Was ist Logik?

Logik (nach dem Altgriechischen „Logos“: „Vernunft“) ist

die Lehre des vernünftigen Schlussfolgerns.

Die für die Logik zentrale Frage:

> *Wie kann man Aussagen miteinander verknüpfen, und auf welche Weise kann man formal Schlüsse ziehen und Beweise führen?*

Logik in der Informatik

Logik wird in der Informatik u.a. genutzt

- für die **Beschreibung, Analyse, Optimierung** und **Verifikation** digitaler Schaltungen,
- für den **Nachweis**, dass ein Programm gewisse wünschenswerte Eigenschaften hat,
- für die **Wissensrepräsentation**, z.B. im Bereich der künstlichen Intelligenz,
- als Grundlage für **Datenbank-Anfragesprachen**,
- für die **automatische Erzeugung von Beweisen** in so genannten „Theorembeweisern“.

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.

Geburtstagsfeier (1/3)

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.

Wie viele Freunde, und welche, werden im besten Fall zur Party kommen?

Geburtstagsfeier (2/3)

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

Um welche "atomaren Aussagen" dreht sich der Text?

$$\begin{array}{lcl}
A & \widehat{=} & \text{Anne kommt zur Feier} \\
B & \widehat{=} & \text{Bernd kommt zur Feier} \\
C & \widehat{=} & \text{Christine kommt zur Feier} \\
D & \widehat{=} & \text{Dirk kommt zur Feier} \\
E & \widehat{=} & \text{Eva kommt zur Feier.}
\end{array}$$

Das im Text zusammengefasste „Wissen“ lässt sich wie folgt repräsentieren:

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

Die Aussagenlogik liefert einen Formalismus, mit dessen Hilfe man solches „Wissen“ modellieren und Schlüsse daraus ziehen kann.

Syntax und Semantik

Syntax und Semantik: Formeln und ihre Bedeutung

- Die **Syntax** legt fest, welche Zeichenketten Formeln der Aussagenlogik sind.
- Die **Semantik** legt fest, welche „Bedeutung“ die einzelnen Formeln haben.

Für Programmiersprachen ist die Situation ähnlich:

- Die Syntax legt fest, welche Zeichenketten korrekte Programme sind,
- während die Semantik bestimmt, was das Programm tut.

Die Syntax der Aussagenlogik

Griechische Buchstaben

In der Literatur werden Formeln einer Logik traditionell meistens mit griechischen Buchstaben bezeichnet.

Buchstabe	ϕ	ψ	χ	θ *bzw.* ϑ	λ	μ	ν	τ
Aussprache	*phi*	*psi*	*chi*	*theta*	*lambda*	*mü*	*nü*	*tau*

Buchstabe	κ	σ	ρ	ξ	ζ	α	β	γ
Aussprache	*kappa*	*sigma*	*rho*	*xi*	*zeta*	*alpha*	*beta*	*gamma*

Buchstabe	δ	ω	ε	ι	π	Δ	Γ
Aussprache	*delta*	*omega*	*epsilon*	*iota*	*pi*	*Delta*	*Gamma*

Buchstabe	Σ	Π	Φ
Aussprache	*Sigma*	*Pi*	*Phi*

Aussagenvariablen

(a) Eine **Aussagenvariable** (kurz: Variable) hat die Form V_i für $i \in \mathbb{N}$.
Die Menge aller Aussagenvariablen bezeichnen wir mit AVAR. D.h.:

$$\text{AVAR} \; = \; \{V_i \; : \; i \in \mathbb{N}\} \; = \; \{V_0, V_1, V_2, V_3, \ldots\}.$$

(b) Das **Alphabet** A_{AL} der Aussagenlogik ist

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

Und was sind aussagenlogische Formeln?

Aussagenlogische Formeln

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

Basisregeln:

(B0) $\mathbf{0} \in \mathrm{AL}$.

(B1) $\mathbf{1} \in \mathrm{AL}$.

(BV) Für jede Variable $X \in \mathrm{AVAR}$ gilt: $X \in \mathrm{AL}$.

Rekursive Regeln:

(R1) Ist $\phi \in \mathrm{AL}$, so ist auch $\neg\phi \in \mathrm{AL}$.

(R2) Ist $\phi \in \mathrm{AL}$ und $\psi \in \mathrm{AL}$, so ist auch

- $(\phi \wedge \psi) \in \mathrm{AL}$
- $(\phi \vee \psi) \in \mathrm{AL}$
- $(\phi \rightarrow \psi) \in \mathrm{AL}$
- $(\phi \leftrightarrow \psi) \in \mathrm{AL}$.

Eine Formel ϕ gehört genau dann zu AL, wenn man ϕ durch (möglicherweise mehrfache) Anwendung der obigen Regeln erzeugen kann.

Beispiele

- $(\neg V_0 \vee (V_5 \rightarrow V_1))$ ist eine korrekt gebildete Formel.
- $V_1 \vee V_2 \vee V_3$ ist keine korrekt gebildete Formel, da die Klammern fehlen,
  - In diesem Fall bleibt die Bedeutung aber klar (warum?).
- $\neg((V_0 \wedge \mathbf{0}) \leftrightarrow \neg V_3)$ ist hingegen wieder eine korrekte Formel,
- $(\neg V_1)$ ist keine korrekte Formel, da sie zu viele Klammern besitzt.
  - Aber auch in diesem Fall bleibt die Bedeutung klar.
- Aber was soll $V_0 \wedge V_1 \vee V_2$ bedeuten?
  - Möglicherweise $(V_0 \wedge V_1) \vee V_2$
  - oder $V_0 \wedge (V_1 \vee V_2)$?

Übrigens, von welcher Bedeutung reden wir überhaupt?

Bezeichnungen

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

(b) Die Symbole $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ heißen **Junktoren**.

