Gesellschaft für Informatik (Hrsg.): SKILL 2019,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2019 1

Baumautomaten als notationelle Variante logischer Matrix-Semantik

Lukas Grätz[1]

Abstract: Automaten sind als Grundlage der theoretischen Informatik in der Regel Bestandteil einer Vorlesung über Automaten und Sprachen im Bachelor. Wahrheitsfunktionen werden für die Semantik der Aussagenlogik ebenfalls in einer Grundlagenveranstaltung behandelt. Wie sich nun herausstellt, sind beide Konzepte ab einem gewissen Abstraktionsgrad äquivalent und nur bezüglich Terminologie und Notation unterschiedlich: (aufsteigende) Baumautomaten und Matrizen für mehrwertige Logik. Neben dem deterministischen Fall ist die nicht-deterministische Semantik Thema meiner Masterarbeit.

Automata are a foundation of theoretical computer science and part of undergraduate lectures on automata and languages. Similarly, truth functions for the semantics of propositional logic are covered in foundational classes. Assuming a certain level of abstraction both concepts are equivalent and only differ in terms of notation and terminology: bottom-up tree automata for tree languages are equivalent to matrix semantics for many-valued logics. The deterministic case is enriched by non-deterministic semantics in my master's thesis.

1 Einleitung

Automaten sind als Grundlagen der theoretischen Informatik in der Regel Bestandteil einer Vorlesung über Automaten und Sprachen im Bachelor. Ein *endlicher deterministischer Automat* (DFA) ist ein Tupel $\mathbf{A}_1 = (\{q_0, \ldots, q_6\}, \Sigma, \delta, q_0, \{q_5\})$ über dem Alphabet $\Sigma = \{A, B, \ldots, Z\}$ mit der durch Abb. 1 gegebenen Übergangsfunktion δ.

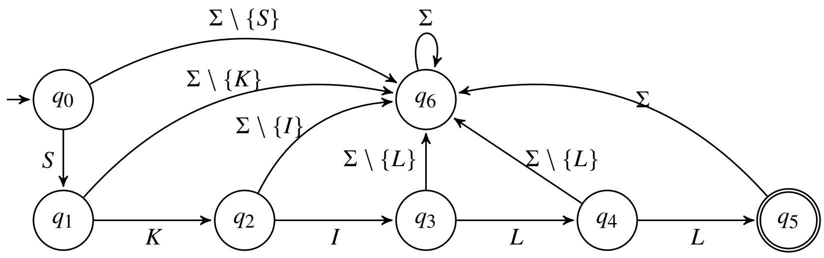

Abb. 1: Übergangsfunktion δ für den DFA $\mathbf{A}_1$

Dieser Automat beginnt im Zustand q_0 zeichenweise ein Eingabewort aus den Symbolen des Alphabets Σ einzulesen. Der nächste Zustand ist durch das neu eingelesene Symbol jeweils

[1] Universität Leipzig, lukas.graetz@studserv.uni-leipzig.de

bezüglich der Übergangsfunktion δ determiniert (δ ist hier durch die Pfeile im Graph aus Abb. 1 gegeben). Falls sich der Automat am Ende der Eingabe im akzeptierenden Zustand q_5 befindet, so wird das Wort akzeptiert. Dieser Automat akzeptiert nur das Wort „SKILL“, also die Sprache $\{\text{SKILL}\}$.

Etwas nützlicher ist ein Automat, der Zeichenketten akzeptiert, die den Teilausdruck „SKILL“ enthalten. Dies kann z. B. mit einem *nichtdeterministischen endlichen Automaten* (NFA) $\mathbf{A}_2 = (\{q_0, \dots, q_5\}, \Sigma, \Delta, q_0, \{q_5\})$ erreicht werden.

