

Sprachen vom Typ 0 und Typ 1

In den letzten beiden Kapiteln haben wir uns ausführlich mit den beiden unteren Ebenen (Typ 3 und Typ 2) der Chomsky-Hierarchie befasst. Bevor wir den ersten Teil zu formalen Sprachen und Automaten beenden erwähnen wir hier kurz einige Ergebnisse zu Automatenmodellen, Abschlusseigenschaften und algorithmischen Fragestellungen für die Sprachklassen der beiden oberen Ebenen der Chomsky-Hierarchie (Typ 0 und Typ 1).

Das zur Klasse der Sprachen vom Typ 0 gehörende Automatenmodell sind *Turingmaschinen*. Diese werden ausführlich im Modul “Theoretische Informatik und Logik” besprochen. Sie bestehen aus einer endlichen Kontrolle und nutzen ein unbeschränktes Band als Eingabemedium und Arbeitsspeicher. Auf die Zellen des Bandes kann sequentiell über einen Lese-/Schreibkopf zugegriffen werden. Hierin liegt der wesentliche Unterschied zu Kellerautomaten, bei denen die Zugriffe auf den Keller nur mittels der üblichen Kelleroperationen push und pop zulässig sind. Das Modell von Turingmaschinen ist nicht nur im Kontext formaler Sprachen von Bedeutung, sondern liefert zugleich ein universelles Rechnermodell, welches zu modernen Rechnern hinsichtlich der Berechenbarkeit äquivalent ist. Der Preis für den Zugewinn an Ausdrucksstärke verglichen mit Kellerautomaten ist jedoch hoch. Alle interessanten algorithmischen Fragestellungen (darunter das Wort-, Leerheits- und Äquivalenzproblem u.v.m.) sind für Turingmaschinen unentscheidbar, d.h., nicht algorithmisch lösbar.

Das zu kontextsensitiven Sprachen gehörende Automatenmodell sind *linear beschränkte Automaten* (LBA), eine Variante von Turingmaschinen, die für die Bearbeitung eines Eingabeworts im Wesentlichen nur diejenigen Bandzellen, in denen initial ein Eingabezeichen steht, benutzen können. Für festes Eingabewort sind also die relevanten Bandzellen a-priori festgelegt. Dies ermöglicht den Entwurf von Algorithmen, um das Wortproblem für kontextsensitive Sprachen zu lösen.

Während die nichtdeterministische und deterministische Variante von Turingmaschinen äquivalent sind, ist die entsprechende Frage für LBA noch ungelöst. Die Klasse der kontextsensitiven Sprachen ist – wie die Klasse der regulären Sprachen – unter den fünf wichtigsten Kompositionsooperatoren Vereinigung, Schnitt, Komplement, Konkatenation und Kleeneabschluss abgeschlossen. Die Klasse der Sprachen vom Typ 0 ist abgeschlossen unter Vereinigung, Schnitt, Konkatenation und Kleeneabschluss, jedoch nicht unter Komplement.

2. Teil: Aussagenlogik

Die Logik ist eine sehr alte Wissenschaftsdisziplin, die sich mit der Lehre vom vernünftigen Schlussfolgern (“Beweisen”) befasst. Klassische Problemstellungen der Logik sind die Frage nach Formalisierungen schlüssigen Denkens, oder nach Beweisregeln, welche maschinell realisierbar sind. Tief verwurzelt in der Informatik spielt die Logik in vielen Teilgebieten der Informatik traditionell eine zentrale Rolle. Beispiele sind die Logiksynthese der technischen Informatik, maschinelle Beweiser, die in der künstlichen Intelligenz oder Robotik eingesetzt werden, logische Kalküle beruhend auf Vor- und Nachbedingungen für sequentielle Programme (z.B. Hoare Logik), modale und temporale Logiken, die zur Spezifikation und Verifikation von parallelen Systemen dienen, sowie deduktive Datenbanken oder Logikprogrammiersprachen wie PROLOG oder DATALOG.

