Boolesche Terme und Boolesche Funktionen

Aussagen

Mit dem Begriff der *Aussage* und der logischen Verknüpfung von Aussagen beschäftigte man sich schon im alten Griechenland. Die Charakterisierung einer Aussage als "Satz (also als ein formalsprachliches Gebilde), der die Eigenschaft hat, entweder wahr oder falsch zu sein" geht auf Aristoteles zurück.
Die Aussagenlogik als klassisches Gebiet der mathematischen Logik beruht auf zwei Grundprinzipien, dem oben genannten *Zweiwertigkeitsprinzip*, welches fordert, dass jede Aussage einen eindeutig bestimmten Wahrheitswert hat, der nur *wahr* oder *falsch* sein kann, und dem *Extensionalitätsprinzip*, nach dem der Wahrheitswert einer zusammengesetzten Aussage nur von den Wahrheitswerten ihrer Bestandteile abhängt.
Wir werden im folgenden eine 1 für den Wahrheitswert *wahr* und eine 0 für *falsch* verwenden. Das Zusammensetzen von Aussagen erfolgt durch die Verwendung von Verknüpfungswörtern wie z.B. *und, oder, nicht, wenn ... dann.*

Beispiele:

1. Der Satz *"7 ist eine Primzahl."* und der Satz *"7 ist eine ungerade Zahl."* sind wahre Aussagen. Dagegen ist der Satz *"7 ist eine gerade Zahl."* eine falsche Aussage. Genauer gesehen ist der letzte Satz die Negation des zweiten Satzes, denn *nicht ungerade* zu sein, ist (zumindest für ganze Zahlen) das gleiche, wie *gerade* zu sein.

2. Der Satz *"7 ist eine Primzahl und 7 ist ungerade."* sowie der Satz *"7 ist eine Primzahl oder 7 ist gerade."* sind wahre Aussagen. Achtung: Auch der Satz *"7 ist eine Primzahl oder 7 ist ungerade."* ist eine wahre Aussage, denn das logische *oder* ist kein ausschließendes *entweder oder*. Dagegen ist der Satz *"7 ist eine Primzahl und 7 ist gerade."* eine falsche Aussage, denn der zweite Teil dieser Aussage ist falsch.

3. Der Satz "$\sqrt{2}$ *ist eine rationale Zahl."* ist – wie man aus der Schulmathematik weiß – eine falsche Aussage, aber es bedarf schon einiger Überlegungen, um das zu zeigen.

4. Der Satz *"Jede gerade natürliche Zahl größer als* 2 *kann als Summe von zwei Primzahlen dargestellt werden"* ist eine Aussage, denn entweder es gibt eine gerade Zahl, die sich nicht als Summe zweier Primzahlen darstellen lässt (dann ist die Aussaage falsch), oder es gibt keine solche Zahl (dann ist die Aussage wahr). Man nimmt an, dass die Aussage wahr ist - die sogenannte Goldbachsche Vermutung - konnte das aber bisher noch nicht beweisen.

5. Der Satz *"Dieser Satz ist falsch."* ist als Russels Paradoxon bekannt. Durch die spezielle Art des Bezugs auf sich selbst kann er weder wahr noch falsch sein und ist deshalb **keine** Aussage.

Wie wir in Punkt zwei gesehen haben, bestimmt sich der Wahrheitswert einer zusammengesetzten Aussage ausschließlich aus den Wahrheitswerten der Ausgangskomponenten. Deshalb ist es sinnvoll, Aussagevariablen einzuführen und die Wahrheitwerte von zusammengestezten Aussagen durch sogenannte Wahrheitsswerttabellen (kurz Wahrheitstafeln) zu beschreiben.