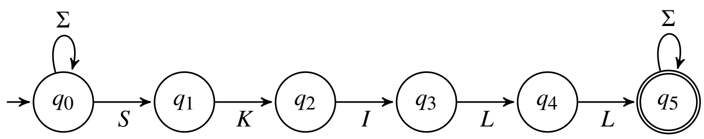

Abb. 2: Übergangsrelation Δ für den NFA $\mathbf{A}_2$

Beim NFA ist der nächste Zustand durch die Übergangsrelation Δ, gegeben durch Abb. 2, nicht mehr eindeutig determiniert, beispielsweise sind in q_0 beim eingelesenen Buchstaben S zwei Übergänge möglich. Der Automat akzeptiert, wenn Zustandsübergänge zum finalen akzeptierenden Zustand möglich sind. Das Wort „ABCSKSKILLDEF“ wird also akzeptiert; die akzeptierte Sprache kann auch durch den regulären Ausdruck $\Sigma^* \text{ SKILL } \Sigma^*$ beschrieben werden.

Wahrheitsfunktionen werden für die Semantik der Aussagenlogik ebenfalls in einer Grundlagenveranstaltung behandelt. Aussagenlogik baut bereits auf einer Sprache von Formelausdrücken auf, die aus den Operatoren $\wedge$, $\vee$, $\rightarrow$ und $\neg$ sowie aus Klammern und Aussagenvariablen (für das Beispiel *motiviert*, *interessiert* und *willkommen*) besteht. Die Semantik ist durch eine Belegung der Aussagenvariablen mit den Wahrheitswerten 0, 1 und Wahrheitsfunktionen nach Abb. 3 gegeben:

$f_\wedge$	0	1
0	0	0
1	0	1

$f_\vee$	0	1
0	0	1
1	1	1

$f_\rightarrow$	0	1
0	1	1
1	0	1

	$f_\neg$
0	1
1	0

Abb. 3: Wahrheitsfunktionen für die Aussagenlogik

Sobald die Aussagenvariablen mit Wahrheitswerten belegt sind (z.B. mit $v(\textit{motiviert}) = 0$, $v(\textit{interessiert}) = 1$ und $v(\textit{willkommen}) = 1$), ist die Bewertung v eines Formelausdrucks aus der Bewertung der Teilausdrücke und den Wahrheitsfunktionen für die Operatoren eindeutig bestimmt. Beispielsweise muss (1) gelten.

$$\begin{aligned} & v\big((\textit{motiviert} \;\vee\; \textit{interessiert}) \;\rightarrow\; \textit{willkommen}\big) \\ &= f_\rightarrow\big(f_\vee(v(\textit{motiviert}), v(\textit{interessiert})),\; v(\textit{willkommen})\big) \end{aligned} \tag{1}$$

Die logische Folgerung ist durch die Konsequenzrelation $\vDash$ definiert: Eine Aussage F folgt aus einer Menge von Aussagen M, falls jede Bewertung, die alle Aussagen aus M mit 1 bewertet, auch die Aussage F mit 1 bewertet. So gilt z.B.

$$(\mathit{motiviert} \lor \mathit{interessiert}) \to \mathit{willkommen}, \mathit{interessiert} \vDash \mathit{willkommen}.$$

Automaten und die Semantik der Aussagenlogik sind also in ihrer Funktionsweise recht ähnlich: Der Zustand eines Automaten beim Wort ws ist durch den Zustand beim Teilwort w und dem neu eingelesenen Symbol s anhand der Übergangsfunktion determiniert. Der Wahrheitswert eines Ausdrucks $A \odot B$ ist durch die Wahrheitswerte der Ausdrücke A und B anhand der Wahrheitsfunktion von $\odot$ determiniert.

Mitte der 1960er Jahre wurden Baumautomaten als natürliche Erweiterung der algebraischen Beschreibung endlicher Automaten entdeckt. Mehrwertige Logiken (und auch die Matrix-Semantik mehrwertiger Logiken) als Erweiterung der klassischen zweiwertigen Logik haben hingegen eine viel längere Tradition ab 1920, mit bekannten Namen wie Jan Łukasiewicz, Emil Post, Paul Bernays und Alfred Tarski (siehe [ŁT30]).