4 Aussagenlogik

In dieser Vorlesung befassen wir uns nur mit der Aussagenlogik. Trotz ihrer sehr einfachen Struktur spielt die Aussagenlogik eine wichtige Rolle in der Informatik und anderen Wissenschaftsdisziplinen. Zugleich bildet die Aussagenlogik das Grundgerüst reichhaltigerer Logiken wie der Prädikatenlogik, das es erlaubt über Relationen zwischen Objekten oder den Eigenschaften der Individuen einer Objektmenge zu sprechen.

4.1 Grundbegriffe der Aussagenlogik

Aussagenlogische Konzepte sind bereits aus mehreren anderen Vorlesungen bekannt. So sind z.B. Boolesche Ausdrücke, wie sie als Schleifenabbruchkriterium in imperativen Programmiersprachen vorkommen können, aussagenlogische Formeln. Die aus der technischen Informatik bekannten Schaltnetze sind graphische Darstellungen aussagenlogischer Formeln und Schaltfunktionen deren Semantik.

In diesem Abschnitt gehen wir auf die formale Syntax und Semantik aussagenlogischer Formeln und zugehörigen Grundbegriffen sowie Normalformen ein. Die folgenden Abschnitte befassen sich dann mit dem Sonderfall von Hornformeln, denen in der Logikprogrammierung eine zentrale Rolle zukommt, und algorithmischen Fragestellungen.

4.1.1 Syntax und Semantik

Die Grundbausteine aussagenlogischer Formeln sind atomare Aussagensymbole, die je nach Kontext wahr oder falsch sein können und durch logische Operatoren wie “und”, “oder”, “nicht” zu komplexen Aussagen verknüpft werden können.

Syntax der Aussagenlogik. Im Folgenden bezeichnet AP eine nicht-leere Menge von paarweise verschiedenen Booleschen Variablen, die in diesem Kontext *Aussagensymbole* oder *Atome* oder *atomare Formeln* genannt werden. Die Abkürzung AP steht für die englische Bezeichnung “atomic propositions”. Intuitiv stehen die Aussagensymbole für Aussagen, die entweder wahr

oder falsch (aber nicht beides) sein können. Beispielsweise können sie die intuitive Bedeutung “Programmvariable X hat den aktuellen Wert 5” oder “NFA \mathcal{M} befindet sich in Zustand q ” haben. Zur Bildung komplexer Aussagen verwenden wir die Booleschen Operatoren \neg (Negation) und \wedge (Konjunktion, das logische “und”), aus denen alle anderen Booleschen Verknüpfungen wie u.a. \vee (Disjunktion, das logische “oder”), \rightarrow (Implikation), \leftrightarrow (Äquivalenz) herleitbar sind.

Wir verwenden hier meist Kleinbuchstaben am Ende des Alphabets wie x, y, z als Aussagensymbole, also Elemente von AP , und griechische Buchstaben am Anfang des Alphabets $\alpha, \beta, \gamma, \dots$ für aussagenlogische Formeln.

Zur Angabe der Syntax aussagenlogischer Formeln verfahren wir ähnlich wie für reguläre Ausdrücke (Abschnitt 2.3). Wir geben zunächst eine induktive Definition aussagenlogischer Formeln und dann die abstrakte Syntax in BNF-ähnlicher Notation an. Die Menge aller aussagenlogischen Formeln über AP ist induktiv durch folgende vier Regeln definiert.

1. $true$ ist eine aussagenlogische Formel über AP .
2. Jedes Aussagensymbol $x \in AP$ ist eine aussagenlogische Formel über AP .
3. Sind α und β aussagenlogische Formeln über AP , dann sind auch $(\neg\alpha)$ und $(\alpha \wedge \beta)$ aussagenlogische Formeln über AP .
4. Nichts sonst ist eine aussagenlogische Formel über AP .