Rechnen mit Wahrheitswerten

Der englische Mathematiker George Boole (1815-1864) hat sich um die mathematische Formalisierung der Aussagenlogik besonders verdient gemacht. Deshalb werden eine Reihe von Begriffen, die wir im Folgenden kennenlernen werden mit seinem Namen verknüpft. In der Literatur findet man eine ganze Reihe von Bezeichnungen für die zwei Wahrheitswerte:

- *wahr* und $falsch$ (oder *true* und $false$),
- in abgekürzter Form w und f (oder t und f),
- besonders in der Informatik 1 für wahr und 0 für falsch,
- in der technischen Informatik L für wahr und O für falsch.

Wir legen uns auf die dritte Variante fest und definieren die Menge der Wahrheitswerte oder Booleschen Werte als $\mathbb{B} = \{0, 1\}$. Die Negation ist eine einstellige Operation auf $\mathbb{B}$, d.h. sie bezieht sich auf ein einzelnes Argument. Sie gibt für die 0 eine 1 und für die 1 eine 0 zurück. Eine Reihe von zweistelligen Operationen verknüpfen jeweils zwei Wahrheitswerte zu einen Wert. Man nennt diese Operationen *Konjunktion* (logisches und), *Disjunktion* (logisches oder), *Implikation*, *Äquivalenz* und *Antivalenz* (entweder oder).
Zur sauberen Unterscheidung zwischen diesen Operationen auf Werten, die zur Semantik der Aussagenlogik gehören, und der symbolischen Verknüpfung von Termen auf der syntaktischen Seite, werden wir zunächst verschiedene Schreibweisen einführen - später, wenn dieser Unterschied klar geworden ist, werden wir für beide Seiten jeweils die gleichen Verknüpfungsymbole verwenden. Die Negation ist eine Funktion $\texttt{neg} : \mathbb{B} \longrightarrow \mathbb{B}$, die durch $\texttt{neg}(0) = 1$ und $\texttt{neg}(1) = 0$ definiert ist. Die gennanten zweistelligen Operationen auf Wahrheitswerten sind Funktionen

$$\texttt{and, or, impl, equiv, antiv} : \mathbb{B} \times \mathbb{B} \longrightarrow \mathbb{B},$$

deren Definition üblicherweise in Tabelleform gegeben wird:

$a \in \mathbb{B}$	$b \in \mathbb{B}$	$\texttt{and}(a, b)$	$\texttt{or}(a, b)$	$\texttt{impl}(a, b)$	$\texttt{equiv}(a, b)$	$\texttt{antiv}(a, b)$
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	0	0	1
1	1	1	1	1	1	0

Aus der Tabelle kann man ablesen, dass die Konjunktion aus a und b genau dann wahr ist, wenn beide Eingabewerte wahr sind. Die Disjunktion aus a und b ist genau dann wahr, wenn mindestens einer der beide Eingabewerte ist und die Implikation aus a und b ist genau dann wahr, wenn a falsch oder b wahr ist. Versuchen Sie selbst, die Äquivalenz und die Antivalenz verbal zu beschreiben!

Boolesche Terme

Auf der syntaktischen Seite verwenden wir das Symbol $\neg$ für die Negation und die Symbole $\wedge, \vee, \rightarrow, \leftrightarrow, \oplus$ entsprechend für Konjunktion, Disjunktion, Implikation, Äquivalenz und Antivalenz. Man bezeichnet diese Symbole als logische Junktoren. Die Grundbausteine für den Aufbau sogenannter Terme oder Formeln werden durch eine Menge E elementarer Aussagen (mit festem Wahrheitswert!), zu der insbesondere eine wahre Aussage `true` und eine falsche Ausage `false` gehören, sowie eine abzählbare Menge Var von Variablen gebildet.

Definition: Die Menge der *Booleschen Terme (Formeln)* und der *Rang* von Booleschen Termen wird induktiv durch die folgenden Punkte definiert:

1. Jede elementare Aussage und jede Variable ist ein Boolescher Term. Diese Terme haben den Rang 0 und werden Primterme genannt.

2. Ist t ein Boolescher Term vom Rang n, dann ist $(\neg t)$ (gesprochen: *nicht* t) ein Boolescher Term vom Rang $n + 1$.