Nun stellt sich sogar heraus, dass Baumautomaten und logische Matrizen für mehrwertige Logiken einfach notationelle Varianten sind. Zunächst lassen sich aussagenlogische Formeln bekanntermaßen auch als Syntaxbaum darstellen, wobei in den Blättern Aussagenvariablen und in den Knoten logische Operatoren stehen. Eine Variablenbelegung entspricht dann genau einer Initialzuweisung der Blattsymbole mit Startzuständen. Generell entsprechen die Zustände eines Automaten den Wahrheitswerten, ausgezeichnete Wahrheitswerte sind die akzeptierenden Endzustände.

Doch zunächst seien hier die Definitionen (bezüglich aktueller Literatur) wiedergegeben. Ich habe die Notation nur unwesentlich geändert, um trotz üblicher (und sinnvoller) Vereinheitlichung der Notation einen Vergleich beider Strukturen zu ermöglichen, ohne dass der zugehörige Begriffsapparat den Umfang dieses Aufsatzes sprengt.

2 Baumautomaten

Baumautomaten unterscheiden sich von DFAs insofern, als dass keine Zeichenketten sondern Bäume eingelesen werden. Ein Baumautomat akzeptiert also keine Sprache aus Wörtern, sondern einen Wald, d.h. eine Menge aus Bäumen. Die folgende Definition ist angelehnt an [GS97, S. 9]. Hierfür sei Σ ein *Rangalphabet*, ein Alphabet in dem jedem Symbol $s \in \Sigma$ eine Stelligkeit $\mathit{Rang}(s) \in \mathbb{N}$ zugeordnet wird. Das auch als Blattalphabet bezeichnete Alphabet X hat keine Stelligkeit. Alternativ kann die Beschriftung eines Blattes auch ein nullstelliges Symbol sein ($s \in \Sigma$ mit $\mathit{Rang}(s) = 0$).

Definition 1. Sei Σ ein *Rangalphabet* und X ein Alphabet. Ein (deterministischer) *aufsteigender Baumautomat* ist ein Tupel $\mathbf{A} = (\mathcal{A}, \alpha, Q_+)$ mit einer endlichen Algebra $\mathcal{A} = (Q, \{s^{\mathcal{A}} \mid s \in \Sigma\})$, einer Initialzuweisung $\alpha\colon X \to Q$ und einer Teilmenge $Q_+ \subseteq Q$, den sogenannten akzeptierenden Endzuständen.

Genau genommen ist zwischen einem Symbol $s \in \Sigma$, dessen Stelligkeit $Rang(s) = m$ und der zugehörigen m-stelligen Operation $s^{\mathcal{A}}$ bezüglich der Algebra $\mathcal{A} = (Q, \Sigma)$ zu unterscheiden [GS97, S. 5]. Falls der Begriff „Algebra" hier für Verunsicherung sorgen sollte: Eine Algebra $\mathcal{A} = (Q, \{s^{\mathcal{A}} \mid s \in \Sigma\})$ ist nichts weiter als eine Menge Q und Operationen $s^{\mathcal{A}}$, deren Argumente und Rückgabe Elemente von Q sind.

Die Initialzuweisung α kann eindeutig zu einer Zuweisung $\alpha^{\mathcal{A}}$ erweitert werden, die jedem Baum einen Zustand $q \in Q$ zuordnet: $\alpha^{\mathcal{A}}(s(t_1, \dots, t_m)) = s^{\mathcal{A}}(\alpha^{\mathcal{A}}(t_1), \dots, \alpha^{\mathcal{A}}(t_m))$ für Teilbäume $t_1, \dots, t_m$. Ein Baum t wird durch $\mathbf{A}$ akzeptiert, falls $\alpha^{\mathcal{A}}(t) \in Q_+$.