(c) Sind ϕ und ψ Formeln (d.h. $\phi \in \text{AL}$ und $\psi \in \text{AL}$), so heißt:

- $\neg\phi$ **Negation** (bzw. Ver*nein*ung) von ϕ,
- $(\phi \wedge \psi)$ **Konjunktion** (bzw. Ver*und*ung) von ϕ und ψ,
- $(\phi \vee \psi)$ **Disjunktion** (bzw. Ver*oder*ung) von ϕ und ψ,
- $(\phi \rightarrow \psi)$ **Implikation** und
- $(\phi \leftrightarrow \psi)$ **Biimplikation** (oder Äquivalenz).

Das Leben ist schon so hart genug!

- Statt $V_0, V_1, V_2, \ldots$ bezeichnen wir Variablen oft auch mit $A, B, C, \ldots, X, Y, Z, \ldots$ oder mit Variablen wie $X', Y_1, \ldots$
- Wir schreiben $\bigwedge_{i=1}^{n} \phi_i$ bzw. $\phi_1 \wedge \cdots \wedge \phi_n$ an Stelle von $\Big(\big((\phi_1 \wedge \phi_2) \wedge \phi_3\big) \wedge \cdots \wedge \phi_n\Big)$
  - 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) $\big((A \wedge B) \to C\big)$.

Wenn Sie eine Formel „vereinfachen“, stellen Sie sicher, dass

die Bedeutung eindeutig ist.

Beispiel für die Formulierung von Formeln (1/2)

Wir möchten die Zeugenaussage

> *„Das Fluchtauto war rot oder grün und*
> *hatte weder vorne noch hinten ein Nummernschild.“*

durch eine aussagenlogische Formel repräsentieren. Dazu verwenden wir die folgenden atomaren 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.

Wir repräsentieren die Zeugenaussage durch

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

Wir greifen das Beispiel der Geburtstagsfeier nochmal auf.

Atomare Aussagen:

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

Die **vollständige Wissensrepräsentation** wird durch folgende Formel gegeben:

$$\begin{aligned} \phi \quad := \quad & \big(E \rightarrow (C \wedge D)\big) \;\wedge\; (C \rightarrow A) \;\wedge\; \big((B \wedge E) \rightarrow \neg D\big) \;\wedge \\ & \big(A \rightarrow (B \vee C)\big) \;\wedge\; \big((B \wedge A) \rightarrow \neg E\big). \end{aligned}$$

Wie viele Personen kommen bestenfalls zur Party?

Die Semantik der Aussagenlogik

Die Variablen einer Formel

Die **Variablenmenge**

$$\mathrm{Var}(\phi)$$

einer aussagenlogischen Formel ϕ ist die Menge aller Variablen $X \in \mathrm{AVAR}$, die in ϕ vorkommen.

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

Belegungen

(a) Eine **partielle** Funktion

$$\mathcal{B} : \text{AVAR} \to \{0,1\}$$

von AVAR nach $\{0,1\}$ heißt eine

Belegung,

bzw. Wahrheitsbelegung.

- 1 steht für den Wert „wahr" und 0 für den Wert „falsch".

(b) Eine Belegung $\mathcal{B}$ ist eine Belegung für die Formel ϕ (bzw. **passend** zu ϕ), wenn $\mathcal{B}$ auf allen Variablen von ϕ definiert ist.

Eine Beispiel: Die Funktion $\mathcal{B}$

$$\text{mit } \mathcal{B}(V_0) = 1, \mathcal{B}(V_1) = 1 \text{ und } \mathcal{B}(V_3) = 0$$

besitzt den Definitionsbereich $\{V_0, V_1, V_3\}$. $\mathcal{B}$ ist eine Belegung für

$$\big(V_0 \wedge (V_1 \vee V_3)\big) \text{ oder } (V_0 \wedge V_3), \text{ nicht aber für } (V_0 \wedge V_2).$$

Wir definieren eine Funktion $[\![\phi]\!]^{\mathcal{B}}$,

- die jeder Formel $\phi \in \mathsf{AL}$
- und jeder zu ϕ passenden Belegung $\mathcal{B}$

einen **Wahrheitswert** (kurz: Wert) $[\![\phi]\!]^{\mathcal{B}} \in \{0, 1\}$ zuordnet.

Wir benutzen wieder, wie im Fall der Definition der Syntax,

eine rekursive Definition über den Aufbau aussagenlogischer Formeln.

1. Zuerst definieren wir Wahrheitswerte für atomare Formeln
2. und dann für Negationen, Konjunktionen, Disjunktionen, Implikationen und Äquivalenzen.

Rekursionsanfang:

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

Rekursionsschritt:

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

Beispiel

Betrachte die Formel

$$\phi \ := \ \big(\neg V_0 \vee (V_5 \rightarrow V_1)\big).$$

Dann ist beispielsweise die Funktion

$$\mathcal{B}\colon \{V_0, V_1, V_5\} \rightarrow \{0, 1\}$$

mit

$$\mathcal{B}(V_0) := 1, \mathcal{B}(V_1) := 1 \text{ und } \mathcal{B}(V_5) := 0$$

eine Belegung für ϕ.

Bestimme den Wahrheitswert $[\![\phi]\!]^{\mathcal{B}}$ von ϕ unter Belegung $\mathcal{B}$.

Wie bestimmt man Wahrheitswerte?
Zum Beispiel mit Wahrheitstafeln!

Wahrheitstafel (1/2)

Für jede Formel ψ kann man

die Wahrheitswerte von ψ unter allen möglichen Belegungen

in einer Wahrheitstafel (manchmal auch Funktionstafel genannt) darstellen.

(a) Die **Wahrheitstafel** hat für jede Belegung $\mathcal{B} : \text{Var}(\psi) \to \{0, 1\}$ eine Zeile.

