Institut für Formale Methoden der Informatik
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Studienarbeit Nr. 2298

Logik erster Stufe ohne Quantorenalternierung über endlichen Wörtern

Martin P. Seybold

Studiengang: Informatik

Prüfer: Prof. Dr. V. Diekert

Betreuer: Dr. M. Kufleitner

begonnen am: 26.07.2010

beendet am: 25.01.2011

CR-Klassifikation: F4.1, F4.3

Inhaltsverzeichnis

1 Einleitung

Reguläre Sprachen sind in vielerlei Hinsicht wichtig für theoretische aber auch praktische Probleme der Informatik. Ein Beispiel für reguläre Sprachen ist das Suchen nach Vorkommen von bestimmten Mustern in größeren Texten. Die Zusammenhänge reguläre Sprachen zu endlichen Automaten sowie zu endlichen Halbgruppen, bzw. Monoide sind klassische Resultate der theoretischen Informatik. [13][6] Wörter kann man auch als lineare Anordnung von Buchstaben interpretieren. Dadurch erscheint es sinnvoll Muster durch Prädikatenlogische Formeln, bezüglich Position und Beschriftung, zu definieren. Durch Einschränkung auf bestimmte Prädikate, Variablenzahl oder Quantorenalternierungen ergeben sich auf natürliche Weise Hierarchien dieser Logikfragmente. Ziel dieser Studienarbeit ist es die Zusammenhänge zwischen regulären Sprachen und Prädikatenlogik erster Stufe ohne Quantorenalternierungen zu untersuchen.

Die Ergebnisse dieser Arbeit entstanden in Zusammenarbeit mit Jonathan Kausch, Manfred Kufleitner und Alexander Lauser.

1.1 Motivation

Trotz des endlichen Charakters der Beschreibung regulärer Sprachen ist ihre Klasse dennoch nicht trivial. Eine echte Unterklasse sind die sternfreien Sprachen. Diese sind durch reguläre Ausdrücke ohne $*$ Operator gegeben, jedoch sind Konkatenationen, von sternfreien Sprachen und Komplementen, erlaubt. Die Punkt-Tiefe einer sternfreien Sprache gibt an, wieviele Konkatenationen benötigt werden, um die Sprache zu definieren. 1971 zeigten McNaughton und Papert, dass sternfreie Sprachen genau der Klasse durch Prädikatenlogik erster Stufe definierbaren Sprachen entsprechen.[10]

Von Schützenberger wurde erstmals gezeigt, dass die sternfreien Sprachen genau die sind, deren syntaktisches Monoid aperiodisch ist. [14] Diese algebraische Eigenschaft liefert für endliche Monoide sofort Entscheidbarkeit. Bald darauf wurde die sogenannte Dot-Depth-Hierarchie B_n der sternfreien Sprachen von Brzozowski und Cohen eingeführt.[2] Diese hat sich als echte und unendliche Hierarchie herausgestellt.[9] Es konnte außerdem gezeigt werden, dass die Anzahl benötigter Quantorenalternierungen einer FO Formel das Level in der Dot-Depth-Hierarchie bestimmt.[20] Effektiv das Level einer gegebenen regulären Sprache in dieser Hierarchie zu bestimmen, ist eine aktuelle Herausforderung der Automatentheorie. Bis heute sind nur Entscheidungsverfahren für die Stufen $n = 1/2, 1, 3/2$ bekannt. Das Ergebnis von Knast für die Stufe $n = 1$ [8][19] wird als schwieriges Resultat betrachtet. Jedoch ist der grundsätzliche Ansatz, Entscheidbarkeit über algebraische Eigenschaften zu erhalten, sehr vielversprechend.

Noch vor dem ersten Entscheidbarkeitsresultat wurde die Straubing-Thérien Hierarchie L_n als alternative Beschreibung der Komplexität sternfreier Sprachen eingeführt.[16] Es konnte gezeigt werden, dass die Level beider Hierarchien über ein Kranzprodukt miteinander zusammenhängen.[11][17] Simon konnte bereits 1975 eine algebraische Charakterisierung für Sprachen geben, die aus L_1 sind.[15] Klíma hat 2009 einen elementaren kombinatorischen Beweis für Simons Theorem vorgestellt [7]. Ziel dieser Studienarbeit ist es, auch für Knasts Theorem einen elementaren kombinatorischen Beweis anzugeben. Dazu sollen zuerst die notwendigen Grundtechniken (Kapitel 2) und Klímas Beweis (Kapitel 3) in einer leichter zu verallgemeinernden Notation dargestellt werden. Anschließend soll Knasts Theorem mit diesen Techniken bewiesen werden (Kapitel 4). Außerdem soll versucht werden, ob sich dies leicht auf weitere Klassen übertragen lässt (Kapitel 5,6). Diese verallgemeinerte Beweismethode liefert sogar neue Entscheidbarkeitsergebnisse für weitere Logikfragmente.

1.2 Grundlegende Definitionen

In diesem Abschnitt werden verschiedene Möglichkeiten beschrieben, um formale Sprachen über endlichen Wörtern zu beschreiben. In der ganzen Arbeit wird mit Γ das endliche Alphabet bezeichnet. Wir betrachten auch oft gewisse Ausschnitte von Wörtern $u = a_1 \dots a_n \in \Gamma^*$ mit der Notation $u|_{[i,j]} = a_i \dots a_j$ für $1 \leq i \leq j \leq n$. Weiter sollen $|u| := n$ die Länge des Wortes u, $first_m(u), last_m(u)$ die ersten bzw. letzten m Zeichen von u und $alph_m(u) = \{\, v \in \Gamma^+ \;\mid\; |v| \leq m, u = pvq \text{ mit } p, q \in \Gamma^* \,\}$ die Fakotren der Länge maximal m von u sein.