3 Mehrwertige Matrizen

Die folgende Definition ist angelehnt an [BB92, S. 20]:

Definition 2. Sei $\mathcal{A} = (A, \{f_i \mid i \in I\})$ eine Algebra. Jedes Tupel der Form $(\mathcal{A}, A_*)$ mit A_* einer nichtleeren Teilmenge heißt Matrix. Wenn $\mathcal{M} = (\mathcal{A}, A_*)$ eine Matrix ist, dann wird $\mathcal{A}$ die Algebra von $\mathcal{M}$ genannt.

Eine Belegung der Aussagenvariablen Var ist eine Funktion $v\colon Var \to A$ in die Wertemenge A der Algebra $\mathcal{A}$. Die Belegung v kann nun zu einer Bewertung $v^{\mathcal{A}}$ erweitert werden, mit $v^{\mathcal{A}}(\odot(a_1, \dots, a_m)) = f_\odot(v^{\mathcal{A}}(a_1), \dots, v^{\mathcal{A}}(a_m))$ für Teilausdrücke $a_1, \dots, a_m$.

Beispiel 3. Die Aussagenlogik lässt sich durch die Matrix $\mathcal{M} = \big((\{0,1\}, \{f_\wedge, f_\vee, f_\to, f_\neg\}), \{1\}\big)$ definieren, wobei die Operationen durch Abb. 3 gegeben sind.

4 Ein Automat für Aussagenlogik

Wie oben schon beschrieben, ist die Übersetzung einer mehrwertigen Matrix in einen Baumautomaten ausschließlich eine Frage der Notation. Dies soll nun kurz an einem schon vorgestellten Beispiel durch den *aufsteigenden Baumautomaten* $\mathbf{A}_3 = (\mathcal{A}_3, \alpha_3, \{1\})$ mit $\mathcal{A}_3 = (Q_3, \Sigma_3)$, $Q_3 = \{0, 1\}$,

$$\begin{aligned}
\Sigma_3 = \{\wedge^{\mathcal{A}_3}\colon &\quad Q_3 \times Q_3 \to Q_3\colon (0,0) \mapsto 0, (0,1) \mapsto 0, (1,0) \mapsto 0, (1,1) \mapsto 1,\\
\vee^{\mathcal{A}_3}\colon &\quad Q_3 \times Q_3 \to Q_3\colon (0,0) \mapsto 0, (0,1) \mapsto 1, (1,0) \mapsto 1, (1,1) \mapsto 1,\\
\to^{\mathcal{A}_3}\colon &\quad Q_3 \times Q_3 \to Q_3\colon (0,0) \mapsto 1, (0,1) \mapsto 1, (1,0) \mapsto 0, (1,1) \mapsto 1,\\
\neg^{\mathcal{A}_3}\colon &\quad Q_3 \to Q_3\colon\ 0 \mapsto 1,\ 1 \mapsto 0\ \}
\end{aligned}$$

und $X_3 = \{\textit{motiviert}, \textit{interessiert}, \textit{willkommen}\}$ dargestellt werden. Eine Belegung ist noch kein Bestandteil einer Matrix, da für die Konsequenzrelation jede Belegung der Aussagenvariablen X_3 betrachtet werden muss. Eine Belegung entspricht dann z. B. der Initialzuweisung

$$\alpha_3\colon X_3 \to Q_3\colon \textit{motiviert} \mapsto 0,\ \textit{interessiert} \mapsto 1,\ \textit{willkommen} \mapsto 1.$$

Der Baum in Abb. 4 stellt die Formel (*motiviert* $\vee$ *interessiert*) $\rightarrow$ *willkommen* dar. Da der Baumautomat diesen Baum akzeptiert, handelt es sich bei α_3 um eine erfüllende Belegung.