(b) Die Zeile für $\mathcal{B}$ enthält mindestens

1. f.a. $X \in \text{Var}(\psi)$ die Werte $\mathcal{B}(X)$ und
2. den Wert $[\![\psi]\!]^{\mathcal{B}}$.

Um die Wahrheitstafel für ϕ auszufüllen, ist es ratsam,
auch Spalten für (alle oder einige) „Teilformeln“ von ϕ einzufügen.

(a) Wahrheitstafel für $\phi := \big(\neg V_0 \vee (V_5 \to V_1)\big)$:

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

(b) Wahrheitstafel für $\phi := \Big(X \wedge ((\mathbf{1} \to \mathbf{0}) \to \mathbf{0})\Big)$:

X	$\mathbf{1}$	$\mathbf{0}$	$(\mathbf{1} \to \mathbf{0})$	$((\mathbf{1} \to \mathbf{0}) \to \mathbf{0})$	ϕ
0	1	0	0	1	0
1	1	0	0	1	1

- **Atomare Formeln**:
  - **1** und **0** bedeuten einfach „wahr“ und „falsch“.
  - Die Variablen $X \in$ AVAR stehen für irgendwelche Aussagen.
    Uns interessiert hier nur, ob diese Aussagen „wahr“ oder „falsch“ sind — und dies wird durch eine Belegung $\mathcal{B}$ angegeben.
- **Negation**: $\neg\phi$ bedeutet „*nicht* ϕ“.

  Zugehörige Wahrheitstafel:

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

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

  Zugehörige Wahrheitstafel:

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

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

  Zugehörige Wahrheitstafel:

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

- **Implikation**: $(\phi \rightarrow \psi)$ bedeutet „ϕ *impliziert* ψ“, d.h. „*wenn* ϕ, *dann auch* ψ“.

  Zugehörige Wahrheitstafel:

$[\![\phi]\!]^{\mathcal{B}}$	$[\![\psi]\!]^{\mathcal{B}}$	$[\![(\phi \rightarrow \psi)]\!]^{\mathcal{B}}$
0	0	1
0	1	1
1	0	0
1	1	1

- **Äquivalenzen**: $(\phi \leftrightarrow \psi)$ bedeutet „ϕ *gilt genau dann, wenn* ψ gilt“.

  Zugehörige Wahrheitstafel:

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

„Oder“ versus „Entweder oder“

Der Sprachgebrauch für „oder“ ist unscharf und meint manchmal wie oben ein „inklusives oder“, bzw das „exklusive oder“.

(a) Das „**inklusive oder**“: Die Formel $\phi \vee \psi$ ist nur dann falsch unter einer Belegung $\mathcal{B}$, wenn ϕ wie auch ψ unter $\mathcal{B}$ falsch sind, d.h. wenn $[\![\phi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}} = 0$.

(b) Das „**exklusive oder**“ (oder „**XOR**“): Man verwendet den neuen Junktor $\oplus$ und definiert seine Bedeutung durch
$$[\![\phi \oplus \psi]\!]^{\mathcal{B}} := \begin{cases} 0, & \text{falls } [\![\phi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}} \\ 1, & \text{sonst.} \end{cases}$$

Unter welchen Belegungen $\mathcal{B}$ ist

$$[\![V_1 \oplus (V_2 \oplus V_3)]\!]^{\mathcal{B}}$$

wahr? Sind Klammern notwendig?

Zurück zum „Geburtstagsproblem“

Sei ϕ die Formel aus dem Geburtagsproblem. Die Frage

> *„Wie viele (und welche) Freunde werden bestenfalls zur Party kommen?“*

können wir lösen, in dem wir

1. die Wahrheitstafel für ϕ ermitteln,
2. alle Zeilen heraussuchen,
   in denen in der mit „ϕ“ beschrifteten Spalte der Wert 1 steht und
3. aus diesen Zeilen all jene heraussuchen, 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 gleichzeitig erscheinenden Freunden.

A	B	C	D	E	$E \to (C \land D)$	$C \to A$	$(B \land E) \to \neg D$	$A \to (B \lor C)$	$(B \land A) \to \neg E$	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

Das Geburtstagsproblem: Die Lösung

Es gibt keine Zeile mit genau 5 Einsen, aber genau zwei Zeilen mit insgesamt 4 Einsen in den mit A bis E beschrifteten Spalten:

Es handelt sich um die beiden Belegungen $\mathcal{B}_1$ und $\mathcal{B}_2$ mit

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

und

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

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.

Widersprüche und Tautologien

Erfüllbare, widersprüchliche und allgemeingültige Formeln

Sei ϕ eine aussagenlogische Formel.

(a) ϕ heißt **erfüllbar**, wenn es **mindestens eine** erfüllende Belegung $\mathcal{B}$ für ϕ gibt.

- Die Belegung $\mathcal{B}$ passt zu ϕ und es gilt $[\![\phi]\!]^{\mathcal{B}} = 1$.

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

(c) ϕ heißt **allgemeingültig** (oder eine **Tautologie**), wenn **jede** zu ϕ passende Belegung ϕ erfüllt,

- d.h. wenn $[\![\phi]\!]^{\mathcal{B}} = 1$ für jede zu ϕ passende Belegung $\mathcal{B}$ gilt.

Beispiele

(a) Die Formel

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

ist **erfüllbar**,

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

(b) Die Formel

$$(X \wedge \neg X)$$

ist **unerfüllbar** (oder **widersprüchlich**), da für jede zur Formel passenden Belegung $\mathcal{B}$ entweder $\mathcal{B}(X) = 1$ oder $\mathcal{B}(X) = 0$ gilt.

- Ist $\mathcal{B}(X) = 1$, so gilt: $[\![(X \wedge \neg X)]\!]^{\mathcal{B}} = 1 \wedge 0 = 0$,
- Ist $\mathcal{B}(X) = 0$, so gilt: $[\![(X \wedge \neg X)]\!]^{\mathcal{B}} = 0 \wedge 1 = 0$.

