Prof. Dr. H. Brenner **Osnabrück SS 2018**

Einführung in die mathematische Logik

Vorlesung 2

Sprache als Symbolketten

Wir knüpfen an die Überlegungen der ersten Vorlesung an, ob es eine Maschine (einen Computer, einen Algorithmus) gibt, der (alle korrekten) mathematische Aussagen ausdrücken, ausdrucken, überprüfen, beweisen oder widerlegen kann. Eine solche Maschine operiert mit Zeichenreihen, die wir in diesem Zusammenhang Wörter einer (formalen) Sprache nennen. Wir beschreiben daher den Aufbau einer formalen Sprache.

Definition 2.1. Es sei A eine Menge von Symbolen. Dann nennt man jede endliche Zeichenreihe, die man mit den Elementen aus A aufstellen kann, ein *Wort über dem Alphabet A*.

Die Menge aller Wörter über dem Alphabet A bezeichnen wir mit A^*.

Das zugrunde liegende Alphabet kann endlich oder unendlich sein, für praktische Anwendungen reicht ein endliches Alphabet. Die Elemente des Alphabets nennt man Buchstaben, Zeichen oder Symbole. Mit einer Zeichenreihe meint man eine hintereinander geschriebene Buchstabenkette (oder Symbolkette). Dazu gehören die einelementigen Ketten, also die Elemente aus A selbst, aber auch die leere Kette (das leere Wort), die wir mit $\emptyset$ bezeichnen. Bei dieser Definition kommt es nicht auf irgendeine Sinnhaftigkeit der Wörter an, es handelt sich um eine rein formale Definition.

Beispiel 2.2. Es sei ein einelementiges Alphabet $A = \{|\}$ gegeben. Dann besitzt jedes Wort über A die Gestalt

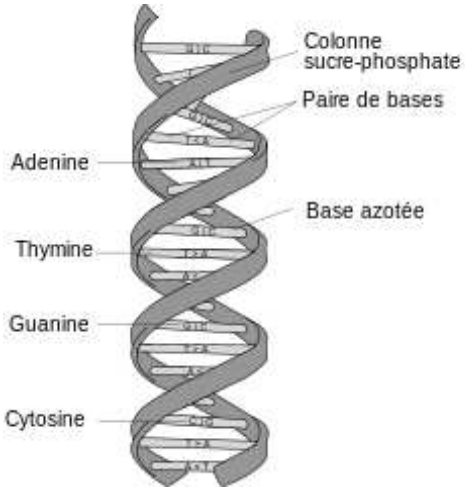
$$|||\ldots|$$

mit einer gewissen Anzahl von Strichen. Zwei solche Wörter sind genau dann gleich, wenn ihre Strichanzahl übereinstimmt. In diesem Fall entsprechen also die Wörter den natürlichen Zahlen (das leere Wort entspricht der 0).

Der Binärcode besteht aus den beiden Symbolen 0 und 1, der Morsecode besteht aus drei Zeichen: kurzes Signal, langes Signal, Pause. Grundsätzlich führt jedes nichtleere endliche Alphabet zu einer abzählbar-unendlichen Menge an Wörtern und hat somit prinzipiell die gleiche Ausdrucksstärke.

Beispiel 2.3. Die DNA-Stränge, die die Erbinformationen aller Lebewesen tragen, sind Doppelketten in Helixform aus Nukleotiden. Die entscheidenden Bestandteile der Nukleotiden sind die Basen, wofür es nur vier Möglichkeiten gibt, nämlich Adenin (A), Thymin (T), Guanin (G) und Cytosin (C). Die

Nukleotiden treten in der Helix stets mit einem festen Partner (nämlich Adenin mit Thymin und Guanin mit Cytosin) auf, so dass die Struktur durch die eine Hälfte der Helix festgelegt ist. Daher entspricht (bis auf die Leserichtung und die Strangauswahl) die genetische Information eines DNA-Stranges einem Wort über dem Alphabet mit den Buchstaben A,T,G,C.