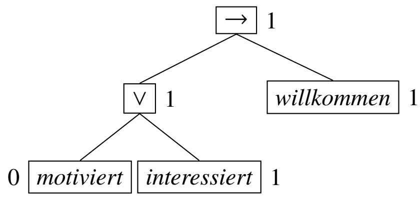

Abb. 4: Baum inklusive Zuweisung bezüglich Baumautomat $\mathbf{A}_3$

5 Reguläre Sprachen als Logiken

Jeder DFA kann in eine logische Matrix umgewandelt werden: Zustände werden dann Wahrheitswerte, akzeptierende Endzustände werden zu ausgezeichneten Wahrheitswerten und aus den Symbolen werden einstellige Postfixoperatoren (in umgekehrter polnischer Notation).

Sei $\mathbf{A} = (Q, \Sigma, \delta, q_0, Q_+)$ ein DFA und $w = s_1 s_2 \ldots s_n$ ein Eingabewort aus den Symbolen $s_i \in \Sigma$ für $1 \leq i \leq n$. Bekanntermaßen wird w von A akzeptiert, falls $\delta(\cdots \delta(\delta(q_0, s_1), s_2) \cdots, s_n) \in Q_+$.

Der Automat lässt sich mit Hilfe des üblichen Transitionsmonoids in eine Matrix $\mathcal{M} = (\mathcal{A}, Q_+)$ umschreiben, mit $\mathcal{A} = (Q, \mathcal{O})$ und

$$\mathcal{O} = \{f_s : q \mapsto \delta(q, s) \mid s \in \Sigma, q \in Q\} \cup \{f_\epsilon : q_0\}.$$

Jedes Symbol im Eingabewort wird zu einem 1-stelligen Postfixoperator, also jeweils angewandt auf den Wahrheitswert (vormals Zustand) des Präfixes. Das leere Wort ϵ als Präfix vor dem ersten Symbol ist eine Ausnahme. Es erhält als 0-stelliger Operator den Wahrheitswert q_0. Umgeschrieben in gewohnte Klammerschreibweise wäre der Ausdruck also $s_n(\cdots s_2(s_1(\epsilon)) \cdots)$.

Der Wahrheitswert eines Wortes unter der Bewertung $v^{\mathcal{A}}$ erfüllt die Bedingungen $v^{\mathcal{A}}(s_1 s_2 \cdots s_i) = f_{s_i}(v^{\mathcal{A}}(s_1 s_2 \cdots s_{i-1}))$ für alle $1 \leq i \leq n$ und es gilt $v^{\mathcal{A}}(\epsilon) = f_\epsilon = q_0$. Damit gilt $v^{\mathcal{A}}(w) \in Q_+$ genau dann, wenn der Automat das Wort akzeptiert. Die Sprache des Automaten $\mathcal{L}(\mathbf{A})$ besteht also aus genau den bezüglich der Matrix $\mathcal{M}$ erfüllbaren Ausdrücken.

6 Fazit und Ausblick

Die zentrale Einsicht aus diesem Aufsatz ist nicht Abschnitt 4, dass aufsteigende Baumautomaten für die Aussagenlogik erstellt werden können (dies wird z.B. auch durch Example

2.2 in [GS84, S. 61] illustriert). Auch die Umwandlung eines endlichen Automaten in eine logische Matrix bietet hier nur eine Illustration, die eine Verbindung zwischen Grammatik und Semantik formaler Sprachen aufzeigt.

Die eigentliche Erkenntnis ist, dass Baumautomaten und die Semantik mehrwertiger Logik zusammenfallen: Einerseits die Verallgemeinerung der klassischen Zeichenkettenautomaten und andererseits die Erweiterung der zweiwertigen Semantik der klassischen Aussagenlogik. Definition 1 und 2 fallen also zusammen, ein feiner Unterschied besteht jedoch: Die Belegung wird unabhängig von der Matrix angegeben, wohingegen die Initialzuweisung Bestandteil des Baumautomaten ist. Ein Forschungsthema könnte sein, die Theoreme, Sätze und Lemmata beider Gebiete ineinander zu überführen. Eine guter Überblick über weiterführende Themen zur Automatentheorie bietet [FV09], für mehrwertige Logik sei auf [Hä01] verwiesen.