Definition 1.1 (Sternfrei)**:** Die sternfreien Sprachen sind Induktiv definiert. $\emptyset$ und $\{a\}$ mit $a \in \Gamma$ sind sternfrei. Sind $L_1, L_2 \subseteq \Gamma^*$ sternfreie Sprachen, dann sind auch $L_1 \cup L_2,\ L_1 \cdot L_2,\ \overline{L_1} := \Gamma^* \setminus L_1$ sternfreie Sprachen.

Es lassen sich nun die Straubing-Thérien und die Dot-Depth Hierarchie jeweils induktiv definieren.[5] Mit $\mathbb{B}$ bezeichnen wir endliche boolsche Kombinationen.

Definition 1.2 (Straubing-Thérien Hierarchie L_n)**:** Sei $n \in \mathbb{N}$, dann sind:

$$\begin{aligned}
L_0 &:= \{\, \Gamma^* a \Gamma^* \;\mid\; a \in \Gamma \,\} \cup \{\emptyset\} \\
L_{n+1/2} &:= \bigcup_{\text{endlich}} K_0 a_1 \dots K_{k-1} a_k K_k && K_i \in L_n, a_i \in \Gamma \\
L_{n+1} &:= \mathop{\mathbb{B}}_{K \in L_{n+1/2}} K
\end{aligned}$$

Definition 1.3 (Dot-Dept Hierarchie B_n)**:** Sei $n \in \mathbb{N}$, dann sind:

$$\begin{aligned}
B_0 &:= \{\, u\Gamma^* v \;\mid\; u, v \in \Gamma^*, |uv| \geq 1 \,\} \cup \{\, \{w\} \;\mid\; w \in \Gamma^+ \,\} \\
B_{n+1/2} &:= \bigcup_{\text{endlich}} K_0 a_1 \dots K_{k-1} a_k K_k && K_i \in B_n, a_i \in \Gamma \\
B_{n+1} &:= \mathop{\mathbb{B}}_{K \in B_{n+1/2}} K
\end{aligned}$$

Bemerkung 1.4: Eine Sprache aus $\mathcal{L}_1$ hat die Form $\mathbb{B}(\Gamma^* a_1 \dots a_n \Gamma^*, a_i \in \Gamma)$. Diese Form wird auch piecewise-testable genannt. Weiter hat eine Sprache aus B_1 die Form $\mathbb{B}(w_0 \Gamma^* \dots \Gamma^* w_n, w_i \in \Gamma^+)$.

Logik erster Stufe

Wenn man sich Wörter als endliche Sequenz von Buchstaben aus dem Alphabet vorstellt, kann man mit prädikatenlogischen Formeln erster Stufe Bedingungen an die Beschriftung einer Position x stellen: $\lambda(x) = a_i, \ a_i \in \Gamma$. Für eine breitere Übersicht geben wir [3] an. Wir definieren nun die Syntax einer solchen Formel:

Definition 1.5 (FO)**:** Eine Formel $\varphi \in FO$ ist induktiv definiert. Die atomaren Formeln $\lambda(x) = a$ und $\top$ sind in FO, wobei x eine Variable, $a \in \Gamma$ ein Buchstabe und $\top$ die konstante Formel, welche stets wahr ist, sind. Wenn $\varphi, \psi \in FO$ dann sind auch folgende zusammengesetzte Formeln in FO:

$$\neg\varphi, \ \varphi \wedge \psi, \ \exists x\varphi \ \in FO$$

Dadurch sind auch in die konstant falsche Formel $\bot = \neg\top$, die Oder-Verknüpfung $\varphi \vee \psi = \neg(\neg\varphi \wedge \neg\psi)$ und All-Quantifizierungen $\forall x \ \varphi = \neg\exists x \ \neg\varphi$ erlaubt. Die Semantik dieser Konstrukte ist wie üblich definiert. Das Prädikat $\lambda(x) = a_i$ ist auf u genau dann wahr, wenn $u|_x = a$.

Erlauben wir zu dem stets enthaltenen $\lambda(x)$ Prädikat zusätzlich weitere atomare Prädikate $<(x, y), \ +1(x, y), \ \max(x), \ \min(y)$, geben wir dies als Signatur des Logikfragments, wie beispielsweise $FO[<, +1. \min]$, an.

Definition 1.6: Die Prädikate $<, \ +1, \ \min, \ \max$ haben die folgende Syntax und Semantik auf einem Wort $u \in \Gamma^*$

$$\begin{array}{lll}
u \models \ <(x, y) & :\Leftrightarrow & 1 \leq x < y \leq |u| \\
u \models \ +1(x, y) & :\Leftrightarrow & x + 1 = y \\
u \models \ \min(x) & :\Leftrightarrow & 1 = x \leq |u| \\
u \models \ \max(x) & :\Leftrightarrow & 1 \leq x = |u|
\end{array}$$

Im Zusammenhang mit $+1$ ist es möglich die min bzw. max-Prädikate ähnlich dem λ Prädikat zu verwenden. Daher verwenden wir auch die folgenden Makros:

$$\min(a_1 \dots a_n) := \exists x_1, \dots, x_n \min(x_1) \wedge \bigwedge_{i=2}^{n} +1(x_{i-1}, x_i) \wedge \bigwedge_{i=1}^{n} \lambda(x_i) = a_i$$
$$\max(a_1 \dots a_n) := \exists x_1, \dots, x_n \max(x_n) \wedge \bigwedge_{i=2}^{n} +1(x_{i-1}, x_i) \wedge \bigwedge_{i=1}^{n} \lambda(x_i) = a_i$$

Nun sind wir in der Lage die von einer Formel definierte Sprache zu definieren.

