Werten von n schon riesige Wahrheitstafeln:

n = Anzahl Variablen	2^n = Anzahl Zeilen der Wahrheitstafel
10	$2^{10} = 1.024 \approx 10^3$
20	$2^{20} = 1.048.576 \approx 10^6$
30	$2^{30} = 1.073.741.824 \approx 10^9$
40	$2^{40} = 1.099.511.627.776 \approx 10^{12}$
50	$2^{50} = 1.125.899.906.842.624 \approx 10^{15}$
60	$2^{60} = 1.152.921.504.606.846.976 \approx 10^{18}$

Zum Vergleich: Das Alter des Universums wird auf 10^{10} Jahre $< 10^{18}$ Sekunden geschätzt.

Ein ganzer Zweig der Informatik (z.B. unter dem Stichwort "SAT-Solving") und viele internationale Forschungsgruppen beschäftigen sich mit der Aufgabe, Verfahren zu entwickeln, die die erfüllenden Belegungen von aussagenlogischen Formeln ermitteln und dabei wesentlich effizienter sind als das vorgestellte Wahrheitstafel-Verfahren.

Ein relativ ernüchterndes Resultat, das Sie in der "Algorithmentheorie"-Vorlesung kennenlernen werden, ist der folgende Satz:

Satz: Das aussagenlogische Erfüllbarkeitsproblem ist NP-vollständig.

Das aussagenlogische Erfüllbarkeitsproblem (kurz: SAT, für englisch: "satisfiability") ist dabei das folgende Berechnungsproblem:

Eingabe: eine aussagenlogische Formel φ

Frage: Gibt es eine erfüllende Belegung für φ?

Was der Begriff "NP-vollständig" genau bedeutet, werden Sie in der "Algorithmentheorie"-Vorlesung lernen; grob gesagt bedeutet "NP-vollständig", dass es "wahrscheinlich keinen effizienten Algorithmus gibt, der das aussagenlogische Erfüllbarkeitsproblem löst".

Andererseits wurden (besonders in den letzten Jahren) Heuristiken und randomisierte Algorithmen entwickelt, die das aussagenlogische Erfüllbarkeitsproblem trotzdem in vielen Fällen erstaunlich effizient lösen können.

Die folgenden Begriffe werden Ihnen in späteren Vorlesungen immer wieder begegnen:

Definition 3.15

Sei φ eine aussagenlogische Formel.

(a) φ heißt erfüllbar, wenn es (mindestens) eine erfüllende Belegung für φ gibt, d.h. wenn es (mindestens) eine zu φ passende Belegung $\mathcal{B}$ gibt mit $[\![\varphi]\!]^{\mathcal{B}} = 1$

(b) φ heißt unerfüllbar, wenn es keine erfüllende Belegung für φ gibt.

(c) φ heißt allgemeingültig (bzw. Tautologie), wenn jede zu φ passende Belegung φ erfüllt, d.h. wenn für jede zu φ passende Belegung $\mathcal{B}$ gilt: $[\![\varphi]\!]^{\mathcal{B}} = 1$

Beispiel 3.16

(a) Die Formel $((X \vee Y) \wedge (\neg X \vee Y))$ ist

- erfüllbar, da z.B. die Belegung $\mathcal{B}$ mit $\mathcal{B}(X) = 0$ und $\mathcal{B}(Y) = 1$ die Formel erfüllt
- nicht allgemeingültig, da z.B. die Belegung $\mathcal{B}'$ mit $\mathcal{B}'(X) = 0$ und $\mathcal{B}'(Y) = 0$ die Formel nicht erfüllt

(b) Die Formel $(X \wedge \neg X)$ ist

unerfüllbar, da für jede zur Formel passende Belegung $\mathcal{B}$ entweder $\mathcal{B}(X) = 1$ oder $\mathcal{B}(X) = 0$ gilt