Wenn zu einem Alphabet A ein neues Zeichen $-$ als „Leerzeichen“ hinzugenommen wird, so werden manchmal die Wörter aus A als (eigentliche) Wörter und die Wörter aus dem Alphabet $A \cup \{-\}$ als Texte (oder Sätze) bezeichnet. Mit der Hinzunahme eines weiteren Satzbeendungssymbols kann man auch zwischen Sätzen und Texten unterscheiden.

Die geschriebene natürliche Sprache umfasst das Alphabet, das aus den Großbuchstaben A,B,C,...,Z,Ä,Ö,Ü, den Kleinbuchstaben a,b,c,...,z,ä,ö,ü,ß, den Ziffern, den Satzzeichen und einem Leerzeichen für den Abstand zwischen den Wörtern besteht. Jede lineare Hintereinanderreihung dieser Zeichen gilt für uns als Text. Im Moment interessieren wir uns nicht dafür, ob die geschriebenen Texte syntaktisch richtig gebildet oder semantisch sinnvoll sind. Im Moment ist also z.B.

$$!!fL33kAs.,r$$

ein erlaubter Text.

Rekursive Definitionen

In der Definition von einem Wort über einem Alphabet haben wir von einer Menge gesprochen und somit eine naive Mengenlehre vorausgesetzt. Im endlichen Fall wird die Symbolmenge einfach durch Auflisten ihrer Elemente gegeben. Für die gebildeten Wörter haben wir implizit verwendet, dass das Bilden von linearen Zeichenreihen unproblematisch ist.

Ein wichtiges Prinzip, Mengen zu definieren, ist das der *rekursiven Definition*. Eine rekursive Definition besteht aus zwei Sorten von Regeln. (1) Einerseits gewisse Startregeln, die sagen, was direkt zu der Menge gehört, und (2) Rekursionsregeln (Generierungsregeln), die die Form einer Bedingung haben, und besagen, dass wenn gewisse Objekte zu der Menge gehören, und wenn neue Objekte aus diesen Objekten in bestimmter Weise gebildet sind, dass dann diese neuen Objekte ebenfalls dazu gehören (die dritte stillschweigende Bedingung an eine rekursive Definition ist, dass es keine weitere Möglichkeit gibt, zu der Menge zu gehören, außer den in (1) und (2) genannten).

Beispiel 2.4. Die Menge der Wörter über einem Alphabet A kann man auch folgendermaßen rekursiv definieren.

(1) $\emptyset$ ist ein Wort über A.
(2) Wenn x ein Wort ist und $a \in A$ ein Buchstabe, so ist auch xa ein Wort.

Hier repräsentiert x (eine Variable) ein beliebiges schon konstruiertes Wort. Dabei ist $\emptyset a$ als a zu lesen, so dass die beiden erlaubten Konstruktionsschritte (also der Anfangsschritt und der Rekursionsschritt) sichern, dass die einzelnen Symbole aus A Wörter sind. Wenn das Alphabet durch $A = \{a, b, c\}$ gegeben ist, so würde der rekursive Nachweis, dass *abbac* ein Wort ist, folgendermaßen gehen.

(1) Wegen der Anfangsbedingung ist $\emptyset$ ein Wort.
(2) Deshalb und wegen des Rekursionsschrittes ist $\emptyset a = a$ ein Wort.
(3) Deshalb und wegen des Rekursionsschrittes ist ab ein Wort (hier ist also $x = a$ das schon nachgewiesene Wort und der Buchstabe b wird angehängt).
(4) Deshalb und wegen des Rekursionsschrittes ist abb ein Wort (hier ist also $x = ab$ das schon nachgewiesene Wort und der Buchstabe b wird angehängt).
(5) Deshalb und wegen des Rekursionsschrittes ist $abba$ ein Wort (hier ist also $x = abb$ das schon nachgewiesene Wort und der Buchstabe a wird angehängt).
(6) Deshalb und wegen des Rekursionsschrittes ist $abbac$ ein Wort (hier ist also $x = abba$ das schon nachgewiesene Wort und der Buchstabe c wird angehängt).