Definition 1.7: Die von einer Formel φ eines Logikfragments definierte Sprache $L(\varphi)$ ist

$$L(\varphi) := \{\, u \in \Gamma^* \ \mid u \models \varphi \,\} .$$

Jede Prädikatenlogische Formel besitzt eine äquivalente Formel in Pränexnormalform (PNF) [12]. Jede Formel hat also eine äquivalente Formel, bei der alle Quantoren vor den aussagenlogischen Verknüpfungen der Prädikate vorkommen. Durch die Anzahl der Quantoren bzw. die Alternierungszahl können wir das folgende Logikfragmente definieren.

Definition 1.8 (Logikfragmente)**:** Mit $\Sigma_n \subseteq FO$ bezeichnen wir die Klasse an Formeln, welche eine äquivalente Formel in PNF mit maximal n Quantorenalternierungen, beginnend mit einem Existenzquantor besitzen. Der boolsche Abschluss $\mathbb{B}\Sigma_n$ solch einer Klasse enthält entsprechend alle endlichen boolschen Kombinationen von Formeln aus Σ_n.

Im Folgenden werden wir auch den Ausdruck $L \in \mathbb{B}\Sigma_n$ für eine Sprache $L \subseteq \Gamma^*$ verwenden, falls $L = L(\varphi)$ für eine Formel $\varphi \in \mathbb{B}\Sigma_n$.

Lemma 1.1 (Σ_1-Normalform)**:** *Eine Formel $\varphi \in \Sigma_1[<, +1]$ lässt sich als $\varphi = \bigvee \varphi_n$ darstellen, wobei die Klauseln φ_n die Gestalt*

$$\varphi_n = \exists x_1, \ldots, x_{k_n} : \bigwedge_{i \in I_n} \lambda(x_i) = a_{n_i} \wedge \bigwedge_{(j_1, j_2) \in J_n} +1(x_{j_1}, x_{j_2}) \wedge \bigwedge_{(k_1, k_2) \in K_n} < (x_{k_1}, x_{k_2})$$

haben.

Beweis: Die Darstellung als disjunktive Normalform (DNF) ist immer möglich. Es verbleibt also zu zeigen, dass keine Negationen notwendig sind. Sei die Formel dazu bereits in DNF. Enthält ein solches φ_n einen Block $\lambda(x) \neq a_1 \wedge \ldots \wedge \lambda(x) \neq a_k$, so kann dieser in

$$\lambda(x) = \Gamma \setminus \{a_1, \ldots, a_k\} = \bigvee_{a \in \Gamma \setminus \{a_1, \ldots, a_k\}} \lambda(x) = a$$

transformiert werden, da Γ endlich ist. $\neg(+1(x, y))$ kann auf $<(y, x) \vee (+1(x, z) \wedge <(z, y))$ und $\neg(<(x, y))$ kann auf $x = y \vee <(y, x)$ zurückgeführt werden. $\square$

Algebraische Grundlagen

Klassische Resultate der theoretischen Informatik sind, dass reguläre Sprachen genau die Sprachen sind, welche von deterministischen endlichen Automaten erkannt werden. Der Minimalautomat besitzt einen engen Zusammenhang zu Halbgruppen und Monoiden. [13] [18].

Definition 1.9 (Erkennbarkeit durch Halbgruppen)**:** Eine Sprache $L \subseteq \Gamma^+$ wird von einer Halbgruppe S erkannt, falls ein surjektiver Homomorphismus $h : \Gamma^+ \to S$ existiert mit $L = h^{-1}(h(L))$. Entsprechendes gilt für die Erkennbarkeit durch ein Monoid M für einen Homomorphismus der Form $h : \Gamma^* \to M$.

Der syntaktische Homomorphismus einer Sprache L lässt sich stets angeben, und ist durch die syntaktische Kongruenz induziert.

Definition 1.10 (Syntaktische Kongruenz)**:** Die syntaktische Kongruenz $\sim_L$ einer Sprache $L \subseteq \Gamma^+$ ist durch folgende Äquivalenzrelation auf Wörtern $u, v \in \Gamma^+$ gegeben:

$$u \sim_L v \;:\Leftrightarrow\; [\forall p, q \in \Gamma^* : puq \in L \Leftrightarrow pvq \in L].$$

Dies ist in der Tat eine Kongruenz. Die Äquivalenzklassen von $\sim_L$ auf Γ^+ ergeben die Elemente der syntaktischen Halbgruppe $Synt(L) := \Gamma^+/\sim_L$ und ihre Operation ist durch die Konkatenation gegeben. Analoges gilt für das syntaktische Monoid für Sprachen $L \subseteq \Gamma^*$.

Die folgenden Relationen lassen sich auf generische Art definieren und werden für die spätere Betrachtung von Wörtern wichtig sein, da wir durch sie generische Faktorisierungen erhalten können.