Fall 1: $\mathcal{B}(X) = 1$:

$$[\![(X \wedge \neg X)]\!]^{\mathcal{B}} = \begin{cases} 1, \text{ falls } \mathcal{B}(X) = 1 \text{ und } \mathcal{B}(X) = 0 \\ 0 \text{ sonst} \end{cases}$$

$$= 0, \text{ da } \mathcal{B}(X) = 1 \neq 0$$

Fall 2: $\mathcal{B}(X) = 0$:

$$[\![(X \wedge \neg X)]\!]^{\mathcal{B}} = \begin{cases} 1, \text{ falls } \mathcal{B}(X) = 1 \text{ und } \mathcal{B}(X) = 0 \\ 0 \text{ sonst} \end{cases}$$

$$= 0, \text{ da } \mathcal{B}(X) = 0 \neq 1.$$

(c) Die Formel $(X \vee \neg X)$ ist

allgemeingültig, da für jede zur Formel passende Belegung $\mathcal{B}$ entweder $\mathcal{B}(X) = 1$ oder $\mathcal{B}(X) = 0$ gilt. Somit gilt $[\![(X \vee \neg X)]\!]^{\mathcal{B}} = 1$, für alle zur Formel passenden Belegungen $\mathcal{B}$.

Beobachtung 3.17

Sei φ eine aussagenlogische Formel.

(a) φ ist erfüllbar $\iff$ in der Wahrheitstafel für φ steht in der mit "φ" beschrifteten Spalte mindestens eine 1.

(b) φ ist unerfüllbar $\iff$ in der Wahrheitstafel für φ stehen in der mit "φ" beschrifteten Spalte nur Nullen.

(c) φ ist allgemeingültig $\iff$ in der Wahrheitstafel für φ stehen in der mit "φ" beschrifteten Spalte nur Einsen.

Folgerung 3.18

Für alle aussagenlogischen Formeln φ gilt:

φ ist allgemeingültig $\iff$ $\neg\varphi$ ist unerfüllbar.

Folgerung und Äquivalenz

Definition 3.19 (semantische Folgerung)

Seien φ und ψ zwei aussagenlogische Formeln.
Wir sagen: ψ folgt aus φ (kurz: $\varphi \models \psi$, "φ impliziert ψ"), falls für jede zu φ und ψ passende Belegung $\mathcal{B}$ gilt:

Falls $[\![\varphi]\!]^{\mathcal{B}} = 1$, so auch $[\![\psi]\!]^{\mathcal{B}} = 1$.

D.h.: $\varphi \models \psi$:$\Leftrightarrow$ Jede Belegung, die zu φ und ψ passt und die φ erfüllt, erfüllt auch ψ.

Beispiel 3.20

Sei $\varphi := ((X \vee Y) \wedge (\neg X \vee Y))$ und $\psi := (Y \vee (\neg X \wedge \neg Y))$

Dann gilt $\varphi \models \psi$, aber nicht "$\psi \models \varphi$" (kurz: $\psi \not\models \varphi$),
denn:

X	Y	$(X \vee Y)$	$(\neg X \vee Y)$	φ	ψ
0	0	0	1	0	1
0	1	1	1	1	1
1	0	1	0	0	0
1	1	1	1	1	1

Hier steht in jeder Zeile (d.h. jede zu φ und ψ passende Belegung), in der in der mit "φ" beschrifteten Spalte eine 1 steht, auch in der mit "ψ" beschrifteten Spalte eine 1.
Somit gilt $\varphi \models \psi$.

Andererseits steht in Zeile 1 in der mit ψ beschrifteten Spalte eine 1 und in der mit φ beschrifteten Spalte eine 0. Für die entsprechende Belegung $\mathcal{B}$ (mit $\mathcal{B}(X)=0$ und $\mathcal{B}(Y)=0$) gilt also

$$[\![\psi]\!]^{\mathcal{B}} = 1 \quad \text{und} \quad [\![\varphi]\!]^{\mathcal{B}} = 0.$$

Daher gilt $\psi \not\models \varphi$.

Beobachtung 3.21:

Seien φ und ψ aussagenlogische Formeln. Dann gilt:

(a) $1 \models \varphi \iff \varphi$ ist allgemeingültig

(b) $\varphi \models 0 \iff \varphi$ ist unerfüllbar

(c) $\varphi \models \psi \iff (\varphi \to \psi)$ ist allgemeingültig

(d) $\varphi \models \psi \iff \varphi \wedge \neg\psi$ ist unerfüllbar

Beweis: Übung.