Und schließlich werden Automaten anders eingesetzt, denn ein Automat berechnet, ob ein gegebenes Wort bzw. ein gegebener Baum Bestandteil der vom Automaten akzeptierten Sprache ist. Mit einem Baumautomaten wird eine Sprache definiert und nicht eine semantische Relation (die logische Folgerung) zwischen den Ausdrücken der Sprache. Als Analogie kann man mit einer logischen Matrix und einer Belegung zeigen, dass ein Ausdruck erfüllbar ist, eine logische Folgerung ist für eine Matrix definiert.

Das Thema meiner Masterarbeit ist eine relativ neue Entwicklung der nichtdeterministischen Semantik [AZ11] und speziell der nichtdeterministischen Semantik für Modallogik auf Basis von [OS16] und [Ke81]. Spannend ist dann auch die Beziehung zwischen nichtdeterministischer Semantik und nichtdeterministischen Automaten.

7 Danksagung

Ein besonderer Dank geht an Dr. Peter Steinacker und Prof. Dr. Andreas Maletti für die Betreuung meiner Masterarbeit.

Literatur

[AZ11] Avron, A.; Zamansky, A.: Non-deterministic Semantics for Logical Systems. In (Gabbay, D.; Guenthner, F., Hrsg.): Handbook of Philosophical Logic. 2. Aufl., Bd. 16, Springer, Dordrecht, S. 227–304, 2011, ISBN: 978-94-007-0478-7.

[BB92] Bolc, L.; Borowik, P.: Many-Valued Logics 1: Theoretical Foundations. Springer, Berlin Heidelberg New York, 1992, ISBN: 3-540-55926-4.

[FV09] Fülöp, Z.; Vogler, H.: Weighted Tree Automata and Tree Transducers. In (Droste, M.; Kuich, W.; Vogler, H., Hrsg.): Handbook of Weighted Automata. Springer, Berlin Heidelberg, S. 313–403, 2009, ISBN: 978-3-642-01491-8.

[GS84] Gécseg, F.; Steinby, M.: Tree Automata. Akadémiai Kiadó, Budapest, 1984, ISBN: 963-05-3170-4.

[GS97] Gécseg, F.; Steinby, M.: Tree Languages. In (Rozenberg, G.; Salomaa, A., Hrsg.): Handbook of Formal Languages. Bd. 3, Springer, Berlin Heidelberg, S. 1–68, 1997, ISBN: 3-540-60649-1.

[Hä01] Hähnle, R.: Advanced Many-valued Logics. In (Gabbay, D.; Guenthner, F., Hrsg.): Handbook of Philosophical Logic. 2. Aufl., Bd. 2, Springer, Dordrecht, S. 297–395, 2001, ISBN: 978-90-481-5753-2.

[Ke81] Kearns, J. T.: Modal Semantics without Possible Worlds. The Journal of Symbolic Logic 46/1, S. 77–86, 1981.

[ŁT30] Łukasiewicz, J.; Tarski, A.: Badania nad ruchunkiem zdań = Untersuchungen über den Aussagenkalkül. Sprawozdania z posiedzeń Towarzystwa Naukowego Warszawskiego, Wydział 3, Nauk Matematyczno-Fizycznych = Comptes rendus des séances de la Société des Sciences et des Lettres de Varsovie, III, Classe des Sciences Mathématiques et Naturelles 23/, S. 30–50, 1930.

[OS16] Omori, H.; Skurt, D.: More Modal Semantics without Possible Worlds. The IfCoLog Journal of Logics and their Applications 3/5, 2016.