Unterstellt man die Regeln für das Setzen von Klammern als Selbstverständlichkeit, so kann obige induktive Definition in kompakter Form in BNF-ähnlicher Notation geschrieben werden (abstrakte Syntax), wobei α und β als Metasymbole für Formeln und x als Metasymbol für die Atome in AP fungieren.

$$\alpha ::= true \mid x \mid \neg\alpha \mid \alpha \wedge \beta$$

Abgeleitete Operatoren. Neben dem Verzicht auf überflüssige Klammern sind vereinfachende Schreibweisen, die sich durch die Verwendung zusätzlicher Symbole ergeben, üblich. Diese werden aus der Konstanten $true$ und den beiden Basisoperatoren \wedge und \neg hergeleitet. Die wichtigsten abgeleiteten Operatoren sind:

$$\begin{aligned} false &\stackrel{\text{def}}{=} \neg true \\ \alpha \vee \beta &\stackrel{\text{def}}{=} \neg(\neg\alpha \wedge \neg\beta) && \text{(Disjunktion)} \\ \alpha \rightarrow \beta &\stackrel{\text{def}}{=} \neg\alpha \vee \beta && \text{(Implikation)} \\ \alpha \leftrightarrow \beta &\stackrel{\text{def}}{=} (\alpha \wedge \beta) \vee (\neg\alpha \wedge \neg\beta) && \text{(Äquivalenz)} \end{aligned}$$

Prioritätsregeln zur Vermeidung von Klammern. Zur Einsparung von Klammern treffen wir die Vereinbarung, dass die Negation am stärksten bindet, dann die Operatoren $\wedge, \vee, \rightarrow$ und \leftrightarrow (in dieser Reihenfolge). Z.B. steht $\neg x \wedge y \vee z$ für $((\neg x) \wedge y) \vee z$ oder $x \wedge y \leftrightarrow \neg y \vee z$ für $(x \wedge y) \leftrightarrow ((\neg y) \vee z)$.

Semantik der Aussagenlogik. Bis hier haben wir nur syntaktische Konstrukte der Aussagenlogik behandelt. Die Interpretation der Konstanten *true* als eine Aussage, die immer wahr ist, oder der Symbole \neg und \wedge als Negation bzw. Konjunktion wird erst durch die Angabe einer entsprechenden Semantik gerechtfertigt. Ebenso ist die Vereinfachung von Formeln durch bekannte ‘‘Gesetze’’ (etwa doppelte Verneinung $\neg\neg\alpha \rightsquigarrow \alpha$) oder der Übergang zu Normalformen erst auf semantischer Ebene sinnvoll, auf der den Symbolen der Logik eine Bedeutung zugewiesen wird. Intuitiv stehen die Atome in AP für Aussagen, die je nach Kontext wahr oder falsch sein können. Der Kontext wird durch eine Abbildung formalisiert, die jedem Aussagensymbol einen Wahrheitswert 0 (‘‘falsch’’) oder 1 (‘‘wahr’’) zuordnet.

Definition 4.1 (Belegung, Interpretation, Wahrheitswert einer Formel). Eine *Interpretation* oder *Belegung* für die Aussagenmenge AP ist eine Abbildung $I: AP \rightarrow \{0, 1\}$. Die Belegungen für $AP = \{x_1, \dots, x_n\}$ geben wir oftmals in der Form $[x_1 = b_1, \dots, x_n = b_n]$ oder $x_1^I = b_1, \dots, x_n^I = b_n$ an und meinen damit die Abbildung $I: \{x_1, \dots, x_n\} \rightarrow \{0, 1\}$, deren Funktionswerte durch $I(x_i) = b_i, i = 1, \dots, n$, gegeben sind. Der durch eine Belegung I induzierte Wahrheitswert $\alpha^I \in \{0, 1\}$ für gegebene aussagenlogische Formel α über AP ist durch strukturelle Induktion definiert, siehe Abbildung 28. ■