(c) Die Formel

$$(X \vee \neg X)$$

ist **allgemeingültig**, da für jede zur Formel passenden Belegung $\mathcal{B}$ entweder $\mathcal{B}(X) = 1$ oder $\mathcal{B}(X) = 0$ gilt.

Und nochmals Wahrheitstafeln

Für jede aussagenlogische Formel ϕ gilt:

(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.
(d)	ϕ ist allgemeingültig	$\iff$	$\neg\phi$ ist unerfüllbar.

Semantische Folgerung und Äquivalenz

(Semantische) Folgerung

ϕ und ψ seien zwei aussagenlogische Formeln.

Wir sagen ψ **folgt aus** ϕ bzw. ϕ **impliziert** ψ und schreiben

$$\phi \models \psi,$$

- falls

  $$[\![\phi \to \psi]\!]^{\mathcal{B}} = 1$$

  für **jede** zu ϕ und ψ passende Belegung $\mathcal{B}$ gilt.
- Oder äquivalent dazu: Für **jede** Belegung $\mathcal{B}$ mit $[\![\phi]\!]^{\mathcal{B}} = 1$ gilt stets $[\![\psi]\!]^{\mathcal{B}} = 1$, bzw.

  $$\phi \to \psi \text{ ist allgemeingültig.}$$

- Oder äquivalent dazu: In jeder Zeile der Wahrheitstafel, in der eine 1 in der Spalte von ϕ steht, muss auch eine 1 in der Spalte von ψ stehen.

Ein Beispiel

Wir betrachten

$$\phi := \big((X \vee Y) \wedge (\neg X \vee Y)\big) \text{ und } \psi := \big(Y \vee (\neg X \wedge \neg Y)\big).$$

- Wir stellen die Wahrheitstafel für ϕ und ψ auf:

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

- In jeder Zeile, in der eine 1 in der Spalte von „ϕ“ steht, steht auch in der Spalte von „ψ“ eine 1.
  - Somit gilt $\phi \models \psi$.
- Aber in Zeile 1, also für die Belegung $\mathcal{B}$ mit $\mathcal{B}(X) = 0$ und $\mathcal{B}(Y) = 0$, steht in der Spalte von “ψ“ eine 1 und in der Spalte von “ϕ“ eine 0.
  - Es gilt also $[\![\psi]\!]^{\mathcal{B}} = 1$ und $[\![\phi]\!]^{\mathcal{B}} = 0$.
  - Daher gilt $\psi \not\models \phi$.

Wie passt das alles zusammen?

Seien ϕ und ψ beliebige aussagenlogische Formeln. Dann gilt:

(a) $\mathbf{1} \models \phi \iff \phi$ ist allgemeingültig,

(b) $\phi \models \mathbf{0} \iff \phi$ ist unerfüllbar,

(c) $\phi \models \psi \iff (\phi \to \psi)$ ist allgemeingültig und

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

Logische Äquivalenz

Zwei aussagenlogische Formeln ϕ und ψ heißen **äquivalent**, kurz:

$$\phi \equiv \psi,$$

wenn für alle zu ϕ und ψ passenden Belegungen $\mathcal{B}$ gilt:

$$[\![\phi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}}.$$

ϕ und ψ sind genau dann äquivalent, wenn in ihrer Wahrheitstafel

die Spalten von ϕ und ψ übereinstimmen,

denn genau dann gilt $[\![\phi]\!]^{\mathcal{B}} = [\![\psi]\!]^{\mathcal{B}}$ für alle zu ϕ und ψ passenden Belegungen $\mathcal{B}$.

Ein Beispiel

Sind

$$\phi := \big(X \wedge (X \vee Y)\big) \text{ und } \psi := X$$

äquivalent?

- Wir stellen die Wahrheitstafel auf:

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

- Die Spalte von „ϕ“ stimmt überein mit der Spalte von „ψ“: Es folgt

$$\phi \equiv \psi.$$

Wie passt das alles zusammen?

Seien ϕ und ψ aussagenlogische Formeln. Dann gilt:

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

(b) ϕ ist allgemeingültig $\iff \phi \equiv \mathbf{1}$.

(c) ϕ ist erfüllbar $\iff \phi \not\equiv \mathbf{0}$, d.h. „$\phi \equiv \mathbf{0}$“ gilt nicht.

Fundamentale Äquivalenzen (1/2)

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

(a) **Idempotenz:**
- $(\phi \wedge \phi) \equiv \phi$
- $(\phi \vee \phi) \equiv \phi$

(b) **Kommutativität:**
- $(\phi \wedge \psi) \equiv (\psi \wedge \phi)$
- $(\phi \vee \psi) \equiv (\psi \vee \phi)$

(c) **Assoziativität:**
- $((\phi \wedge \psi) \wedge \chi) \equiv (\phi \wedge (\psi \wedge \chi))$
- $((\phi \vee \psi) \vee \chi) \equiv (\phi \vee (\psi \vee \chi))$

(d) **Absorption:**
- $(\phi \wedge (\phi \vee \psi)) \equiv \phi$
- $(\phi \vee (\phi \wedge \psi)) \equiv \phi$

(e) **Distributivität:**
- $(\phi \wedge (\psi \vee \chi)) \equiv ((\phi \wedge \psi) \vee (\phi \wedge \chi))$
- $(\phi \vee (\psi \wedge \chi)) \equiv ((\phi \vee \psi) \wedge (\phi \vee \chi))$

(f) **Doppelte Negation:**
- $\neg\neg\phi \equiv \phi$

(g) **De Morgansche Regeln:**
- $\neg(\phi \wedge \psi) \equiv (\neg\phi \vee \neg\psi)$
- $\neg(\phi \vee \psi) \equiv (\neg\phi \wedge \neg\psi)$