Natürlich kann man *abbac* sofort ansehen, dass es sich um eine linear angeordnete Zeichenreihe über $\{a, b, c\}$ handelt, und der rekursive Nachweis scheint übertrieben pedantisch zu sein. Bei komplexer gebildeten Mengen ist aber die rekursive Definition unerlässlich, vor allem auch deshalb, da sie ermöglicht, Eigenschaften der Elemente einer Menge über den rekursiven Aufbau nachzuweisen.

Bemerkung 2.5. Es sei M eine rekursiv definierte Menge, die durch eine Startmenge $S \subseteq M$ und gewisse Rekursionsvorschriften gegeben sei. Nehmen wir an, wir möchten für alle Elemente der Menge M eine gewisse Eigenschaft E nachweisen. Das *Beweisprinzip durch Rekursion*[1] oder *Beweisprinzip über den rekursiven Aufbau der Menge* funktioniert folgendermaßen.

(1) Man zeigt, dass jedes Element aus der Startmenge die Eigenschaft E erfüllt (Rekursionsanfang).
(2) Man zeigt für jede Rekursionsvorschrift, dass unter der Voraussetzung, dass die in dieser Vorschrift verwendeten Elemente die Eigenschaft E besitzen, dann auch das durch die Vorschrift produzierte Element die Eigenschaft besitzt (Rekursionsschritt).

Daraus kann man dann schließen, dass jedes Element aus M die Eigenschaft erfüllt. Die Richtigkeit dieses Beweisprinzips beruht auf folgender Betrachtung: Es sei $N \subseteq M$ die Menge aller Elemente, für die die Eigenschaft E gilt. Aufgrund des Rekursionsanfangs gilt $S \subseteq N$. Aufgrund des Rekursionsschrittes ist N abgeschlossenen unter sämtlichen Rekursionsvorschriften. Dies ist aber die Definition für M, also ist $N = M$ und die Eigenschaft gilt für ganz M.

Ein Spezialfall dieses Beweisprinzips ist das Prinzip der vollständigen Induktion für natürliche Zahlen. Die natürlichen Zahlen sind rekursiv durch das Startelement 0 und eine einzige Rekursionsregel, nämlich die Nachfolgerregel festgelegt: Wenn n eine natürliche Zahl ist, so ist auch der Nachfolger $n' = n + 1$ von n eine natürliche Zahl.

Bemerkung 2.6. Es sei M eine rekursiv definierte Menge mit der Startmenge $S \subseteq M$. Verwandt mit dem Beweisprinzip über den rekursiven Aufbau der Menge ist das Prinzip, eine Abbildung φ von M in eine weitere Menge N rekursiv zu definieren. Dazu geht man folgendermaßen vor.

(1) Man legt eine Abbildung
$$\varphi_0 \colon S \longrightarrow N$$
fest.
(2) Für jede Rekursionsvorschrift erklärt man, wie man aus den schon festgelegten Werten $\varphi(m_1), \ldots, \varphi(m_k)$ der Elemente $m_1, \ldots, m_k$, auf die die Vorschrift Bezug nimmt, den Wert unter φ für das durch die Vorschrift produzierte Element m festlegt.
(3) Man muss sicherstellen, dass, falls es für ein Element mehrere Möglichkeiten gibt, dieses rekursiv zu erzeugen, die verschiedenen Möglichkeiten zum gleichen Wert führen.

Wenn diese Bedingungen erfüllt sind, ist eine wohldefinierte Abbildung
$$\varphi \colon M \longrightarrow N$$

[1] Oft spricht man einfach von einem Beweis durch Induktion.

definiert.

Eine Sprache besteht aus sinnvollen Wörtern und sinnvollen Sätzen, nicht aus der beliebigen Aneinanderreihung von Symbolen oder Buchstaben (oder Lauten). Es ist aber vorteilhaft, erstmal alle Möglichkeiten zuzulassen und daraus durch eine Vorgabe von Regeln die sinnvollen Ausdrücke, Wörter, Lautkombinationen herauszufiltern. So funktioniert auch der kleinkindliche Spracherwerb und der Aufbau der formalen Sprachen. Wir werden nun den rekursiven Aufbau von syntaktisch korrekten Aussagen besprechen.