Definition 1.11 (Green'sche Relationen)**:** Sei S eine Halbgruppe. S^1 bezeichne das Monoid, welches durch hinzufügen eines neutralen Elements entsteht $(1 \cdot s = s \cdot 1 = s \;\forall\; s \in S)$.

$$\begin{aligned} s\mathcal{R}t &:\Leftrightarrow sS^1 = tS^1 & \qquad s <_{\mathcal{R}} t &:\Leftrightarrow sS^1 \subset tS^1 \\ s\mathcal{L}t &:\Leftrightarrow S^1s = S^1t & s <_{\mathcal{L}} t &:\Leftrightarrow S^1s \subset S^1t \\ s\mathcal{J}t &:\Leftrightarrow S^1sS^1 = S^1tS^1 & s <_{\mathcal{J}} t &:\Leftrightarrow S^1sS^1 \subset S^1tS^1 \end{aligned}$$

Bemerkung 1.12: Die Anzahl der $\mathcal{R}$ bzw. $\mathcal{L}$-Klassen ist durch die Größe der Halbgruppe beschränkt, denn jede Klasse enthält mindestens ein Element und die Klassen sind eine Partition der Halbgruppe.

Definition 1.13: Ein Monoid M heißt $\mathcal{J}$, $\mathcal{R}$, $\mathcal{L}$-trivial, falls die entsprechenden Äquivalenzklassen trivial sind. Für alle $x, y \in M$ gilt also jeweils:

$$\begin{aligned} x\mathcal{J}y &\Rightarrow x = y && \text{bzw.} \\ x\mathcal{R}y &\Rightarrow x = y && \text{bzw.} \\ x\mathcal{L}y &\Rightarrow x = y \end{aligned}$$

Definition 1.14 (Erzeugtes Idempotentes)**:** Sei S eine endliche Halbgruppe und $s \in S$ ein beliebiges Element. Mit $s^\pi =: t$ bezeichnen wir das davon erzeugte Idempotente, sprich es gilt $t^2 = t$. Es existiert nach dem folgenden Lemma.

Lemma 1.2 (Existenz von Idempotenten)**:** *Sei S eine endliche Halbgruppe und $s \in S$. Es existiert ein Element $s^\pi \in S$ mit $(s^\pi)^2 = s^\pi$ wobei $\pi := |S|!$ stets genügt.*

Beweis: Wir betrachten Potenzen von mindestens $n =: |S|$ des Elements s. Durch Schubfachschluss müssen zwei Präfixe den gleichen Wert in S annehmen $s^i \cdot s^p = s^i$

mit $i < p$. Falls $i = 1$ haben wir bereits $s^\pi := s^p$ gefunden. Anderenfalls betrachten wir das Element $s^{(i \cdot p)}$ und prüfen ob es idempotent ist.

$$s^{2(ip)} = s^{ip} \cdot s^{ip} = \underbrace{s^i \cdots (s^i}_{p\text{-mal}} \cdot \underbrace{s^p) \cdots s^p}_{i\text{-mal}} = s^{ip}$$

Wir betrachten noch $s^{n!} = (s^{ip})^{n!/ip} = s^{ip}$. □

Endliche Halbgruppen S besitzen also Idempotente und wir bezeichnen mit $E(S) \subseteq S$ die Elemente, welche idempotent sind. Im Folgenden wollen wir algebraische Identitäten für Klassen von Monoiden definieren. Um die Notation zu erleichtern, schreiben wir $M \in [\![I]\!]$, falls für alle Elemente $a, b, \ldots$ aus M die Identität I gilt.

Lemma 1.3 ($\mathcal{R}$, $\mathcal{L}$, und $\mathcal{J}$-Identitäten)**:** *Sei M ein Monoid, dann gilt:*

$$\begin{aligned} M \textit{ ist } \mathcal{R}\textit{-trivial} &\Leftrightarrow M \in [\![(ab)^\pi = (ab)^\pi a]\!] && \textit{bzw.} \\ M \textit{ ist } \mathcal{L}\textit{-trivial} &\Leftrightarrow M \in [\![(cd)^\pi = d(cd)^\pi]\!] && \textit{bzw.} \\ M \textit{ ist } \mathcal{J}\textit{-trivial} &\Leftrightarrow M \in [\![(ab)^\pi a(cd)^\pi = (ab)^\pi d(cd)^\pi]\!] \end{aligned}$$

Beweis: Wir geben den Beweis für die $\mathcal{J}$-Identität an.
"⇒" Seien $a, b, c, d \in M$ bliebig. Wir betrachten das Idempotente $(ab)^\pi = (ab)^\pi(ab)^\pi$. Für einen Präfix der rechten Seite gilt $(ab)^\pi a \mathrel{\mathcal{R}} (ab)^\pi$ und wegen der $\mathcal{R}$-Trivialität sogar Gleichheit. Das Duale Argument für einen Suffix von $(cd)^\pi(cd)^\pi$ mit $\mathcal{L}$-Trivialität liefert $d(cd)^\pi = (cd)^\pi$. Somit gilt für das Produkt dieser Idempotenten:

$$(ab)^\pi(cd)^\pi = (ab)^\pi a(cd)^\pi = (ab)^\pi d(cd)^\pi.$$

"⇐" Mit $c = d = 1$ gilt die $\mathcal{R}$-Identität und mit $a = b = 1$ die $\mathcal{L}$-Identität. Sei nun $a\mathcal{J}b$, dann existieren Elemente $m_1, m_2, m_3, m_4 \in M$ mit $a = m_1 b m_2$ und $b = m_3 a m_4$.

$$\begin{aligned} a &= (m_1m_3)^\pi m_1 b m_2 (m_4 m_2)^\pi && \text{Einsetzen} \\ &= m_3(m_1m_3)^\pi m_1 b m_2 (m_4 m_2)^\pi m_4 && \mathcal{L} \text{ und } \mathcal{R} \text{ Identität} \\ &= m_3 a m_4 && \text{Abpumpen} \\ &= b \end{aligned}$$

□

Um diese Charakterisierungen auf ein allgemeineres Niveau zu heben, benötigen wir die folgende größere Klasse von Monoide.