Definition 3.22 (logische Äquivalenz)

Zwei aussagenlogische Formeln φ und ψ heißen **äquivalent** (kurz: $\varphi \equiv \psi$), wenn für alle zu φ und ψ passenden Belegungen $\mathcal{B}$ gilt: $[\![\varphi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}}$.

Beispiel 3.23

Sei $\varphi := (X \wedge (X \vee Y))$ und $\psi := X$.

Dann ist $\varphi \equiv \psi$, denn

X	Y	$(X \vee Y)$	φ	ψ
0	0	0	0	0
0	1	1	0	0
1	0	1	1	1
1	1	1	1	1

Hier ist die mit "φ" beschriftete Spalte identisch zur mit "ψ" beschrifteten Spalte. D.h. für alle zu φ und ψ passenden Belegungen $\mathcal{B}$ gilt $[\![\varphi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}}$.
D.h.: $\varphi \equiv \psi$.

Beobachtung 3.24:

Seien φ und ψ aussagenlogische Formeln. Dann gilt:

(a) $\varphi \equiv \psi \iff (\varphi \leftrightarrow \psi)$ ist allgemeingültig
$\iff \varphi \models \psi$ und $\psi \models \varphi$.

(b) φ ist allgemeingültig $\Leftrightarrow$ $\varphi \equiv 1$

(c) φ ist erfüllbar $\Leftrightarrow$ $\varphi \not\equiv 0$
(d.h.: "$\varphi \equiv 0$" gilt nicht).

Beweis: Übung.

Fundamentale Äquivalenzen der Aussagenlogik

Satz 325:
Seien φ, ψ und χ aussagenlogische Formeln. Dann gilt:

(1) (Idempotenz):
- $(\varphi \wedge \varphi) \equiv \varphi$
- $(\varphi \vee \varphi) \equiv \varphi$

(2) (Kommutativität):
- $(\varphi \wedge \psi) \equiv (\psi \wedge \varphi)$
- $(\varphi \vee \psi) \equiv (\psi \vee \varphi)$

(3) (Assoziativität):
- $((\varphi \wedge \psi) \wedge \chi) \equiv (\varphi \wedge (\psi \wedge \chi))$
- $((\varphi \vee \psi) \vee \chi) \equiv (\varphi \vee (\psi \vee \chi))$

(4) (Absorption):
- $(\varphi \wedge (\varphi \vee \psi)) \equiv \varphi$
- $(\varphi \vee (\varphi \wedge \psi)) \equiv \varphi$

(5) (Distributivität): • $(\varphi \wedge (\psi \vee \chi)) \equiv ((\varphi \wedge \psi) \vee (\varphi \wedge \chi))$
• $(\varphi \vee (\psi \wedge \chi)) \equiv ((\varphi \vee \psi) \wedge (\varphi \vee \chi))$

(6) (doppelte Negation): $\neg\neg\varphi \equiv \varphi$

(7) (De Morgansche Regeln): • $\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$
• $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$

(8) (Tertium non Datur): • $(\varphi \wedge \neg\varphi) \equiv 0$
• $(\varphi \vee \neg\varphi) \equiv 1$

(9) • $(\varphi \wedge 1) \equiv \varphi$
• $(\varphi \wedge 0) \equiv 0$
• $(\varphi \vee 1) \equiv 1$
• $(\varphi \vee 0) \equiv \varphi$

(10) • $1 \equiv \neg 0$
• $0 \equiv \neg 1$

(11) (Elimination der Implikation): $(\varphi \to \psi) \equiv (\neg\varphi \vee \psi)$

(12) (Elimination der Biimplikation): $(\varphi \leftrightarrow \psi) \equiv ((\varphi \to \psi) \wedge (\psi \to \varphi))$

Beweis: Durch einfaches Nachrechnen. Details: Übung.

Bemerkung 3.26:

Durch schrittweises Anwenden der in Satz 3.25 aufgelisteten Äquivalenzen kann man eine gegebene aussagenlogische Formel in eine zu ihr äquivalente Formel umformen.