(h) **Tertium non Datur:**
- $(\phi \wedge \neg\phi) \equiv \mathbf{0}$
- $(\phi \vee \neg\phi) \equiv \mathbf{1}$

(i) **True/False 1**
- $(\phi \wedge \mathbf{1}) \equiv \phi$
- $(\phi \wedge \mathbf{0}) \equiv \mathbf{0}$
- $(\phi \vee \mathbf{1}) \equiv \mathbf{1}$
- $(\phi \vee \mathbf{0}) \equiv \phi$

(j) **True/False 2**
- $\mathbf{1} \equiv \neg\mathbf{0}$
- $\mathbf{0} \equiv \neg\mathbf{1}$

(k) **Elimination der Implikation:**
- $(\phi \to \psi) \equiv (\neg\phi \vee \psi)$

(l) **Elimination der Äquivalenz:**
- $(\phi \leftrightarrow \psi) \equiv ((\phi \to \psi) \wedge (\psi \to \phi))$

Logisches Schließen

Wir können die obigen Folgerungen und Äquivalenzen
auch als Schlußregeln auffassen.

- Durch schrittweises Anwenden der Äquivalenzen kann man eine gegebene aussagenlogische Formel in eine zu ihr äquivalente Formel umformen.
- Durch schrittweises Anwenden der Folgerungen erhalten wir eine implizierte Formel.

Beispiel: Sind ϕ und ψ aussagenlogische Formeln, so gilt

$$\begin{aligned}(\phi \leftrightarrow \psi) \quad &\equiv \big((\phi \to \psi) \wedge (\psi \to \phi)\big) \\ &\equiv \big((\neg\phi \vee \psi) \wedge (\neg\psi \vee \phi)\big)\end{aligned}$$

Hoppela

Etwas zu den vielen Äquivalenzen Ähnliches haben wir doch schon im Abschnitt „Mengenalgebra“ gesehen!

Eine **boolesche Algebra** $(B, \wedge, \vee, \neg)$ besteht aus einer Trägermenge B und den Funktionen

$$\wedge, \vee : B \times B \to B, \text{ bzw. der Funktion } \neg : B \to B.$$

Es ist $0, 1 \in B$ und die folgenden Gleichheiten gelten für alle $a, b, c \in B$:

$$\begin{array}{ll} \text{Kommutativität:} & a \wedge b = b \wedge a \\ & a \vee b = b \vee a \\ \text{Distributivität:} & a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c) \\ & a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c) \\ \text{True/False 1:} & a \wedge 1 = a \\ & a \vee 0 = a \\ \text{True/False 2:} & \neg 0 = 1 \\ & \neg 1 = 0. \end{array}$$

Boolesche Algebra

Wir kennen bereits zwei Beispiele boolescher Algebren, nämlich

1. Die Mengenalgebra $(\mathcal{P}(U), \cap, \cup, \text{Komplement})$ für jede Menge U.
   - Die Mengenalgebra besteht aus allen Teilmengen von U und
   - besitzt die Operation „Durchschnitt“, „Vereinigung“ sowie die „Komplementbildung“.
2. Die Aussagenlogik $(\{\mathbf{0}, \mathbf{1}\}, \wedge, \vee, \neg)$.

Man trifft sich im Leben immer zweimal.

Wir bauen eine disjunktive Normalform aus einer Wahrheitstafel

Literale, Konjunktionsterme und disjunktive Normalformen

(a) Ein **Literal** ℓ ist eine Formel der Form $\ell = X$ oder $\ell = \neg X$ mit $X \in \text{AVAR}$, d.h. X ist eine Aussagenvariable.

- Das Literal X wird auch **positives Literal** genannt,
- das Literal $\neg X$ heißt **negatives Literal**.

(b) Eine Konjunktion von (positiven oder negativen) Literalen heißt ein **Konjunktionsterm** (bzw. eine **konjunktive Klausel**).

(c) Eine Disjunktion

$$\phi = \bigvee_{i=1}^{m} \left(\bigwedge_{j=1}^{m_i} \ell_{i,j} \right)$$

von Konjunktionstermen $\bigwedge_{j=1}^{m_i} \ell_{i,j}$ ist in

disjunktiver Normalform (DNF),

wenn $m, m_1, \ldots, m_n \in \mathbb{N}_{>0}$ und $\ell_{i,j}$ für jedes $i \in \{1, \ldots, m\}, j \in \{1, \ldots, m_i\}$ ein Literal ist.

Wahrheitstafel $\Rightarrow$ DNF

Die Wahrheitstafel T sei gegeben: Baue eine DNF ϕ mit T als Wahrheitstafel.

1. Sei z eine Zeile der Wahrheitstafel T
   - z heißt eine **1-Zeile**, wenn in der Spalte des Wahrheitswerts eine „1“ steht.
   - Steht in der Spalte des Wahrheitswerts aber eine „0“, nennen wir z eine **0-Zeile**.
2. Um eine DNF zu bauen,
   - baue zuerst **für jede 1-Zeile z** einen Konjunktionsterm, der von der Belegung von z erfüllt wird, aber von allen anderen Belegungen verworfen wird.
   - Dann ver*oder*e all diese Konjunktionsterme!

Ein Konjunktionsterm K ist ein **Minterm**, wenn

jede Variable entweder als positives oder negatives Literal in K vorkommt.

Die Wahrheitstafel T sei gegeben: Baue eine DNF ϕ mit T als Wahrheitstafel.

1. Für jede 1-Zeile z mit Belegung $\mathcal{B}_z : \text{AVAR} \to \{0,1\}$ konstruiere

   den Minterm von z,

   als den Konjunktionsterm

   der von $\mathcal{B}_z$ erfüllt, aber von allen anderen Belegungen verworfen wird.