Aussagenlogik

Die mathematische Logik beschäftigt sich hauptsächlich mit Prädikatenlogik, da in dieser ein Großteil der Mathematik beschrieben werden kann. Als Vorstufe dazu behandeln wir jetzt die Aussagenlogik. Wie bei der Prädikatenlogik später folgen wir dem Schema

Formale Sprache - Interpretationen und semantische Tautologien - Syntaktische Tautologien und Ableitungskalkül - Vollständigkeit.

Die Sprache der Aussagenlogik

Die formallogische Sprache der Aussagenlogik wird ausgehend von einer Variablenmenge V und einer einfachen Menge an Junktoren rekursiv aufgebaut. Die Aussagenvariablen werden wir zumeist mit $p, q, r, p_1, \ldots, p_k, p_n$ $(n \in \mathbb{N})$ etc. bezeichnen. Sie repräsentieren Aussagen, haben aber keinen eigenen Inhalt, sondern teilen mit Aussagen lediglich gewisse syntaktische Eigenschaften (semantische Eigenschaften werden hier noch nicht besprochen). Beispiele für solche syntaktischen Eigenschaften sind, dass man zu einer Aussage eine Negation bilden kann, oder dass man zwei Aussagen durch „und“ verknüpfen kann. Die Aussagenvariablen repräsentieren Grundaussagen, die durch solche Verknüpfungen zu komplexeren Aussagen zusammengesetzt werden können, die selbst wiederum zu weiter verschachtelten Aussagen verbunden werden können. Die folgende Definition fixiert die formale Sprache der Aussagenlogik; es handelt sich um eine rekursive Definition, wobei die Aussagenvariablen die Startelemente sind und die logischen Operationen als Generierungsregeln auftreten. Das dieser rekursiven Definition zugrunde liegende Alphabet besteht neben einer Menge V an Aussagenvariablen aus den Symbolen

$$\neg,\ \wedge,\ \vee,\ \rightarrow,\ \leftrightarrow,\ (,\ )\,,$$

die als

nicht, und, oder, impliziert, genau dann, wenn, Klammer auf, Klammer zu

gelesen werden; die zugehörigen Substantive sind *Negation*, *Konjunktion*, *Disjunktion*, *Implikation* und *Äquivalenz*. Die Bezeichnungen orientieren sich natürlich an den später einzuführenden Bedeutungen, im Moment sind es lediglich Wörter für bestimmte Symbole. Die Sprache der Aussagenlogik wird als Teilmenge von A^* realisiert, wobei $A = V \cup \{\neg, \wedge, \vee, \to, \leftrightarrow, (,)\}$ ist.

Definition 2.7. Es sei V eine Menge (deren Elemente wir als *Aussagenvariable* bezeichnen). Dann wird die zugehörige *Sprache der Aussagenlogik* L^V (zu V) rekursiv durch folgende Regeln definiert.

(1) Jedes $p \in V$ gehört zu L^V.
(2) Wenn $\alpha \in L^V$, so ist auch $\neg(\alpha) \in L^V$.
(3) Wenn $\alpha, \beta \in L^V$, so sind auch $(\alpha) \wedge (\beta)$, $(\alpha) \vee (\beta)$, $(\alpha) \to (\beta)$, $(\alpha) \leftrightarrow (\beta) \in L^V$.

Häufig verwendet man weniger Symbole, beispielsweise verzichtet man auf $\to, \leftrightarrow$. Die Klammerungen werden oft auch anders gesetzt. Z.B. erlaubt man manchmal $\neg\alpha$ (ohne Klammer) oder man macht die Klammern außen, also $(\alpha \wedge \beta)$. Sehr oft lässt man Klammern, um die Lesbarkeit der Aussagen zu erhöhen, einfach weg, obwohl dies vom syntaktischen Standpunkt aus ein schweres Vergehen ist.