3. Sind t_1 und t_2 Boolesche Terme vom Rang n_1 und n_2, dann sind $(t_1 \wedge t_2)$ (gesprochen: t_1 *und* t_2) sowie $(t_1 \vee t_2)$ (gesprochen: t_1 *oder* t_2) Boolesche Terme vom Rang $max(n_1, n_2) + 1$.

4. Jeder Boolesche Term kann aus Primtermen durch eine endliche Folge von Anwendungen der Regeln 2) und 3) gebildet werden.

Anmerkungen zur Definition:

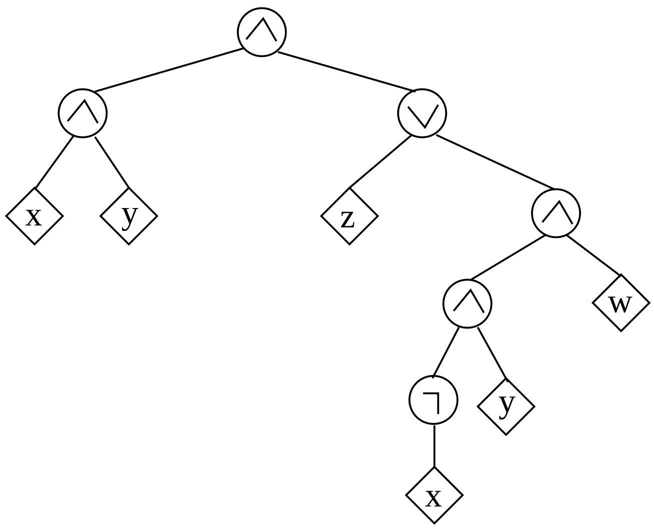
1) In der Literatur findet man häufig einen leicht modifizierten Ansatz, bei dem man auf die elementaren Aussagen verzichtet und nur Variable als Primterme zulässt. Der Unterschied ist marginal, wir werden später noch genauer darauf eingehen.

2) Etwas allgemeiner ist die Definition von Termen der Aussagenlogik, bei der im Punkt 3) zusätzlich die Terme $(t_1 \rightarrow t_2)$ (gesprochen: t_1 *impliziert* t_2 oder *aus* t_1 *folgt* t_2 oder auch *wenn* t_1, *dann* t_2) sowie $(t_1 \leftrightarrow t_2)$ (gesprochen: t_1 *ist äquivalent zu* t_2 oder t_1 *genau dann, wenn* t_2) und $(t_1 \oplus t_2)$ (gesprochen: *entweder* t_1 *oder* t_2) mit dem Rang $max(n_1, n_2) + 1$ eingeführt werden.

Eine anschauliche Erklärung für den Aufbau von Termen und ihrem Rang bekommt man durch den sogenannten Syntaxbaum des Terms. Der Syntaxbaum einer Primformel besteht nur aus einem einzelnen Knoten, der mit der entsprechenden Variable markiert wird. Der Syntaxbaum eines Terms, der durch Regel 2 als $t = (\neg t')$ entsteht, hat einen mit $\neg$ markierten Wurzelknoten, unter dem der Syntaxbaum von t' steht.

Für Terme die nach Regel 3 entstehen, wird der Wurzelknoten mit dem entsprechenden Junktor markiert und zu den Syntaxbäumen der zwei verwendeten Teilterme verbunden. Knoten, die mit Primformeln markiert sind, nennt man Blätter des Syntaxbaums. Der längste Weg vom Wurzelknoten zu einem Blatt beschreibt den Rang der Formel.

Beispiel: Der Term $((x \wedge y) \wedge (z \vee (((\neg x) \wedge y) \wedge w)))$ hat den folgenden Syntaxbaum:



Durch die Klammersetzung kann man genau nachvollziehen, wie ein Term entstanden ist, aber oft ist diese Darstellung sehr unübersichtlich. Deshalb werden zur Vereinfachung der Notation die folgenden Regeln vereinbart:

1. Wenn ein Term für sich steht (d.h. nicht als Baustein eines größeren Terms verwendet ist), kann man auf sein äußeres Klammerpaar verzichten.

