3. Logik

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")
- letztendlich können Computer als "logische Systeme" betrachtet werden

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 dabei sein.

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

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

Die "Elementaraussagen", 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.

<u>Definition 3.2</u>: (Aussagenvariablen und Alphabet der Aussagenlogik)

(a) Eine <u>Aussagenvariable</u> (kurz: <u>Variable</u>) 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

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

Definition 3.3 (aussagenlogische Formeln: Syntax)

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

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

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

Notation 3.4

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

(b) Die Symbole $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ heißen <u>Junktoren</u>

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

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

<u>Beispiel 3.5</u>

Beispiele für Formeln (dh 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 <u>keine</u> Formel (d.h kein Element in AL) (da die Klammern fehlen). Auch $(\neg V_1)$ ist <u>keine</u> Formel (da die Klammern "zu viel sind").

□ Ende Bsp 3.5

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

Definition 3.6

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

<u>Beispiel</u>:

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

Definition 3.7

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

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

<u>Intuitive Bedeutung</u>: 1 steht für "wahr", 0 steht für "falsch".

<u>Definition 3.8</u> (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 <u>Wahrheitswert</u> (kurz: <u>Wert</u>) $[\![\varphi]\!]^{\mathcal{B}} \in \{0,1\}$ zuordnet:

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

<u>Rekursionsschritt</u>:
- 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
  - $[\![(\varphi \wedge \psi)]\!]^{\mathcal{B}} := \begin{cases} 1 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 1 \text{ und } [\![\psi]\!]^{\mathcal{B}} = 1 \\ 0 & \text{sonst} \end{cases}$
  - $[\![(\varphi \vee \psi)]\!]^{\mathcal{B}} := \begin{cases} 1 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 1 \text{ oder } [\![\psi]\!]^{\mathcal{B}} = 1 \\ 0 & \text{sonst} \end{cases}$
  - $[\![(\varphi \rightarrow \psi)]\!]^{\mathcal{B}} := \begin{cases} 1 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = 0 \text{ oder } [\![\psi]\!]^{\mathcal{B}} = 1 \\ 0 & \text{sonst} \end{cases}$
  - $[\![(\varphi \leftrightarrow \psi)]\!]^{\mathcal{B}} := \begin{cases} 1 & \text{falls } [\![\varphi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}} \\ 0 & \text{sonst} \end{cases}$

Intuitive Bedeutung der Semantik:

- <u>Atome</u>: **1** und **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.

- <u>Negation</u>: $\neg\varphi$ bedeutet "nicht φ".

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

  Darstellung durch eine so genannte <u>Verknüpfungstafel</u> (bzw. <u>Wahrheitstafel</u>):

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

- <u>Konjunktion</u>: $(\varphi \wedge \psi)$ bedeutet "φ und ψ".

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

- <u>Disjunktion</u>: $(\varphi \vee \psi)$ 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

- <u>Implikation</u>: $(\varphi \to \psi)$ 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

- <u>Biimplikation</u>: $(\varphi \leftrightarrow \psi)$ 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 $\varphi := (\neg V_0 \vee (V_5 \to V_1))$.

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

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

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

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

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

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

<u>Notation 3.10:</u>

- 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 $(\cdots((\varphi_1 \wedge \varphi_2) \wedge \varphi_3) \wedge \cdots \wedge \varphi_n)$ (analog für "$\vee$" an Stelle von "$\wedge$").

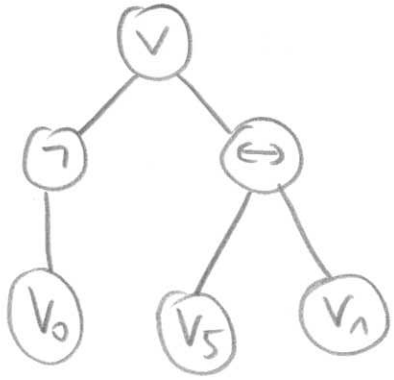
- Die äußeren Klammern einer Formel lassen wir manchmal weg und schreiben z.B. $(A \wedge B) \to C$ an Stelle des (formal korrekten) $((A \wedge B) \to C)$

- Ist φ eine Formel und $\mathcal{B}$ eine Belegung für φ, so sagen wir "<u>$\mathcal{B}$ erfüllt φ</u>" (bzw. "$\mathcal{B}$ ist eine <u>erfüllende Belegung</u> für φ", falls $[\![\varphi]\!]^{\mathcal{B}} = 1$.

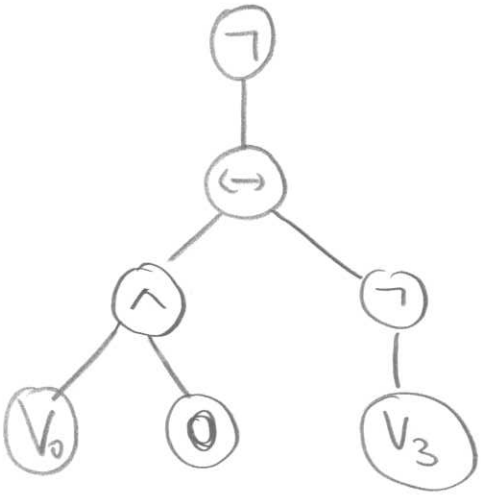
<u>Bemerkung</u> (Syntaxbäume zur graphischen Darstellung von Formeln)

Die Struktur einer Formel lässt sich bequem durch einen <u>Syntaxbaum</u> (englisch: <u>parse tree</u>) 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:

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

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:

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

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 <u>Wahrheitstafeln</u> zu benützen

<u>Wahrheitstafeln</u>:

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 $[\![\varphi]\!]^{\mathcal{B}}$ enthält.

Um die Wahrheitstafel für φ auszufüllen, ist es bequem, auch Spalten für (alle oder einige) "Teilformeln" von φ einzufügen

<u>Beispiel</u>: 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

<u>Beispiel</u>: 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 <u>erfüllenden</u> 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

<u>Beispiel 3.13</u>:

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 rausuchen, in denen in der mit "φ" beschrifteten Spalte der Wert 1 steht
3) aus diesen Zeilen all jene rausuchen, 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	

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

In der Wahrheitstafel sieht man, dass es <u>keine</u> erfüllende Belegung gibt, bei der in den mit A bis E beschrifteten Spalten insgesamt 5 Einsen stehen, und dass es genau <u>zwei</u> 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 $Def(\mathcal{B}) = Var(\varphi)$

<u>Beweis</u>:

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

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

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

Wir wissen, dass

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

$\square$

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:

<u>Satz</u>: Das aussagenlogische Erfüllbarkeitsproblem ist NP-vollständig.

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

<u>Eingabe</u>: eine aussagenlogische Formel φ

<u>Frage</u>: Gibt es eine erfüllende Belegung für φ?

Was der Begriff "<u>NP-vollständig</u>" <u>genau</u> bedeutet, werden Sie in der "Algorithmentheorie"-Vorlesung lernen; grob gesagt bedeutet "NP-vollständig", dass es "wahrscheinlich keinen <u>effizienten</u> 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:

<u>Definition 3.15</u>

Sei φ eine aussagenlogische Formel.

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

(b) φ heißt <u>unerfüllbar</u>, wenn es <u>keine</u> erfüllende Belegung für φ gibt.

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