   Der Minterm von z hat also die Form

   $$\Big(\bigwedge_{i:\mathcal{B}_z(V_i)=1} V_i \Big) \wedge \Big(\bigwedge_{j:\mathcal{B}_z(V_j)=0} \neg V_j \Big).$$

   .
2. Die Disjunktion

   $$\phi = \bigvee_{K \text{ ist der Minterm einer 1-Zeile}} K,$$

   (also die „Ver*oder*ung“ aller Minterme zu 1-Zeilen der Wahrheitstafel) heißt

   kanonische DNF.

DNFs: Ein erstes Beispiel

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

Es gibt genau drei 1-Zeilen (für die in der „Spalte von ϕ“ eine 1 steht), nämlich

X	Y	Z	ϕ	zur Belegung der Zeile gehörender Minterm:
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	$X \wedge \neg Y \wedge Z$

Wir erhalten die zur Wahrheitstafel T passende kanonische DNF

$$\phi \;:=\; (\neg X \wedge \neg Y \wedge \neg Z) \;\vee\; (X \wedge \neg Y \wedge \neg Z) \;\vee\; (X \wedge \neg Y \wedge Z).$$

Implikant und Primimplikant

Um mit DNFs effizient arbeiten zu können, versuchen wir DNFs mit möglichst wenigen Konjunktionstermen zu erhalten.

(a) Ein Konjunktionsterm M heißt ein **Implikant** der Formel ϕ, wenn

$$M \models \phi$$

gilt. Ein Implikant von ϕ „impliziert" also die Formel ϕ.

(b) Ein Implikant M heißt ein **Primimplikant**, wenn M nicht verkürzbar ist, wenn also M, nach Streichen irgendeines Literals kein Implikant von ϕ mehr ist.

Satz von Quine

Warum haben wir den Begriff eines Primimplikanten eingeführt?

Jede DNF mit einer kleinstmöglichen Anzahl von Konjunktionstermen
ist eine Disjunktion von Primimplikanten.

Beweis: Siehe Tafel.

In der Veranstaltung „**Hardware Architekturen und Rechensysteme**“ lernen Sie das Quine-McCluskey Verfahren kennen, das alle Primplikanten bestimmt und dann versucht eine möglichst kleine DNF zu „bauen“.

Im letzten Beispiel haben wir die kanonische DNF

$$\phi \;:=\; (\neg X \wedge \neg Y \wedge \neg Z) \vee (X \wedge \neg Y \wedge \neg Z) \vee (X \wedge \neg Y \wedge Z).$$

gefunden.

1. Welche Primimplikanten hat ϕ?
2. Besitzt ϕ eine kleinere DNF?

Antworten: Siehe Tafel.

$X_1, \ldots, X_n, Y_1, \ldots, Y_n$ seien aussagenlogische Variablen. Betrachte die Formel

$$\psi_n \quad := \quad \bigvee_{i=1}^{n} (X_i \oplus Y_i).$$

1. Wieviele 0-Zeilen hat die Wahrheitstafel für ϕ_n? Aus wievielen Konjunktionstermen besteht die zu ψ äquivalente kanonische KNF für ψ?
2. Wieviele Primimplikanten hat ψ?

ψ_n hat die DNF

$$\psi_n = \bigvee_{i=1}^{n} (X_i \wedge \neg Y_i) \vee \bigvee_{i=1}^{n} (\neg X_i \wedge Y_i)$$

mit $2n$ Konjunktionstermen, die kanonische DNF für ψ_n besteht aber aus 2^n Konjunktionstermen! Die Minimierung von DNFs ist ein wichtiges Problem.

Achtung, Achtung, Achtung!

Gibt es aber **immer** **kleine** DNFs?

$X_1, \ldots, X_n, Y_1, \ldots, Y_n$ seien aussagenlogische Variablen. Betrachte die Formel

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

Dann hat jede zu ϕ_n äquivalente Formel in DNF mindestens

$$\mathbf{2^n}$$

Konjunktionsterme.

Beweis: Siehe Übungen.

Die konjunktive Normalform

Disjunktionsterme und KNFs

- Ein **Disjunktionsterm** (oder eine **(disjunktive) Klausel**) ist eine Disjunktion von Literalen.
- Eine Konjunktion

$$\phi = \bigwedge_{i=1}^{m} \left(\bigvee_{j=1}^{m_i} \ell_{i,j} \right)$$

  von Disjunktionstermen $\bigvee_{j=1}^{m_i} \ell_{i,j}$ ist in **konjunktiver Normalform (KNF)**, wenn $m, m_1, \ldots, m_n \in \mathbb{N}_{>0}$ und $\ell_{i,j}$ für jedes $i \in \{1, \ldots, m\}, j \in \{1, \ldots, m_i\}$ ein Literal ist.

KNFs: Beispiele

DNFs für

$$\phi_n = \bigwedge_{i=1}^{n} X_i \leftrightarrow Y_i$$

benötigen viele Konjunktionsterme. Aber ϕ_n hat doch eine „kleine“ KNF **:-)))**

Wie erhält man eine konjunktive Normalform aus einer Wahrheitstafel?

Die Wahrheitstafel T sei gegeben: Baue eine KNF ϕ mit T als Wahrheitstafel.

1. Die Idee: Schließe die Belegungen aller 0-Zeilen mit Disjunktionstermen aus.
2. Wie schließen wir die Belegung einer 0-Zeile z aus?
   - baue einen Disjunktionsterm, der von der Belegung von z verworfen wird, aber von allen anderen Belegungen erfüllt wird.
   - Verwende den **negierten Minterm** von z.
3. Dann ver*unde* all negierten Minterme von 0-Zeilen!

Ein Disjunktionsterm D ist ein **Maxterm**, wenn

jede Variable entweder als positives oder negatives Literal in D vorkommt.