$$\begin{aligned} true^I &\stackrel{\text{def}}{=} 1 \\ x^I &\stackrel{\text{def}}{=} I(x) \\ (\neg\alpha)^I &\stackrel{\text{def}}{=} 1 - \alpha^I = \begin{cases} 1 & : \text{ falls } \alpha^I = 0 \\ 0 & : \text{ sonst} \end{cases} \\ (\alpha \wedge \beta)^I &\stackrel{\text{def}}{=} \min\{\alpha^I, \beta^I\} = \begin{cases} 1 & : \text{ falls } \alpha^I = 1 \text{ und } \beta^I = 1 \\ 0 & : \text{ sonst} \end{cases} \end{aligned}$$

Abbildung 28: Definition der Wahrheitswerte aussagenlogischer Formeln unter Belegung I

Die formale Definition entspricht genau der intuitiven Bedeutung des Negations- und Konjunktionsoperators. Dies kann man sich klarmachen, indem man *Wertetafeln* betrachtet und die oben angegebene Definition von α^I einsetzt.

α^I	$(\neg\alpha)^I$
0	1
1	0

α^I	β^I	$(\alpha \wedge \beta)^I$
0	0	0
0	1	0
1	0	0
1	1	1

Für die abgeleiteten Operatoren ergibt sich die erwartete Semantik, siehe Abbildung 29.

Offenbar spielen für den Wahrheitswert α^I nur die Werte $I(x) = x^I$ der in α vorkommenden Atome $x \in AP$ eine Rolle. Für festes α genügt es daher, die Werte x^I der in α vorkommenden Atome vorzugeben, um den Wahrheitswert von α unter I zu bestimmen.

Wir betrachten die Formel $\alpha = (x \vee y) \wedge (\neg x \vee \neg z)$. Die Wahrheitswerte von α unter den acht möglichen Belegungen für x, y, z ergeben sich aus folgender Wertetafel:

$$\begin{aligned}
false^I &= 0 \\
(\alpha \vee \beta)^I &= \max\{\alpha^I, \beta^I\} = \begin{cases} 1 & : \text{ falls } \alpha^I = 1 \text{ oder } \beta^I = 1 \\ 0 & : \text{ sonst} \end{cases} \\
(\alpha \rightarrow \beta)^I &= \begin{cases} 1 & : \text{ falls } \alpha^I \leq \beta^I \\ 0 & : \text{ sonst} \end{cases} \\
(\alpha \leftrightarrow \beta)^I &= \begin{cases} 1 & : \text{ falls } \alpha^I = \beta^I \\ 0 & : \text{ sonst} \end{cases}
\end{aligned}$$

Abbildung 29: Semantik einiger abgeleiteter Operatoren

x^I	y^I	z^I	$(x \vee y)^I$	$(\neg x \vee \neg z)^I$	α^I
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	1	1	1
0	1	1	1	1	1
1	0	0	1	1	1
1	0	1	1	0	0
1	1	0	1	1	1
1	1	1	1	0	0

Erfüllende Belegung, Modell. Falls $\alpha^I = 1$, so nennen wir I eine *erfüllende Belegung* oder ein *Modell* für α . Z.B. ist $I = [x = 1, y = 0, z = 0]$ ein Modell für $(x \vee y) \wedge (\neg x \vee \neg z)$, jedoch ist I nicht erfüllend für $(x \wedge y) \vee (\neg x \wedge \neg z)$. Wir verwenden oftmals lässige Sprechweisen wie “ α ist falsch unter I”, falls $\alpha^I = 0$. Entsprechend machen wir von Formulierungen der Art “ α ist unter I wahr” oder “ α ist unter I erfüllt” Gebrauch, falls I ein Modell für α ist.