Beispiel: Sind φ und ψ aussagenlogische Formeln, so gilt:

$$(\varphi \leftrightarrow \psi) \equiv ((\varphi \to \psi) \wedge (\psi \to \varphi)) \quad \text{(Satz 3.25 (12))}$$

$$\equiv ((\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)) \quad \text{(Satz 3.25 (11))}$$

Normalformen

Bisher haben wir gesehen, wie man für eine gegebene aussagenlogische Formel φ eine Wahrheitstafel aufstellen kann.

Frage: Wie kann man umgekehrt zu einer gegebenen Wahrheitstafel eine Formel φ finden, zu der die Wahrheitstafel passt?

Beispiel 3.27: Betrachte die Wahrheitstafel $T :=$

X	Y	Z	φ
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

Eine Formel φ, so dass T die Wahrheitstafel für φ ist, kann man folgendermaßen erzeugen:

- Betrachte alle Zeilen von T, in denen in der mit "φ" beschrifteten Spalte eine 1 steht

- Für jede solche Zeile konstruiere eine Formel, die genau von der zur Zeile gehörenden Belegung erfüllt wird
- Bilde die Disjunktion (d.h. Ver-ODER-ung) über all diese Formeln.
  Dies liefert die gesuchte Formel φ.

In unserer Beispiel-Wahrheitstafel T gibt es genau 3 Zeilen, in denen in der mit φ beschrifteten Spalte eine 1 steht, nämlich die Zeilen

X	Y	Z	φ	zur Belegung der jeweiligen Zeile gehörende Formel:
0	0	0	1	$(\neg X \wedge \neg Y \wedge \neg Z)$
⋮	⋮	⋮	⋮	
1	0	0	1	$(X \wedge \neg Y \wedge \neg Z)$
1	0	1	1	$(X \wedge \neg Y \wedge Z)$
⋮	⋮	⋮	⋮	

$\Rightarrow$ zur Wahrheitstafel T passende Formel $\varphi :=$

$$(\neg X \wedge \neg Y \wedge \neg Z) \vee (X \wedge \neg Y \wedge \neg Z) \vee (X \wedge \neg Y \wedge Z).$$

Generell kann man auf die beschriebene Art zu jeder beliebigen Wahrheitstafel eine aussagenlogische Formel konstruieren, die zur Wahrheitstafel passt. Die so konstruierten Formeln haben eine besonders einfache Form: Sie sind Disjunktionen von Formeln, die aus Konjunktionen von Variablen oder negierten Variablen bestehen.

Formeln, die diese spezielle Struktur besitzen, nennt man auch Formeln in disjunktiver Normalform (kurz: DNF).

Definition 3.28:

(a) Ein Literal ist eine Formel der Form X oder $\neg X$, wobei $X \in AVAR$ (d.h. X ist eine Aussagenvariable).
Ein Literal der Form X, mit $X \in AVAR$, wird auch positives Literal genannt; eine Formel der Form $\neg X$, mit $X \in AVAR$, negatives Literal.

(b) Eine aussagenlogische Formel ist in disjunktiver Normalform (DNF), wenn sie eine Disjunktion von Konjunktionen von Literalen ist, d.h. wenn sie die Gestalt

$$\bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} \ell_{ij} \right)$$

hat, wobei $n, m_1, \dots, m_n \in \mathbb{N}_{>0}$ und ℓ_{ij} ein Literal ist (für jedes $i \in \{1, \dots, n\}$, $j \in \{1, \dots, m_i\}$).
Die Teilformeln $H_i := \bigwedge_{j=1}^{m_i} \ell_{ij}$ (für $i \in \{1, \dots, n\}$) heißen konjunktive Klauseln.

(c) Eine aussagenlogische Formel ist in konjunktiver Normalform (KNF), wenn sie eine Konjunktion von Disjunktionen von Literalen ist, d.h. wenn sie die Gestalt

$$\bigwedge_{i=1}^{n} \left(\bigvee_{j=1}^{m_i} \ell_{ij} \right)$$

hat, wobei $n, m_1, \ldots, m_n \in \mathbb{N}_{>0}$ und ℓ_{ij} ein Literal ist (für jedes $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, m_i\}$).