Definition 1.15 ($\mathbb{DA}$)**:** Mit $\mathbb{DA}$ bezeichnen wir die folgende Klasse von Monoide

$$\begin{aligned} \mathbb{DA} :&= \{\, M \text{ Monoid } \mid \forall x, y \in M : (xy)^\pi x (xy)^\pi = (xy)^\pi \,\} \\ &= [\![(xy)^\pi x (xy)^\pi = (xy)^\pi]\!] \end{aligned}$$

Durch geschickte Wahl von Elementen lässt sich die folgende Teilmengenbeziehung sofort zeigen.

Lemma 1.4: *Es gilt $\mathcal{J}$-trivial $\subseteq \mathbb{DA}$*

Beweis: Sei M $\mathcal{J}$-trivial. Wir wählen für die $\mathcal{J}$-Identität folgende Elemente $x, y, (xy)^\pi \in M$ und setzen $a := x, b := y$ und $d = c := (xy)^\pi$. Betrachten wir nun die $\mathcal{J}$-Identität:

$$\begin{aligned} (xy)^\pi x((xy)^\pi(xy)^\pi))^\pi &= (xy)^\pi(xy)^\pi((xy)^\pi(xy)^\pi))^\pi \\ (xy)^\pi x(xy)^\pi &= (xy)^\pi \\ &\Rightarrow M \in \mathbb{DA}. \end{aligned}$$

□

Diese Zusammenhänge lassen sich auch auf Halbgruppen übertragen, indem man die beschriebenen Eigenschaften nur für durch Idempotente eingeschlossene, sogenannt lokalisierte, Unterstrukturen fordert. Wir definieren also den L Operator auf Klassen von Monoide.

Definition 1.16 (Lokalisierung)**:** Sei $\mathbb{V}$ eine Klasse von Monoide, dann bezeichnet

$$L\mathbb{V} = \{\, S \text{ Halbgruppe } \mid \forall e \in E(S) : eSe \in \mathbb{V} \,\}$$

die Klasse von Halbgruppen, welche durch sog. Lokalisierungen an Idempotenten aus $\mathbb{V}$ hervorgeht.

Es ist auch offensichtlich, dass stets $\mathbb{V} \subseteq L\mathbb{V}$ gilt. Außerdem übertragen sich sofort die Lemmata 1.3 und 1.4 auf die lokalisierten Klassen. Im Folgenden werden wir auch die Notation $L \in L\mathbb{V}$ für eine Sprache $L \subseteq \Gamma^+$ verwenden, falls ihre syntaktische Halbgruppe $Synt(L) \in L\mathbb{V}$ liegt.

Lemma 1.5 (Stabilisierende Idempotente)**:** *Sei S eine endliche Halbgruppe. Betrachten wir ein Produkt mit $n := |S| + 1$ Faktoren, dann existiert ein sogenanntes stabilisierendes Idempotentes $e \in E(S)$ so, dass*

$$\begin{aligned} s_1 \cdots s_n &= s_1 \cdots s_k \cdot e \cdot s_{k+1} \cdots s_n \\ &= s_1 \cdots s_k \cdot e \cdot s_{k+1} \cdots s_l \cdot e \cdot s_{l+1} \cdots s_n. \end{aligned}$$

Beweis: Betrachten wir die Präfixe dieses Produktes, dann existiert nach dem Schubfachprinzip Positionen $1 \leq k < l \leq n$ so, dass

$$s_1 \cdots s_k = s_1 \cdots s_l.$$

Und wir haben bereits ein solches Idempotentes gefunden $e := (s_k \cdots s_l)^\pi$. □

Obiges Lemma wird uns in Kapitel 4 ein wertvolles Werkzeug sein, um ein Idempotentes in Faktoren gewisser Länge sicherzustellen. Außerdem definieren wir noch die folgende $\mathcal{B}_1$-Identität. In Kapitel 4 wird sich in der Tat herausstellen, dass genau die syntaktischen Halbgruppen von Sprachen des Level 1 der Dot-Depth Hierarchie diese Identität erfüllen.[8]

Definition 1.17 ($\mathcal{B}_1$-Identität)**:** Als $\mathcal{B}_1$-Identität bezeichnen wir die Gleichung:

$$(eafb)^\pi eaf(cedf)^\pi = (eafb)^\pi edf(cedf)^\pi$$

Und eine Halbgruppe S ist in $[\![\mathcal{B}_1]\!]$, falls sie für beliebige $a, b, c, d \in S$ und $e, f \in E(S)$ erfüllt ist.

Bemerkung 1.18: Betrachten wir noch einmal die Identität für die Klasse $L\mathcal{J}$

$$L\mathcal{J} = [\![(eaeb)^\pi eae(cede)^\pi = (eaeb)^\pi ede(cede)^\pi]\!]$$

Stellen wir fest, dass diese nur ein Spezialfall von $\mathcal{B}_1$ für die Wahl von $f = e$ ist. Daher gilt: $[\![\mathcal{B}_1]\!] \subseteq L\mathcal{J}$.