Definition 4.2 (Erfüllbarkeit, Gültigkeit). Sei α eine aussagenlogische Formel über AP. α heißt *erfüllbar*, wenn es eine erfüllende Belegung für α gibt, also wenn eine Belegung I für die in α vorkommenden Atome existiert, so dass $\alpha^I = 1$, Formel α heißt *unerfüllbar*, falls α nicht erfüllbar ist, also wenn $\alpha^I = 0$ für alle Belegungen I, Formel α wird *Tautologie* oder *gültig* genannt, wenn $\alpha^I = 1$ für alle Belegungen I. ■

Beispielsweise sind $\alpha = (x \vee y) \wedge \neg(\neg y \rightarrow x)$ oder $\beta = x \leftrightarrow \neg x$ unerfüllbar. Das formale Argument für die Unerfüllbarkeit von β ist, dass unter jeder Belegung I die atomare Formel x und deren Negat $\neg x$ unterschiedliche Wahrheitswerte haben, also $x^I \neq (\neg x)^I$ und somit $\beta^I = (x \leftrightarrow \neg x)^I = 0$. Zum Nachweis der Unerfüllbarkeit von α betrachten wir eine beliebige Interpretation I und zeigen, dass α unter I falsch ist.

- Falls $x^I = 1$, so ist $(\neg y \rightarrow x)^I = 1$ und somit $\neg(\neg y \rightarrow x)^I = 0$ und daher $\alpha^I = 0$.
- Wir nehmen nun $x^I = 0$ an. Falls $y^I = 0$, so ist $(x \vee y)^I = 0$ und somit $\alpha^I = 0$. Falls $y^I = 1$, so ist $(\neg y \rightarrow x)^I = (\neg \neg y \vee x)^I = 1$ und somit $\neg(\neg y \rightarrow x)^I = 0$. Wir erhalten ebenfalls $\alpha^I = 0$.

Die Formel $\gamma = x \vee \neg(x \wedge y)$ ist eine Tautologie. Dies folgt aus der Tatsache, dass jede Belegung I für x und y entweder erfüllend für die linke Teilformel x ist (nämlich dann, wenn $x^I = 1$) oder ein Modell der rechten Teilformel $\neg(x \wedge y)$ ist (nämlich dann, wenn $x^I = 0$).

Die Formeln $\alpha = x \vee \neg y$ und $\beta = x \wedge \neg y$ sind erfüllbar (da $\alpha^I = \beta^I = 1$ für $x^I = 1$ und $y^I = 0$), aber nicht gültig (da $\alpha^I = \beta^I = 0$ für $x^I = 0$ und $y^I = 1$).

Wir fassen zusammen. Der Nachweis der Erfüllbarkeit oder Nicht-Gültigkeit kann durch Angabe eines Modells bzw. einer nicht-erfüllenden Belegung erfolgen. Hingegen erfordert der Nachweis der Unerfüllbarkeit oder Gültigkeit die Betrachtung aller Interpretationen, für die dann zu zeigen ist, dass die betreffende Formel stets falsch (Unerfüllbarkeit) bzw. stets wahr (Gültigkeit) ist. Weitere Beweisoptionen für den Nachweis der Unerfüllbarkeit bzw. Gültigkeit ergeben sich durch die semantischen Begriffe der Äquivalenz von Formeln oder der logischen Konsequenz. Darauf gehen wir später ein. Zunächst stellen wir folgenden offensichtlichen Zusammenhang zwischen Gültigkeit und Unerfüllbarkeit fest:

$$\begin{aligned} \alpha \text{ ist unerfüllbar} & \quad \text{gdw} \quad \alpha^I = 0 \text{ für alle Belegungen } I \\ & \quad \text{gdw} \quad (\neg\alpha)^I = 1 \text{ für alle Belegungen } I \\ & \quad \text{gdw} \quad \neg\alpha \text{ ist gültig} \end{aligned}$$

Beispiel 4.3 (n-Damen-Problem). Das n -Damen-Problem fragt nach einer Platzierung von n Damen auf einem $n \times n$ -Schachbrett, so dass sich die Damen nicht gegenseitig bedrohen. Abweichend von der sonst üblichen Nummerierung der Felder durch Buchstaben und Zahlen denken wir uns das Schachbrett als $n \times n$ -Matrix und verwenden Nummern $1, \dots, n$ für die Zeilen und Spalten.