Die Teilformeln $\chi_i := \bigvee_{j=1}^{m_i} \ell_{ij}$ (für $i \in \{1, \ldots, n\}$) heißen disjunktive Klauseln.

Normalformen spielen in vielen Anwendungsgebieten eine wichtige Rolle. Beispielsweise geht man in der Schaltungstechnik (Hardware-Entwurf) oft von DNF-Formeln aus, während bei der aussagenlogischen Modellbildung oftmals KNF-Formeln auftreten, da sich eine Sammlung von einfach strukturierten Aussagen sehr gut durch eine Konjunktion von Klauseln ausdrücken lässt.

Satz 3.29:
Für jede aussagenlogische Formel φ gibt es eine Formel ψ_D in DNF und eine Formel ψ_K in KNF, so dass $\varphi \equiv \psi_D \equiv \psi_K$

(D.h.: Jede Formel ist äquivalent zu einer Formel in DNF und zu einer Formel in KNF.)

Beweisidee:
- Zur Konstruktion einer zu φ äquivalenten Formel ψ_D in DNF stellen wir zunächst die

Wahrheitstafel für φ auf. Falls diese in der mit "φ" beschrifteten Spalte nur Nullen hat (d.h. φ ist unerfüllbar), so setzen wir $\psi_D := (V_0 \wedge \neg V_0)$ — offensichtlich ist ψ_D in DNF und unerfüllbar, also äquivalent zu φ).

Falls die mit "φ" beschriftete Spalte der Wahrheitstafel mindestens eine 1 enthält, so gehen wir wie in Beispiel 3.27 vor, um eine zu φ äquivalente Formel ψ_D in DNF zu konstruieren.

- Zur Konstruktion einer zu φ äquivalenten Formel ψ_K in KNF können wir folgendermaßen vorgehen:

  1) Sei $\varphi' := \neg\varphi$

  2) Konstruiere eine zu φ' äquivalente Formel ψ'_D in DNF

     Sei $\bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{m_i} l_{ij} \right)$ die Gestalt von ψ'_D.

  3) Für alle i,j sei $\tilde{l}_{ij} := \begin{cases} \neg X & \text{falls } l_{ij} = X \text{ für ein } X \in AVAR \\ X & \text{falls } l_{ij} = \neg X \text{ für ein } X \in AVAR \end{cases}$

  4) Setze $\psi_K := \bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{m_i} \tilde{l}_{ij} \right)$.

Offensichtlich ist ψ_K eine Formel in KNF.

Außerdem gilt:

$$\varphi \equiv \neg\varphi'$$

$$\equiv \neg\psi_D'$$

$$\equiv \neg\left(\bigvee_{i=1}^{n}\left(\bigwedge_{j=1}^{m_i} l_{i,j}\right)\right)$$

$$\equiv \left(\bigwedge_{i=1}^{n}\neg\left(\bigwedge_{j=1}^{m_i} l_{i,j}\right)\right)$$

Satz 3.25 (7) "De Morgan"

$$\equiv \bigwedge_{i=1}^{n}\left(\bigvee_{j=1}^{m_i}\neg l_{i,j}\right)$$

Satz 3.25 (7) "De Morgan"

$$\equiv \bigwedge_{i=1}^{n}\left(\bigvee_{j=1}^{m_i}\tilde{l}_{i,j}\right)$$

Def. $\tilde{l}_{i,j}$

$$\underbrace{\bigwedge_{i=1}^{n}\left(\bigvee_{j=1}^{m_i}\tilde{l}_{i,j}\right)}$$

$$\stackrel{\text{Def}}{=} \psi_K$$

□

Abgesehen von DNF und KNF gibt es noch eine weitere wichtige Normalform, die so genannte Negationsnormalform:

Definition 3.30

Eine aussagenlogische Formel ist in Negationsnormalform, (NNF), wenn sie keins der Symbole $\rightarrow, \leftrightarrow, 0, 1$ enthält und Negationszeichen nur unmittelbar vor Variablen auftreten.