Wahrheitstafel $\Rightarrow$ KNF (2/2)

Die Wahrheitstafel T sei gegeben: Baue eine KNF ϕ mit T als Wahrheitstafel.

1. Für jede 0-Zeile z mit Belegung $\mathcal{B}_z : \text{AVAR} \to \{0,1\}$ konstruiere

   den Maxterm von z,

   als den Disjunktionsterm

   der von $\mathcal{B}_z$ verworfen, aber von allen anderen Belegungen erfüllt wird.

   Der Maxterm von z hat also die Form

   $$\Big(\bigvee_{i:\mathcal{B}_z(V_i)=1} \neg V_i \Big) \vee \Big(\bigvee_{j:\mathcal{B}_z(V_j)=0} V_j \Big).$$

   .
2. Die Konjunktion

   $$\phi = \bigwedge_{K \text{ ist der Maxterm einer 0-Zeile}} K,$$

   also die „Ver*und*ung" aller Maxterme zu 0-Zeilen der Wahrheitstafel, heißt

   kanonische KNF.

KNFs: Beispiele

Wir betrachten wieder 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

Um die kanonische KNF zu bestimmen,
müssen wir die Maxterme zu allen 0-Zeilen bestimmen und verunden.

Wie sieht die kanonische KNF in diesem Beispiel aus? Siehe Tafel

DNFs und KNFs

Normalformen spielen in vielen Anwendungsgebieten eine wichtige Rolle.

- Beispielsweise geht man in der Schaltungstechnik (Hardware-Entwurf) oft von DNF- oder KNF-Formeln aus,
- während bei der Wissensrepräsentation oftmals nur KNF-Formeln auftreten, da sich eine **Sammlung** einfach strukturierter Aussagen durch eine **Konjunktion** von Klauseln ausdrücken lässt.

Für jede aussagenlogische Formel ϕ gibt es eine Formel ψ_D in DNF und eine Formel ψ_K in KNF, so dass

$$\phi \equiv \psi_D \text{ und } \phi \equiv \psi_K.$$

Jede Formel ist äquivalent zu einer Formel in DNF und zu einer Formel in KNF.

Boolesche Funktionen und Formeln

Boolesche Funktionen

Eine Funktion $f : X \to Y$ heißt eine **boolesche Funktion**, wenn

(a) $Y = \{0, 1\}$ und

(b) wenn es eine natürliche Zahl $n \in \mathbb{N}$ mit $X = \{0, 1\}^n$ gibt.

Beispiele:

(a) Die boolesche Funktion

$$\mathrm{Und}_n : \{0, 1\}^n \to \{0, 1\}$$

nimmt genau dann den Wert 1 für Eingabe x an, wenn $x = 1^n$.

(b) Die Paritätsfunktion $\oplus_n$ ist die boolesche Funktion

$$\oplus_n : \{0, 1\}^n \to \{0, 1\}$$

mit der rekursiven Definition $\oplus_1(x_1) = x_1$ und

$$\oplus_{n+1}(x_1, \ldots, x_{n+1}) = \begin{cases} 0 & \oplus_n(x_1, \ldots, x_n) = x_{n+1} \\ 1 & \text{sonst.} \end{cases}$$

Das „**Paritätsbit**“ $p = \oplus_n(x_1, \ldots, x_n)$ ändert sich, wenn genau ein Bit „geflippt“ wird (**warum?**) und wird deshalb zur Fehlererkennung eingesetzt.

Boolesche Funktionen und Formeln

Boolesche Funktionen und Formeln der Aussagenlogik sind äquivalente Konzepte.

(a) Für eine Formel ϕ mit $|\mathrm{VAR}(\phi)| = n$ bilde die Wahrheitstafel und definiere die **boolesche Funktion $\mathbf{f}_\phi : \{0,1\}^n \to \{0,1\}$ für die Formel ϕ** durch

$$f_\phi(x) = \text{Wahrheitswert der Zeile von } x.$$

(b) Für eine boolesche Funktion $f : \{0,1\}^n \to \{0,1\}$
bilde die Wahrheitstafel, die in Zeile x den Wahrheitswert $f(x)$ besitzt.
Dann bestimme die DNF ϕ_f zur Wahrheitstafel:

$\phi_\mathbf{f}$ ist eine Formel für die boolesche Funktion f.

Der Entwurf von Schaltungen für boolesche Funktionen ist ein wichtiges Ziel der technischen Informatik.

Haben boolesche Funktionen immer kleine DNFs oder kleine KNFs?

Wir erinnern an die **Paritätsfunktion** $\oplus_n : \{0,1\}^n \to \{0,1\}$.
Dann hat

(a) jede DNF für $\oplus_n$ mindestens $\mathbf{2^{n-1}}$ Konjunktionsterme und

(b) jede KNF für $\oplus_n$ mindestens $\mathbf{2^{n-1}}$ Disjunktionsterme.

Beweis: Siehe Übungen.

Es gibt also boolesche Funktionen,
die sich weder durch DNFs noch durch KNFs kurz beschreiben lassen!

Wie groß sind denn Wahrheitstafeln?

Sei ϕ eine aussagenlogische Formel und sei

$$n := |\operatorname{Var}(\phi)|$$

die Anzahl der in ϕ vorkommenden Variablen. Dann gibt es

$$\mathbf{2^n}$$

verschiedene zu ϕ passende Belegungen $\mathcal{B}$.

(a) Eine zu ϕ passende Belegung $\mathcal{B}$ ist eine Funktion, die jede Variable in $\operatorname{Var}(\phi)$ auf einen Wahrheitswert abbildet.

(b) Wie viele Funktionen $\mathcal{B} : \operatorname{Var}(\phi) \to \{0, 1\}$ gibt es?