1		×			
2					×
3			×		
4	×				
5				×	
	1	2	3	4	5

mögliche Lösung für $n = 5$:

je eine Dame in den Feldern
(1, 2), (2, 5), (3, 3), (4, 1), (5, 4)

Das Ziel ist nun eine aussagenlogische Formel α_n anzugeben, deren Modelle genau den Lösungen, also den legitimen Platzierungen der n Damen, entsprechen. Hierzu verwenden wir die Aussagensymbole $x_{i,j}$ für $1 \leq i, j \leq n$ mit der intuitiven Bedeutung, dass $x_{i,j}$ für die Aussage "auf Feld (i, j) befindet sich eine Dame" steht. Die Formel α_n codiert nun die Eigenschaften, dass in jeder Zeile eine Dame steht (damit ist sichergestellt, dass tatsächlich n Damen plaziert werden) und dass keine Dame auf einem Feld steht, das durch eine andere Dame in einem Spielzug erreichbar ist:¹¹

$$\alpha_n = \underbrace{\bigwedge_{1 \leq i \leq n} \bigvee_{1 \leq j \leq n} x_{i,j}}_{\text{in jeder Zeile } i \text{ steht wenigstens eine Dame}} \quad \wedge \quad \underbrace{\bigwedge_{1 \leq i, j \leq n} (x_{i,j} \rightarrow \bigwedge_{(k,l) \in H(i,j)} \neg x_{k,l})}_{\text{die Damen bedrohen sich sich nicht gegenseitig}}$$

¹¹Wir verwenden hier mehrstellige Konjunktions- und Disjunktionsoperatoren. Diese können durch beliebige Klammerung auf den zweistelligen Fall zurückgeführt werden. Eine formale Rechtfertigung hierfür wird später durch die Äquivalenzgesetze (Assoziativität und Kommutativität) für Konjunktionen und Disjunktionen gegeben. Siehe Abbildung 30.

Für jedes Feld (i, j) definieren wir die Menge $H(i, j)$ als diejenige Teilmenge von $\{1, \dots, n\}^2$, die genau aus solchen Feldern (k, ℓ) besteht, die von einer Dame in Feld (i, j) erreichbar sind. Es gilt also $(k, \ell) \in H(i, j)$ genau dann, wenn sich Damen an den Feldpositionen (i, j) und (k, ℓ) gegenseitig bedrohen:

$$\begin{aligned} H(i, j) = & \{(i, \ell) : 1 \leq \ell \leq n, \ell \neq j\} \cup \\ & \{(k, j) : 1 \leq k \leq n, k \neq i\} \cup \\ & \{(k, \ell) : 1 \leq k, \ell \leq n, (k, \ell) \neq (i, j), k - \ell = i - j\} \cup \\ & \{(k, \ell) : 1 \leq k, \ell \leq n, (k, \ell) \neq (i, j), k + \ell = i + j\} \end{aligned}$$

Die erfüllenden Belegungen von α_n stehen dann für die Lösungen des n -Damen-Problems. Insbesondere ist α_n genau dann erfüllbar, wenn das n -Damen-Problem lösbar ist. Z.B. entspricht die oben angegebene Lösung des 5-Damen-Problems mit je einer Dame in den Feldern $(1, 2)$, $(2, 5)$, $(3, 3)$, $(4, 1)$ und $(5, 4)$ der Interpretation I, die durch

$$x_{1,2}^I = x_{2,5}^I = x_{3,3}^I = x_{4,1}^I = x_{5,4}^I = 1 \quad \text{und} \quad x_{k,\ell}^I = 0 \quad \text{für alle anderen Felder}$$

gegeben ist. Man kann sich nun davon überzeugen, dass I tatsächlich ein Modell für α_5 ist. ■