Rekursiv lässt sich die Menge der Formeln in NNF folgendermaßen definieren:

Basisregeln: Für jedes $X \in AVAR$ ist sowohl X als auch $\neg X$ eine Formel in NNF

Rekursive Regeln: Sind φ und ψ Formeln in NNF, so sind auch $(\varphi \wedge \psi)$ und $(\varphi \vee \psi)$ Formeln in NNF.

Beobachtung 3.31:

Jede Formel, die in KNF oder in DNF ist, ist auch in NNF. Aus Satz 3.29 folgt also insbesondere, dass jede aussagenlogische Formel äquivalent zu einer Formel in NNF ist.

Beachte: Nicht jede Formel in NNF ist auch in KNF oder in DNF,
Beispiel: $(((X \wedge \neg Y) \vee (\neg X \wedge Y)) \wedge \neg Z)$ ist in NNF, aber weder in KNF noch in DNF.

Ein einfaches Verfahren zur Transformation einer gegebenen aussagenlogischen Formel in eine äquivalente Formel in NNF beruht auf der wiederholten Anwendung der De Morganschen Regeln (Satz 3.25(7)) und der Regel für "doppelte Negation" (Satz 3.25(6)):

Mit den De Morganschen Regeln ($\neg(\varphi \wedge \psi) \equiv (\neg\varphi \vee \neg\psi)$ bzw. $\neg(\varphi \vee \psi) \equiv (\neg\varphi \wedge \neg\psi)$) ziehen wir das

Negationszeichen nach innen; mit der "doppelte Negation"-Regel $(\neg\neg\varphi \equiv \varphi)$ können wir Schritt für Schritt mehrfach hintereinander vorkommende Negationszeichen eliminieren. Eventuell in der Formel vorkommende Implikationspfeile "$\to$" oder Biimplikationspfeile "$\leftrightarrow$" eliminieren wir durch Verwenden von Satz 3.25 (11) und (12).

Eventuelle Vorkommen der Symbole **0** bzw **1** ersetzen wir durch die Formel $(V_0 \wedge \neg V_0)$ bzw. $(V_0 \vee \neg V_0)$.

Beispiel 3.32

Ziel: Bringe die Formel $\Big(\big(\neg V_0 \wedge \neg\big((V_0 \vee V_1) \to V_0\big)\big) \to 0 \Big)$ in NNF, d.h. finde eine zur gegebenen Formel äquivalente Formel in NNF

Lösung:

$$\Big(\big(\neg V_0 \wedge \neg\big((V_0 \vee V_1) \to V_0\big)\big) \to \underline{0} \Big)$$

$$\equiv \Big(\big(\neg V_0 \wedge \neg\big((V_0 \vee V_1) \to V_0\big)\big) \to (V_0 \wedge \neg V_0) \Big)$$

$$\equiv \Big(\neg\big(\neg V_0 \wedge \neg\big((V_0 \vee V_1) \to V_0\big)\big) \vee (V_0 \wedge \neg V_0) \Big)$$

$$\equiv \Big(\neg\big(\neg V_0 \wedge \neg\big(\neg(V_0 \vee V_1) \vee V_0\big)\big) \vee (V_0 \wedge \neg V_0) \Big)$$

$$\equiv \Big(\big(\neg\neg V_0 \vee \neg\neg\big(\neg(V_0 \vee V_1) \vee V_0\big)\big) \vee (V_0 \wedge \neg V_0) \Big)$$

$$\equiv \Big(\big(V_0 \vee \big(\neg(V_0 \vee V_1) \vee V_0\big)\big) \vee (V_0 \wedge \neg V_0) \Big)$$

$$\equiv \Big(\big(V_0 \vee \big((\neg V_0 \wedge \neg V_1) \vee V_0\big)\big) \vee (V_0 \wedge \neg V_0) \Big) \quad \text{in NNF}$$

Unter zusätzlicher Verwendung der "Distributivitätsregel" (Satz 3.25 (5)) erhält man Verfahren zur Transformation einer gegebenen Formel in eine äquivalente Formel in DNF bzw KNF, bei denen man nicht zuerst eine Wahrheitstafel aufstellen muss. Diese Verfahren sind vor allem dann ratsam, wenn die gegebene Formel sehr viele verschiedene Variablen enthält, die zugehörige Wahrheitstafel also sehr groß wird.