2. Die Reihenfolge $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$ repräsentiert die fallende Bindungsstärke und damit die ansteigende Trennungsstärke der Junktoren. Wird ein Term t durch einen Junktor $*$ als $t = (t_1 * t_2)$ definiert und bindet der Junktor von t_1 bzw. von t_2 stärker als $*$, dann kann man die Klammern um t_1 bzw. um t_2 weglassen.

3. Bei einer Kette gleicher Junktoren, die von links nach rechts geklammert sind, kann man die Klammern weglassen, oder anders gesagt gilt die Regel, dass bei einer Kette gleicher Junktoren ohne Klammerung der zu Grunde liegende Term als linksassoziativ geklammert verstanden wird.

Beispiel: Durch Anwendung dieser Regeln kann man für dem im obigen Beispiel dargestellten Term fast alle Klammern weglassen. Nur für die Konjunktion hinter z ist die Klammerung notwendig:

$$t = ((x \wedge y) \wedge (z \vee (((\neg x) \wedge y) \wedge w))) = x \wedge y \wedge (z \vee \neg x \wedge y \wedge w)$$

Legt man für alle in einem Term auftretenden Variablen Wahrheitswerte fest, so induziert diese sogenannte Belegung auch einen Wahrheitswert für den Term. Man nennt diesen induktiven Prozess auch Auswertung (oder Interpretation) des Terms unter der Belegung. Im folgenden geben wir eine formale Beschreibung dieser Idee.

Definition: Eine Belegung einer Variablenmenge V ist eine Funktion $\omega : V \longrightarrow \mathbb{B}$, die jeder Variable einen Wahrheitswert zuweist.
Die Interpretation $I_\omega(t)$ eines Booleschen Terms t über eine Variablenmenge V unter einer Belegung $\omega : V \longrightarrow \mathbb{B}$ ist (analog zur Termdefinition) rekursiv definiert, wobei der Rang als Rekursionsparameter verwendet wird:

- Verankerung für Terme vom Rang 0: Für $t \in E$ ist $I_\omega(t)$ der Wahrheitswert dieser Aussage und für $t \in V$ ist $I_\omega(t) = \omega(t)$.
- Rekursionsschritt: Sei die Interpretation für alle Terme vom Rank n bereits bekannt und t ein Term vom Rang $n+1$:
  - Für $t = \neg t'$ ist $I_\omega(t) = \texttt{neg}(I_\omega(t'))$;
  - Für $t = t_1 \wedge t_2$ ist $I_\omega(t) = \texttt{and}\ (I_\omega(t_1), I_\omega(t_2))$;
  - Für $t = t_1 \vee t_2$ ist $I_\omega(t) = \texttt{or}\ (I_\omega(t_1), I_\omega(t_2))$

Zur Auswertung von aussagenlogischen Termen fügt man noch die folgenden Ergänzungsregeln hinzu:

- Für $t = t_1 \rightarrow t_2$ ist $I_\omega(t) = \texttt{impl}\ (I_\omega(t_1), I_\omega(t_2))$;
- Für $t = t_1 \leftrightarrow t_2$ ist $I_\omega(t) = \texttt{equiv}\ (I_\omega(t_1), I_\omega(t_2))$
- Für $t = t_1 \oplus t_2$ ist $I_\omega(t) = \texttt{antiv}\ (I_\omega(t_1), I_\omega(t_2))$

Die Ergebnisse der Auswertungen einer Formel unter allen möglichen Belegungen werden in einer Wahrheitstafel zusammengefasst.

Nachdem nun die Unterschiede zwischen Junktoren auf der symbolischen Seite (links vom Gleichheitszeichen) und Operationen auf Wahrheitswerten (rechts vom Gleichheitszeichen) deutlich geworden sind, erlaubt man jetzt auch die Verwendung der Junktoren als Operationssymbole für Wahrheitswerte. Welche der beiden Deutungen im konkreten Fall richtig ist, ergibt sich aus dem jeweiligen Kontext.