$L\mathcal{J}$ wiederum erfüllt durch Wahl von $a = b = e$ die $L\mathcal{L}$-Identität: $e(cede)^\pi = ede(cede)^\pi$), bzw. durch Wahl von $c = d = e$ die $L\mathcal{R}$-Identität: $(eaeb)^\pi e = (eaeb)^\pi eae$).

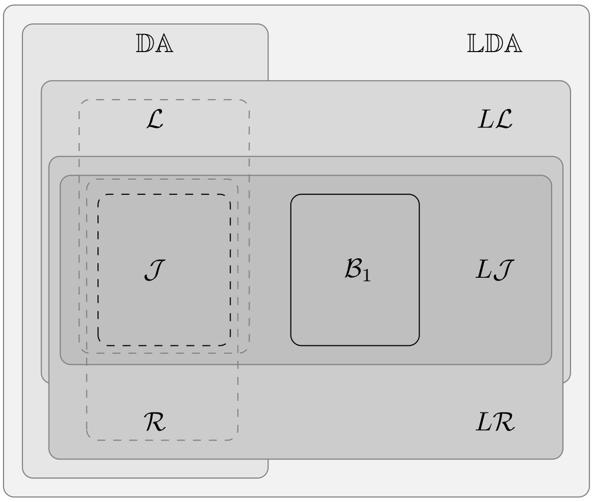


Abbildung 1.1: Inklusionseigenschaften der betrachteten algebraischen Klassen.

Ranker und Rankersprachen

Ranker sind ein natürliches und intuitives Mittel um formale Sprachen zu beschreiben. Außerdem ermöglichen sie es leicht, verschiedene kombinatorische Bedingungen zu formulieren.

Definition 1.19 (Ranker)**:** Ein Ranker ist ein Wort der Länge k über dem Alphabet R von neXt und Yesterday Anweisungen $R \subseteq \{ X_a, Y_a \mid a \in \Gamma \}$. k wird auch als Rankertiefe bezeichnet. Die Semantik eines Rankers $r \in R^*$ auf einem Wort $u \in \Gamma^*$ ist induktiv definiert. Hat r die atomare Gestalt $r = X_a$ bzw. $r = Y_a$, dann gilt:

$$\begin{aligned} X_a(u) = X_a(u,0) &= \min \{ y \in \mathbb{N} \mid u|_y = a \text{ und } y > 0 \} \\ Y_a(u) = Y_a(u,|u|+1) &= \max \{ y \in \mathbb{N} \mid u|_y = a \text{ und } y < |u|+1 \} \end{aligned}$$

Hierbei ist $\min \emptyset = \max \emptyset = undef$. Der leere Ranker soll außerdem stets definiert sein, beispielsweise $\varepsilon(u) = 0$. Hat r die zusammengesetzte Form $r = X_a r'$ bzw. $r = Y_a r'$, dann gilt:

$$\begin{aligned} X_a r'(u) &= r'(u, X_a(u)) \\ Y_a r'(u) &= r'(u, Y_a(u)). \end{aligned}$$

Die Anweisungen eines Rankers werden also von links nach rechts abgearbeitet.

Beispiel 1.1 Ranker
Sei $u = bab \in \Gamma^*$.

$$\begin{aligned} X_a(u) &= X_a(u,0) = 2 \\ X_a Y_b(u) &= X_a Y_b(u,0) = 1 \\ Y_a(u) &= Y_a(u,4) = 2 \\ \text{aber: } X_a X_a(u) &= undef. \end{aligned}$$

Definition 1.20 (Wortranker)**:** Ein Wortranker ist ein Wort der Länge k über dem Alphabet R von neXt und Yesterday Anweisungen $R \subseteq \{ X_w, Y_w \mid w \in \Gamma^+ \}$. k wird auch als Rankertiefe und die Länge des längsten Wortes in einer Anweisung von r als Rankergröße bezeichnet. Die Semantik eines Wortrankers r auf einem Wort $u \in \Gamma^*$ ist induktiv definiert. Hat r die atomare Gestalt $r = X_w$ bzw. $r = Y_w$, dann gilt:

$$\begin{aligned} X_w(u) = X_w(u,0) &= \min \left\{ y \in \mathbb{N} \;\middle|\; u|_{[y, y+|w|-1]} = w \text{ und } y > 0 \right\} \\ Y_w(u) = Y_w(u,|u|+1) &= \max \left\{ y \in \mathbb{N} \;\middle|\; u|_{[y-|w|+1, y]} = w \text{ und } y < |u|+1 \right\} \end{aligned}$$

Hierbei ist $\min \emptyset = \max \emptyset = undef$. Hat r die zusammengesetzte Form $r = X_w r'$ bzw. $r = Y_w r'$, dann gilt:

$$\begin{aligned} X_w r'(u) &= r'(u, X_w(u)) \\ Y_w r'(u) &= r'(u, Y_w(u)). \end{aligned}$$

Die Anweisungen eines Wortrankers werden also von links nach rechts abgearbeitet. Insbesondere ist es für einen Wortranker X_w durch abarbeiten von Suffixen möglich exakt beliebige Positionen $1 \leq i \leq |w| =: n$ innerhalb dieses Faktors zu definieren. Hierfür verwenden wir zur Notation das folgende Makro:

$$X_w^i := X_w X_{w_{[2:n]}} \dots X_{w_{[i:n]}}$$

Beispiel 1.2 Wortranker
Sei $u = bab \in \Gamma^*$.