Beispiel 2.8. Es seien p, q, r Aussagenvariablen. Dann sind beispielsweise

$$p,\ \neg(p),\ \neg(\neg(\neg(p))),\ (p) \wedge (\neg(q)),\ ((p) \wedge (\neg(q))) \wedge (\neg(r)),$$
$$(((\neg(\neg(p))) \to (\neg(q))) \vee ((\neg(r))) \leftrightarrow (\neg(r)) \wedge (q))$$

korrekt gebildete Aussagen, d.h. sie gehören zu L^V. Dagegen sind

$$\neg,\ \to,\ p\wedge,\ p \wedge q,\ (p) \wedge (\neg q,\ (p) \wedge (q) \wedge (r),$$

keine Aussagen in L^V (aber natürlich Wörter über dem gegebenen Alphabet).

Der Nachweis, dass ein gegebenes Wort eine korrekt gebildete Aussage ist, erfolgt über eine Ableitungskette oder einen Aussagestammbaum. Bei einer *Ableitungskette* listet man Zeile für Zeile korrekt gebildete Aussagen auf, wobei diese Aussagen entweder Aussagenvariablen oder aber mittels einer Rekursionsregel aus vorhergehenden Aussagen entstanden sind. Die letzte Zeile enthält die Aussage, deren Korrektheit man zeigen will.

Beispiel 2.9. Eine Ableitungskette für

$$((p) \wedge (r)) \to ((\neg(q)) \vee (r))$$

sieht folgendermaßen aus.

(1) p (Aussagenvariable),
(2) q (Aussagenvariable),
(3) r (Aussagenvariable),
(4) $\neg(q)$ (Negation auf 2),
(5) $(p) \wedge (r)$ (Konjunktion auf 1 und 3),

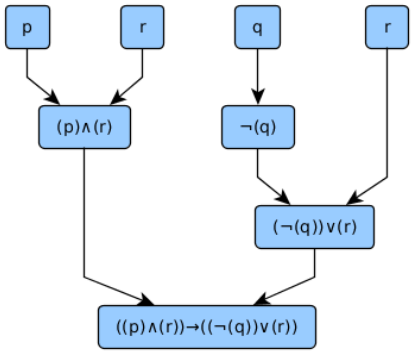
(6) $(\neg(q)) \vee (r)$ (Disjunktion auf 3 und 4),
(7) $((p) \wedge (r)) \rightarrow ((\neg(q)) \vee (r))$ (Implikation auf 5 und 6).

Ein Aussagenstammbaum ist eine graphisch übersichtliche Version einer Ableitungskette. Er beginnt mit den verwendeten Aussagenvariablen als Blättern und erzeugt dann Schritt für Schritt durch Vereinigungen von Zweigen die beteiligten Teilaussagen, bis schließlich die in Frage stehende Aussage als Stamm erzeugt ist.

Beispiel 2.10. Wir wollen uns anhand eines Stammbaumes klar machen, dass die Zeichenkette

$$((p) \wedge (r)) \rightarrow ((\neg(q)) \vee (r))$$

eine Aussage ist, also gemäß den Regeln korrekt gebildet ist. Der Abstammungsbaum entsteht ausgehend von den Blättern, die die vorkommenden Aussagenvariablen (mit ihrer Häufigkeit) repräsentieren, indem man Schritt für Schritt komplexere Teilaussagen zusammensetzt.



Jede Aussage hat eine eindeutige „rekursive Geschichte“, d.h. es gibt für jede Aussage nur eine Abfolge von Rekursionsschritten, um sie aus Aussagenvariablen aufzubauen, siehe Aufgabe 2.19.

Aussagenlogische Interpretationen

Wir kommen zur Interpretation einer aussagenlogischen Sprache und insbesondere der darin auftretenden Junktoren. Der Ansatz ist, dass eine Aussagenvariable nur wahr oder falsch sein kann.

Definition 2.11. Es sei V eine Menge von Variablen und L^V die zugehörige aussagenlogische Sprache. Unter einer *Wahrheitsbelegung* versteht man eine Abbildung

$$\lambda\colon\ V \longrightarrow \{0,1\}$$

(oder mit $\{f,w\}$ als Wertebereich).