- Genau

$$|\operatorname{Abb}(\operatorname{Var}(\phi), \{0,1\})| = |\{0,1\}|^{|\operatorname{Var}(\phi)|} = 2^n$$

viele.

Die Größe von Wahrheitstafeln (2/2)

Die Wahrheitstafel einer Formel mit n Variablen hat genau 2^n Zeilen.

Ist das gut oder schlecht?

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 $13,7$ Milliarden Jahre, das sind ungefähr 10^{18} Sekunden, geschätzt.

Das aussagenlogische Erfüllbarkeitsproblem (KNF-SAT)

Eingabe:	Eine aussagenlogische Formel ϕ in KNF.
Frage:	Ist ϕ erfüllbar?

Wie schwierig ist das Erfüllbarkeitsproblem? (1/3)

- Natürlich kann man das Erfüllbarkeitsproblem mit einer Wahrheitstafel lösen.
  - Teste, ob es in der mit „ϕ“ beschrifteten Spalte mindestens eine 1 gibt.
  - Aber die Wahrheitstafel hat **exponentiell** viele Zeilen, nämlich 2^n Zeilen, wenn ϕ von n Variablen abhängt.
- Ein unter dem Stichwort **SAT-Solving** bekannter Teilbereich der Informatik versucht, wesentlich effizientere Verfahren zu entwickeln.
- Solche Verfahren können nicht für alle Formeln schnell sein, denn in der Veranstaltung „**Theoretische Informatik 1**“ (3. Semester) wird gezeigt:

Satz von Cook

KNF-SAT ist $\mathtt{NP}$-vollständig.

Wie schwierig ist das Erfüllbarkeitsproblem? (2/3)

- Eine präzise Definition des Begriffs „NP-vollständig“ wird in der **Theoretischen Informatik 1** gegeben.
  - Grob gesagt bedeutet die NP-Vollständigkeit von KNF-SAT, dass es wahrscheinlich kein **effizientes** Verfahren gibt, das KNF-SAT für **alle** Eingabeformeln löst. **:-((**
    - Ein Verfahren wird effizient gennant, wenn das Verfahren „relativ schnell“ Antworten selbst für große Eingaben findet.
    - Genauer: Die Laufzeit muss polynomiell in der Eingabelänge sein!
- Einige Heuristiken (die so genannten SAT-Solver) lösen KNF-SAT trotzdem für **viele** Eingabe-Formeln **erstaunlich effizient**.
  - Ich muss hoffen, dass ich für **meine** Eingabe-Formel ein Verfahren finde, dass meine Formel schnell genug löst.
  - Aber es gibt wahrscheinlich kein „einigermaßen“ schnelles Verfahren für **alle** Formeln!

Wie arbeiten denn diese SAT-Solver?

Wie schwierig ist das Erfüllbarkeitsproblem? (3/3)

Wenn die Disjunktionsterme $\ell \vee E_1$ und $\neg\ell \vee E_2$ wahr sind, dann ist auch

$$E_1 \vee E_2$$

wahr. Man sagt, dass $E_1 \vee E_2$ in einem **Resolutionsschritt** aus $\ell \vee E_1$ und $\neg\ell \vee E_2$ ableitbar ist.

Wir sollen feststellen, ob die KNF $\phi = D_1 \wedge D_2 \wedge \cdots \wedge D_m$ erfüllbar ist!

:-)) Wenn der leere Disjunktionsterm in einer Folge von Resolutionsschritten aus den Disjunktionstermen $D_1, \ldots, D_m$ ableitbar ist, dann ist ϕ unerfüllbar!

:-)) Man kann zeigen: Ist ϕ unerfüllbar, dann gibt es einen **Resolutionsbeweis**,

- also eine Folge von Resolutionsschritten ausgehend von den Disjunktionstermen $D_1, \ldots, D_m$,
- so dass am Ende der leere Disjunktionsterm produziert wird.

:-((Leider, leider gibt es aber KNFs, deren Unerfüllbarkeit nur mit exponentiell vielen Resolutionsschritten nachgewiesen werden kann!

- Nicht nur ist das Finden eines „Resolutionsbeweises" kompliziert,
- sondern mgl. gibt es nur sehr, sehr lange Beweise.

Resolution: Ein Beispiel

1. Die Kunden der Bahn sind nicht **z**ufrieden, wenn
   - sich die **P**reise erhöhen: $P \rightarrow \neg Z$, bzw. $\neg P \vee \neg Z$,
   - oder sich die **F**ahrzeiten verlängern: $F \rightarrow \neg Z$, bzw. $\neg F \vee \neg Z$.
2. Wenn der Frankfurter Kopfbahnhof nicht in einen Durchgangsbahnhof umge**b**aut wird, verlängern sich die Fahrzeiten: $\neg B \rightarrow F$, bzw. $B \vee F$.
3. Der Bahnhof kann nur dann umgebaut werden, wenn die Fahrpreise erhöht werden: $B \rightarrow P$, bzw. $\neg B \vee P$.

Die Bahn kann es niemandem recht machen, denn die KNF

$$(\neg P \vee \neg Z) \wedge (\neg F \vee \neg Z) \wedge (B \vee F) \wedge (\neg B \vee P) \wedge Z.$$

ist unerfüllbar.

Wie sieht ein Resolutionsbeweis aus?

Model Checking mit SAT-Solvern

(a) Wir haben ein System (z.B. eine Schaltung, ein Softwarepaket) entwickelt und haben einige Eigenschaften des Systems durch eine Formel $\phi \in$ AL modelliert.

(b) Wir müssen überprüfen, ob das System die Eigenschaft ψ erfüllt: D.h. gilt

$$\phi \models \psi?$$

Wenn ϕ als KNF und ψ als DNF vorliegt, dann überprüfe, ob die KNF

$$\phi \wedge \neg\psi$$

unerfüllbar ist! Und wie? Mit einem SAT-Solver.