$$\begin{aligned} X_{ab}(u) &= X_{ab}(u,0) = 2 \\ X_{ab}Y_{ba}(u) &= X_{ab}Y_{ba}(u,0) = 2 \\ Y_{ab}(u) &= Y_{ab}(u,4) = 3 \\ X_{ab}^2(u) &= X_{ab}X_b(u) = 3 \\ Y_{ab}^2(u) &= Y_{ab}Y_a(u) = 2 \\ \text{aber: } X_{ab}X_{ab}(u) &= undef. \end{aligned}$$

Definition 1.21 (Rankersprache)**:** Die von einem Ranker bzw. Wortranker r definierte Sprache $L(r)$ ist die Menge von Wörtern aus Γ^*, auf denen r definiert ist.

$$L(r) = \{\, u \in \Gamma^* \;\mid\; r(u) \text{ ist definiert} \,\}\,.$$

Diese Definition lässt sich auf natürliche Art auf endliche boolsche Kombinationen von Rankern, durch Kombination ihrer Sprachen, erweitern.

2 Algebraische Werkzeuge für formale Sprachen

Dieses Kapitel legt wichtige Grundlagen für die kombinatorischen Beweise der Kapitel 3, 4 und 5.

Lemma 2.1 (Abstiegslemma $\mathbb{DA}$)**:** *Sei $u, v, a \in M \in \mathbb{DA}$. Falls $u\mathcal{R}uv$ und $v \in MaM$, dann ist auch $u\mathcal{R}uva$.*

Beweis: Siehe [3] Lemma 2. $\square$

Lemma 2.2 (Abstiegslemma $\mathbb{LDA}$)**:** *Sei $S \in \mathbb{LDA}$ eine Halbgruppe mit $e^2 = e, ue = u, uae = ua$ und $u\mathcal{R}ua$, dann gilt $u\mathcal{R}uaa$.*

Beweis: Nach Definition gilt für Präfixe immer $uaa \leq_{\mathcal{R}} u$. Außerdem existiert mit $u = ue \in uS = uaS$ in solchen Halbgruppen stets ein $\bar{a}$ so, dass $u = ua\bar{a}$.

Es genügt also $u \leq_{\mathcal{R}} uaa$ zu zeigen. Sei also $\bar{a}$ so gewählt, dann gilt:

$$\begin{aligned}
u &= ue && e \text{ stabilisiert } u \\
&= (ua\bar{a})e && u\mathcal{R}ua \\
&= ueae\bar{a}e && e \text{ stabilisiert } ua \text{ und } u \\
&= u(eae\bar{a}e)^{\pi} && \text{aufpumpen} \\
&= u(eae\bar{a}e)^{\pi}eae(eae\bar{a}e)^{\pi} && \text{DA-Gleichung der Lokalisierung} \\
&= uaeae(eae\bar{a}e)^{\pi} && \text{abpumpen} \\
&= uaa \cdot \underbrace{e(eae\bar{a}e)^{\pi}}_{=:z}
\end{aligned}$$

Damit gilt $uS = uaazS \subseteq uaaS$ und die gewünschte Relation ist erfüllt. $\square$

Die Existenzaussage des Lemmas lässt sich graphisch schön darstellen:

$$\begin{tikzcd}
u \arrow[loop above, "\cdot e"] \arrow[r, bend left, "\cdot a"] & ua \arrow[loop above, "\cdot e"] \arrow[l, bend left, "\cdot \bar{a}"] \arrow[r, bend left, "\cdot a"] & uaa \arrow[ll, dashed, bend left, "\exists \cdot \overline{aa}"]
\end{tikzcd}$$

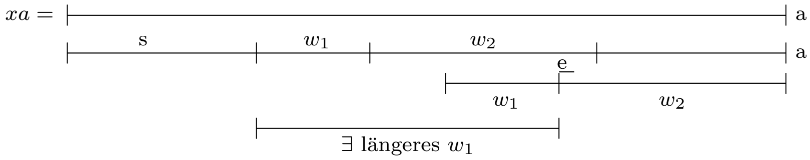
Dies lässt sich auch auf Faktoren großer Länge erweitern.

Lemma 2.3 (Faktor-Abstiegslemma für $\mathbb{LDA}$)**:** *Sei $u, x \in \Gamma^+, a \in \Gamma, m \in \mathbb{N}$ mit $|x| \geq m > |S|$, wobei $h : \Gamma^+ \to S \in \mathbb{LDA}$ der erkennende Homomorphismus ist. Sei außerdem $last_m(xa) \in alph_m(x)$. Dann gilt:*

$$u\mathcal{R}ux \Rightarrow u\mathcal{R}uxa.$$

Beweis: Sei $w := last_m(xa), x := swt, |w| = m > |M|$. Dann existiert eine Faktorisierung von $w = w_1 w_2 a$ mit $|w_1|$ maximal, die von einem Idempotenten $e \in \Gamma^+$ stabilisiert wird, d.h. $h(w_1) = h(w_1)e$ (vgl. Lemma 1.5). Da x und xa nur gleiche Faktoren haben, muss der Faktor w mindestens zweimal in xa vorkommen. Wir betrachten nun das erste und letzte Vorkommen von w in xa.