Definition: Zwei Terme t_1 und t_2 sind logisch äquivalent, wenn jede beliebige Belegung der Variablen für beide Formeln den gleichen Wahrheitswert induziert. Wir schreiben dafür $t_1 \equiv t_2$

Wie das folgende Beispiel zeigt, kann die Äquivalenz von zwei Formeln prinzipiell durch Wahrheitstafeln überprüft werden: Um festzustellen, ob die Formeln

$$t_1 = \neg(x_1 \vee ((x_1 \vee x_2) \wedge x_2) \quad \text{und} \quad t_2 = \neg x_1 \wedge \neg x_2$$

logisch äquivalent sind, wertet man beide Terme in einer Tabelle aus und stellt fest, ob die zu t_1 und t_2 gehörenden Spalten identisch sind. Um Platz zu sparen, führen wir für die Zwischenterme die Namen $s = (x_1 \vee x_2) \wedge x_2$ sowie $s' = x_1 \vee s$ ein.

x_1	x_2	$x_1 \vee x_2$	s	s'	$t_1 = \neg s'$	$\neg x_1$	$\neg x_2$	$t_2 = \neg x_1 \wedge \neg x_2$
0	0	0	0	0	1	1	1	1
0	1	1	1	1	0	1	0	0
1	0	1	0	1	0	0	1	0
1	1	1	1	1	0	0	0	0

Es gibt aber auch eine alternative Methode zum Beweis solcher Äquivalenzen, nämlich durch die schrittweise Anwendung einiger Grundregeln. Diese Vorgehensweise ist eng verwandt zu Beweisen von arithmetischen Identitäten wie z.B. der bekannten binomischen Formel $(a+b)^2 = a^2+2ab+b^2$ durch Anwendung von Distributiv–, Kommutativ– und Assoziativgesetz. Für das Rechnen mit Booleschen Werten gelten nicht nur diese drei Gesetze, sondern es eine ganze Reihe weiterer Regeln.

Satz: Für beliebige Variable x, y, z gelten die folgenden Term–Äquivalenzen:

$$
\begin{array}{ccccl}
(x \wedge y) \wedge z & \equiv & x \wedge (y \wedge z) & & \text{(Assoziativgesetz)} \\
(x \vee y) \vee z & \equiv & x \vee (y \vee z) & & \\
x \wedge y & \equiv & y \wedge x & & \text{(Kommutativgesetz)} \\
x \vee y & \equiv & y \vee x & & \\
x \wedge (y \vee z) & \equiv & (x \wedge y) \vee (x \wedge z) & & \text{(Distributivgesetz)} \\
x \vee (y \wedge z) & \equiv & (x \vee y) \wedge (x \vee z) & & \\
\neg(x \wedge y) & \equiv & \neg x \vee \neg y & & \text{(deMorgansche Regel)} \\
\neg(x \vee y) & \equiv & \neg x \wedge \neg y & & \\
\neg\neg x & \equiv & x & & \text{(Involutionsgesetz)} \\
x \wedge x \equiv x & \text{und} & x \vee x \equiv x & & \text{(Idempotenz)} \\
x \wedge \texttt{true} \equiv x & \text{und} & x \vee \texttt{false} \equiv x & & \text{(Neutralität)} \\
x \wedge \neg x \equiv \texttt{false} & \text{und} & x \vee \neg x \equiv \texttt{true} & & \text{(Komplementäreigenschaft)} \\
x \wedge \texttt{false} \equiv \texttt{false} & \text{und} & x \vee \texttt{true} \equiv \texttt{true} & & \text{(Dominanzeigenschaft)} \\
x \wedge (x \vee y) \equiv x & \text{und} & x \vee (x \wedge y) \equiv x & & \text{(Absorption)}
\end{array}
$$