Algorithmus 3.33 (Ein KNF-Algorithmus)

Eingabe: Eine aussagenlogische Formel φ

Ausgabe: Eine zu φ äquivalente Formel φ' in KNF

Verfahren:

1) Konstruiere eine zu φ äquivalente Formel φ' in NNF (beispielsweise mit dem in Beobachtung 3.31 beschriebenen Verfahren)
2) Wiederhole folgende Schritte:
3) Falls φ' in KNF ist, so halte mit Ausgabe φ'
4) Ersetze eine Teilformel von φ' der Gestalt $(\psi_1 \vee (\psi_2 \wedge \psi_3))$ durch $((\psi_1 \vee \psi_2) \wedge (\psi_1 \vee \psi_3))$ oder ersetze eine Teilformel von φ' der Gestalt $((\psi_2 \wedge \psi_3) \vee \psi_1)$ durch $((\psi_2 \vee \psi_1) \wedge (\psi_3 \vee \psi_1))$. Sei φ'' die resultierende Formel
5) Setze $\varphi' := \varphi''$

Algorithmus 3.34 (Ein DNF-Algorithmus)

Eingabe: Eine aussagenlogische Formel φ

Ausgabe: Eine zu φ äquivalente Formel φ' in DNF

Verfahren:

1) Konstruiere eine zu φ äquivalente Formel φ' in NNF.
2) Wiederhole folgende Schritte:
3) Falls φ' in DNF ist, so halte mit Ausgabe φ'.
4) Ersetze eine Teilformel von φ' der Gestalt $(\psi_1 \wedge (\psi_2 \vee \psi_3))$ durch $((\psi_1 \wedge \psi_2) \vee (\psi_1 \wedge \psi_3))$ oder ersetze eine Teilformel von φ' der Gestalt $((\psi_2 \vee \psi_3) \wedge \psi_1)$ durch $((\psi_2 \wedge \psi_1) \vee (\psi_3 \wedge \psi_1))$.
   Sei φ'' die resultierende Formel.
5) Setze $\varphi' := \varphi''$

Satz 3.35 (Korrektheit der Algorithmen 3.33 und 3.34)

Für jede aussagenlogische Formel φ gilt:

(a) Algorithmus 3.33 hält bei Eingabe der Formel φ nach endlich vielen Schritten an und gibt eine zu φ äquivalente Formel in KNF aus.

(b) Algorithmus 3.34 hält bei Eingabe der Formel φ nach endlich vielen Schritten an und gibt eine zu φ äquivalente Formel in DNF aus.

(hier ohne Beweis)

Beispiel 3.36

Sei $\varphi := \big((\neg V_0 \wedge (V_0 \to V_1)) \wedge (V_2 \to V_3)\big)$

Transformation von φ in NNF:

$$\varphi = \big((\neg V_0 \wedge (V_0 \to V_1)) \vee (V_2 \to V_3)\big)$$
$$\equiv \big((\neg V_0 \wedge (\neg V_0 \vee V_1)) \vee (\neg V_2 \vee V_3)\big) =: \varphi'$$

Transformation von φ in DNF mittels Algorithmus 3.34:

Schritt 1: Transformation von φ in NNF
$\rightsquigarrow$ liefert $\varphi' = \big((\neg V_0 \wedge (\neg V_0 \vee V_1)) \vee (\neg V_2 \vee V_3)\big)$

1-maliges Anwenden von Zeile 4 des Algo. auf φ' liefert:

$$\varphi'' := \big(\big((\neg V_0 \wedge \neg V_0) \vee (\neg V_0 \wedge V_1)\big) \vee (\neg V_2 \vee V_3)\big)$$

Diese Formel ist die DNF-Formel, die von dem Algo. ausgegeben wird.