Eine Wahrheitsbelegung ist also einfach dadurch gegeben, dass einer jeden Aussagenvariablen ein Wahrheitswert, nämlich 0 oder 1 bzw. f oder w zugeordnet wird. Eine solche Wahrheitsbelegung möchte man auf die gesamte Sprache fortsetzen, wobei die folgenden Festlegungen die inhaltliche Bedeutungen der Junktoren widerspiegeln. Die folgende Definition ist möglich, da der rekursive Aufbau einer Aussage eindeutig bestimmt ist.

DEFINITION 2.12. Es sei V eine Menge von Variablen, L^V die zugehörige aussagenlogische Sprache und

$$\lambda\colon\ V \longrightarrow \{0,1\}$$

eine Wahrheitsbelegung. Unter der zugehörigen *Interpretation* $I = I^\lambda$ versteht man die über den rekursiven Aufbau der Sprache festgelegte Abbildung

$$I\colon\ L^V \longrightarrow \{0,1\}$$

mit

(1) $I(v) = \lambda(v)$ für jede Aussagenvariable $v \in V$.
(2) Bei $\alpha = \neg(\beta)$ ist

$$I(\alpha) = \begin{cases} 1, \text{ falls } I(\beta) = 0\,, \\ 0, \text{ falls } I(\beta) = 1\,. \end{cases}$$

(3) Bei $\alpha = (\beta) \wedge (\gamma)$ ist

$$I(\alpha) = \begin{cases} 1, \text{ falls } I(\beta) = I(\gamma) = 1\,, \\ 0 \text{ sonst}\,. \end{cases}$$

(4) Bei $\alpha = (\beta) \vee (\gamma)$ ist

$$I(\alpha) = \begin{cases} 1, \text{ falls } I(\beta) = 1 \text{ oder } I(\gamma) = 1\,, \\ 0, \text{ falls } I(\beta) = I(\gamma) = 0\,. \end{cases}$$

(5) Bei $\alpha = (\beta) \to (\gamma)$ ist

$$I(\alpha) = \begin{cases} 1, \text{ falls } I(\beta) = 0 \text{ oder } I(\gamma) = 1\,, \\ 0, \text{ falls } I(\beta) = 1 \text{ und } I(\gamma) = 0\,. \end{cases}$$

(6) Bei $\alpha = (\beta) \leftrightarrow (\gamma)$ ist

$$I(\alpha) = \begin{cases} 1, \text{ falls } I(\beta) = I(\gamma)\,, \\ 0, \text{ falls } I(\beta) \neq I(\gamma)\,. \end{cases}$$

Bei

$$I(\alpha) = 1$$

sagt man, dass der Ausdruck α bei der Wahrheitsbelegung λ (oder der Interpretation I) wahr wird (oder gilt), andernfalls, dass er falsch wird (nicht gilt). Dafür schreibt man auch $I \vDash \alpha$ bzw. $I \nvDash \alpha$. Mit $I^\vDash$ bezeichnen wir die Menge aller bei der Interpretation I wahren Ausdrücke aus der Sprache.

Wenn $\Gamma \subseteq L^V$ eine Menge an Ausdrücken ist, so bedeutet $I \vDash \Gamma$, dass $I \vDash \alpha$ für alle $\alpha \in \Gamma$ gilt. Dafür sagt man auch, dass Γ bei der Interpretation I gilt oder dass I ein *Modell* für Γ ist.

Beispiel 2.13. Es sei $V = \{p, q, r\}$ und λ sei die Wahrheitsbelegung mit $\lambda(p) = 1$, $\lambda(q) = 1$, $\lambda(r) = 0$. Es sei I die zugehörige Interpretation. Zur Berechnung des Wahrheitswertes von

$$\alpha = (\neg((p) \wedge (\neg(q)))) \to (r)$$

unter dieser Interpretation muss man rekursiv gemäß Definition 2.12 die einzelnen Bestandteile auswerten. Es ist

$$I(\neg q) = 0$$

und somit

$$I((p) \wedge (\neg(q))) = 0.$$

Also ist

$$I(\neg((p) \wedge (\neg(q)))) = 1.$$

Andererseits ist

$$I(r) = 0$$

und daher ist

$$I(\alpha) = 0.$$

Der Ausdruck ist also bei dieser Wahrheitsbelegung nicht wahr.

Abbildungsverzeichnis