Diese Gesetze können leicht mit Wahrheitstafeln bewiesen werden, genau wie die zwei folgenden Regeln, die eine Beschreibung liefern, wie man Implikation bzw. Äquivalenz als Boolesche Formel darstellen kann:

$$x \rightarrow y \equiv \neg x \vee y \qquad x \leftrightarrow y \equiv x \wedge y \vee \neg x \wedge \neg y \equiv (\neg x \vee y) \wedge (x \vee \neg y)$$

Bei der Anwendung dieser Regeln möchte man sich aber nicht nur auf die oben aufgeführten, aus zwei oder drei Variablen aufgebauten Terme beschränken, sondern jede dieser Variablen auch durch einen beliebigen anderen Term ersetzen können. Die Grundlage dafür wird durch das sogenannte Substitutionstheorem gesichert.

Definition: Mit $t[t'/x]$ bezeichnet man den aus t enstehenden Term, wenn jedes Auftreten der Variable x in t durch den Term t' ersetzt wird.

Beispiel: Für den Term $t = x \wedge (y \vee x)$ ergibt sich durch Ersetzung von x durch den Term $t' = x \wedge z$ der folgende Term:

$$t[t'/x] = t[(x \wedge z)/x] = (x \wedge z) \wedge (y \vee (x \wedge z))$$

Eine besonders anschauliche Deutung dieser Definition erhält man durch Betrachtung der Syntaxbäume: Der Syntaxbaum des Terms $t[t'/x]$ entsteht aus dem Syntaxbaum von t, wenn jedes mit x markierte Blatt durch den Syntaxbaum von t' ersetzt wird.

Substitutionstheorem: Seien t_1 und t_2 zwei semantisch äquivalente Terme und x eine Variable. Dann gelten für jeden Term t die folgenden Äquivalenzen:

$$t[t_1/x] \equiv t[t_2/x] \qquad \text{und} \qquad t_1[t/x] \equiv t_2[t/x].$$

Wir führen hier keinen formalen Beweis und merken lediglich an, dass die zu einem Beweis notwendige Fallunterscheidung durch Wahrheitstafeln realisiert werden kann.

Beispiel: Der Beweis der Äquivalenz $x \vee ((y \vee z) \wedge \neg(\neg x \wedge (\neg x \vee u))) \equiv x$ mit einer Wahrheitstafel würde 16 Zeilen erfordern. Verwendet man dagegen den Satz kombiniert mit der Absorptionsregel, dem Involutionsgesetz und noch einmal mit der Absorptionsregel, so erhält man einen einfachen und kurzen Beweis:

$$x \vee ((y \vee z) \wedge \neg(\neg x \wedge (\neg x \vee u))) \equiv x \vee ((y \vee z) \wedge \neg\neg x) \equiv x \vee ((y \vee z) \wedge x) \equiv x$$

Erfüllbarkeit, Tautologie und Kontradiktion

Definition: Eine Term t wird *erfüllbar* genannt, wenn es eine Belegung ω der Variablen von t gibt, so dass die Auswertung von t unter ω den Wert 1 ergibt, d.h. $I_\omega(t) = 1$. Man nennt t *allgemeingültig* oder eine *Tautologie*, wenn t für jede Belegung den Wert 1 annimmt und *unerfüllbar* oder eine *Kontradiktion*, wenn wenn t für jede Belegung den Wert 0 annimmt.

Da t genau dann eine Tautologie ist, wenn $\neg t$ eine Kontradiktion (also nicht erfüllbar) ist, kann man das allgemeine algorithmische Problem des Testens, ob ein Term eine Tautologie ist, auf das Erfüllbarkeitsproblem zurückführen und umgekehrt.

Diese Aufgabenstellungen haben konzeptionell eine sehr einfache Lösung, nämlich das Auswerten unter allen möglichen Belegungen. Leider wächst aber die Anzahl der Belegungen exponentiell mit der Anzahl der Variablen und deshalb führt dieser Ansatz zu keinem effizienten Algorithmus. Mehr noch, diese Probleme erweisen sich als NP–schwer. Dieser Begriff beschreibt eine ganze Problemklasse für die man keine effizienten Algorithmen kennt und für die allgemein vermutet wird, dass es keine effizienten Algorithmen zu ihrer Lösung gibt.