Transformation von φ in KNF mittels Algorithmus 3.33:

Schritt 1: Transformation von φ in NNF
$\rightsquigarrow$ liefert $\varphi' = \big((\neg V_0 \wedge (\neg V_0 \vee V_1)) \vee (\neg V_2 \vee V_3)\big)$

1-maliges Anwenden von Zeile 4 des Algo. liefert:

$$\varphi'' := \big((\neg V_0 \vee (\neg V_2 \vee V_3)) \wedge ((\neg V_0 \vee V_1) \vee (\neg V_2 \vee V_3))\big)$$

Dies ist die KNF-Formel, die von dem Algo. ausgegeben wird.

Unmittelbar vor Definition 3.15 wurde darauf hingewiesen, dass die Aufgabe, für eine gegebene Formel φ herauszufinden, ob sie erfüllbar ist, im Allgemeinen ein recht schwieriges Problem ist. Für den Spezialfall, dass φ eine Formel in **DNF** ist, lässt sich das Erfüllbarkeitsproblem allerdings sehr effizient lösen, wie die folgende Beobachtung zeigt.

Beobachtung 3.37: (effizienter Erfüllbarkeitstest für DNF-Formeln)

Sei φ eine Formel in DNF, d.h. φ ist von der Form

$$\bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} l_{i,j} \right), \quad \text{für Literale } l_{i,j}.$$

D.h. φ ist von der Form

$$\alpha_1 \vee \cdots \vee \alpha_n, \quad \text{wobei}$$

für jedes $i \in \{1, \ldots, n\}$ α_i die konjunktive Klausel

$$\alpha_i := l_{i,1} \wedge \cdots \wedge l_{i,m}$$

ist.

Klar: φ ist erfüllbar $\Leftrightarrow$ Für mindestens ein $i \in \{1, \ldots, n\}$ ist α_i erfüllbar.

Da α_i eine Konjunktion von Literalen (d.h. von Variablen und/oder negierten Variablen) ist, gilt:

K_i ist erfüllbar $\Leftrightarrow$ es gibt keine $j, j' \in \{1, \dots, m_i\}$, so dass $\ell_{i,j} = \neg \ell_{i,j'}$.

Daher ist der folgende Algorithmus dazu geeignet, zu testen, ob eine gegebene DNF-Formel erfüllbar ist.

Eingabe: Eine aussagenlogische Formel $\varphi = \bigvee_{i=1}^{n} \left(\bigwedge_{j=1}^{m_i} \ell_{i,j} \right)$ in DNF

Ziel: Entscheide, ob φ erfüllbar ist

Verfahren:

1.) Für $i = 1, \dots, n$

2.) Für $j = 1, \dots, m_i$

3.) Für $j' = j+1, \dots, m_i$

4.) Falls $\ell_{i,j} = \neg \ell_{i,j'}$ oder $\ell_{i,j'} = \neg \ell_{i,j}$,
dann (falls $i = n$ ist, so mache in Zeile 6 weiter;
ansonsten setze $i := i+1$ und mache in Zeile 2 weiter)

5.) Halte mit Ausgabe "φ ist erfüllbar"

6.) Halte mit Ausgabe "φ ist unerfüllbar"

Um aussagenlogische Formeln φ von beliebiger Form auf Erfüllbarkeit zu testen, kann man dann folgendermaßen vorgehen:

Schritt 1: Transformiere φ in eine äquivalente Formel φ' in DNF (z.B. mit Algo. 3.34)

Schritt 2: Entscheide, ob φ' erfüllbar ist (z.B. mit dem obigen Verfahren)

Das Durchführen von Schritt 1 kann dabei u.U. aber leider wieder sehr lange dauern, da es einige Formeln gibt, zu denen äquivalente Formeln in DNF zwangsläufig sehr groß sind. Dies wird durch den folgenden Satz präzisiert:

Satz 3.38:

Sei $n \in \mathbb{N}_{>0}$, seien $X_1, \dots, X_n, Y_1, \dots, Y_n$ $2n$ verschiedene aussagenlogische Variablen, und sei

$$\varphi_n := \bigwedge_{i=1}^{n} (X_i \leftrightarrow Y_i).$$

Dann hat jede zu φ_n äquivalente Formel in DNF mindestens 2^n konjunktive Klauseln.

Beweis: Übung.