Boolesche Funktionen

Definition: Eine n–stellige Boolesche Funktion ist eine Abbildung $f : \mathbb{B}^n \longrightarrow \mathbb{B}$, wobei $\mathbb{B}^n = \underbrace{\mathbb{B} \times \ldots \times \mathbb{B}}_{n \text{ mal}}$ die Menge aller n–Tupel von Booleschen Werten bezeichnet.

Damit beschriebt eine n–stellige Boolesche Funktion das Eingabe–Ausgabe–Verhalten einer Schaltung mit n 0/1–Eingängen und einem 0/1–Ausgang. In der Regel verwendet man Wertetabellen zur Darstellung von Booleschen Funktionen.

Betrachtet man Boolesche Terme über der Variablenmenge $V = \{x_1, \ldots, x_n\}$, dann kann man jede Belegung $\omega : V \longrightarrow \mathbb{B}$ eindeutig durch das n–Tupel $(b_1, \ldots, b_n) = (\omega(x_1), \ldots, \omega(x_n))$ repräsentieren und umgekehrt beschreibt jedes n–Tupel aus $\mathbb{B}^n$ eine Belegung von V.

Wir nutzen diesen Zusammenhang, um jedem Term t über V eine n–stellige Boolesche Funktion f_t zuzuordnen, indem man den Term über allen Belegungen auswertet, d.h. $f_t(b_1, \ldots, b_n) = I_\omega(t)$, wobei ω die durch $(b_1, \ldots, b_n)$ repräsentierte Belegung ist. Wir haben damit ein weiteres Beispiel eines Informationsystems, in dem die Booleschen Terme als Repräsentationen der Information auftreten und die durch sie dargestellte abstrakte Information jeweils eine Boolesche Funktion ist.

Die logische Äquivalenz von zwei Termen beschreibt hier gleichzeitig die semantische Äquivalenz von zwei Repräsentationen. Offensichtlich hat jede Boolesche Funktion f, die durch einen Booleschen Term t dargestellt werden kann, unendlich viele Repräsentationen, nämlich $t \vee (x \wedge \neg x)$, $t \vee (x \wedge \neg x) \vee (x \wedge \neg x)$, usw.
Die Frage, ob jede Boolesche Funktion durch einen Booleschen Term repräsentiert werden kann, ist dagegen noch ungeklärt - wir werden in der nächsten Vorlesung den Nachweis führen, dass man sie positiv beantworten kann.

Bereits mit wenigstelligen Booleschen Funktionen kann man relativ komplexe Zusammenhänge ausdrücken. Ein deutliches Indiz dafür ist die doppelt exponentiell wachsende Anzahl der n–stelligen Booleschen Funktionen. Die Bestimmung dieser Anzahl ist relativ einfach:
Allgemein beschreibt die Potenz k^m die Anzahl der Funktionen, die einen bestimmten m–elementigen Definitionsbereich A in einen k–elementigen Wertebereich B abbilden, denn man hat für jedes der m Elemente aus A genau k Möglichkeiten der Zuordnung ihres Bildes, also insgesamt $\underbrace{k \cdot k \cdot \ldots \cdot k}_{m \text{ mal}} = k^m$ Möglichkeiten.
Im konkreten Fall ist $A = \mathbb{B}^n$ eine 2^n–elementige Menge und $B = \mathbb{B}$ ist 2–elementig. Folglich gibt es $2^{(2^n)}$ n–stellige Boolesche Funktionen. Das enorme Wachstum dieser Größe wird schon an einem kleinen Beispiel deutlich:
Während für $n = 3$ die Anzahl der 3–stelligen Booleschen Funktionen mit $2^8 = 256$ noch recht übersichtlich ist, ergibt sich für $n = 5$ (und $2^n = 2^5 = 32$), dass die Anzahl der 5–stelligen Booleschen mit 2^{32} bereits über 4 Milliarden liegt.