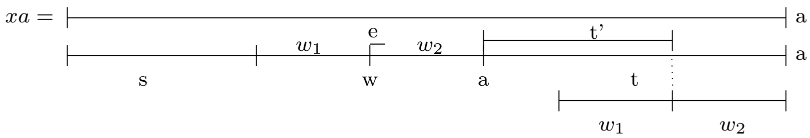
Angenommen der Anfang vom letzen w_2 liegt vor dem Ende des ersten w_2, insbesondere liegt also das letzte w_1 ganz innerhalb des ersten w_2.



Wie das Diagram zeigt, lässt sich dann ein größeres w_1 mit den gewünschten Eigenschaften wählen, was im Widerspruch zur Maximalität von $|w_1|$ steht.

Seien also die beiden w_2-Faktoren überlappungsfrei, d.h.

$$x = s(w_1 w_2 a)t = s(w_1 w_2 a)t' w_2$$



Mit folgenden Festlegungen lässt sich das Abstiegslemma anwenden.

$$u' := h(usw_1) \qquad\qquad a' := h(w_2 a t')$$

Zur besseren Lesbarkeit werden ab sofort Wörter mit ihrem Bild unter h identifiziert. Da w_1 Suffix von t' ist, wird auch a' von e stabilisiert. Mit

$$\begin{aligned}
u' &= usw_1 \\
&\leq_{\mathcal{R}} u \\
&\leq_{\mathcal{R}} ux && \text{(nach Voraussetzung } u\mathcal{R}ux\text{)} \\
&= (usw_1)(w_2 a t')w_2 \\
&\leq_{\mathcal{R}} (usw_1)w_2 a t' \\
&= u'a' \\
&\leq_{\mathcal{R}} u'
\end{aligned}$$

ist die letzte Voraussetzung $u'\mathcal{R}u'a'$ für das Abstiegslemma erfüllt. Außerdem zeigt die Rechnung $u\mathcal{R}u'$. Somit gilt

$$u \; \mathcal{R} \; u' \; \mathcal{R} \; u'a' \; \mathcal{R} \; u'a'a'$$

und wegen

$$\begin{aligned} h(uxat') &= h(u(sw_1w_2at'w_2)at') \\ &= h((usw_1)(w_2at')(w_2at')) \\ &= u'a'a' \end{aligned}$$

folgt $uxa\mathcal{R}u$. □

3 Klimas Beweisidee

Dieses Kapitel beschäftigt sich nun mit L_1, dem Level 1 der Straubing-Thérien Hierarchie. Wie wir bereits in Bemerkung 1.4 gesehen haben, werden diese Sprachen auch picewise-testable genannt. Es ist ein wohlbekanntes Ergebnis, dass die syntaktischen Monoide von Sprachen dieser Klasse, genau die $\mathcal{J}$-trivialen Monoide sind.[15] 2009 wurde für dieses Ergebnis ein einfacher kombinatorischer Beweis gegeben, welchen wir hier in verallgemeinerter Notation und unter Verwendung unserer algebraischen Werkzeuge des vorherigen Kapitels wiedergeben möchten.

Definition 3.1: Als $\mathcal{R}$-Faktorisierung (bzw. $\mathcal{L}$-Faktorisierung) von einem Wort $u \in \Gamma^*$ bezeichnen wir die Menge $R(u)$ (bzw. $L(u)$) an Indexpositionen $1 \leq i \leq |u|$ von Zeichen a_j in u, die eine Faktorisierung $u = u_0 a_1 u_1 a_2 u_2 \dots u_{k-1} a_k u_k$ so vermittelt, dass:

$$h(u_0 \dots a_j) \mathrel{\mathcal{R}} h(u_0 \dots a_j u_j) \gneq_{\mathcal{R}} h(u_0 a_j u_j a_{j+1}), \text{ wobei } 1 \mathrel{\mathcal{R}} h(u_0)$$

bzw.

$$h(a_{j+1} \dots u_k) \lneq_{\mathcal{L}} h(u_j a_j \dots u_k) \mathrel{\mathcal{L}} h(a_j \dots u_k), \text{ wobei } h(u_k) \mathrel{\mathcal{L}} 1$$

Bemerkung 3.2 (Anwendung des Abstiegslamma)**:** Diese Form von Faktorisierung ist für ein Wort bei gegebener Sprache L eindeutig. Erfüllt $Synt(L)$ die $\mathcal{J}$-Identität, insbesondere $Synt(L) \in \mathbb{DA}$, dann sind nach dem Abstiegslemma (2.1) die Buchstaben a_i der Abstiegsstellen nicht in den davor liegenden Faktoren u_{i-1} enthalten. Dadurch sind die Abstiegsstellen durch Ranker definierbar. Beispielsweise befindet sich die Abstiegsstelle a_i auf u an Position $X_{a_1} \dots X_{a_i}(u)$. Duales gilt für die $\mathcal{L}$-Faktorisierung mit Y-Rankern.

Der Beweis des nun folgenden Satzes bildet die Grundlage für unseren neuen kombinatorischen Beweis von Knasts Theorem (4.1).

Satz 3.1 (Level 1 der Straubing-Thérien Hierarchie)**:** *Sei $L \subseteq \Gamma^*$ eine reguläre Sprache. Die folgenden Charakterisierungen sind äquivalent:*

1. *L ist endliche boolsche Kombination von Ranker-Sprachen,*
   d.h. $L = \mathop{\mathbb{B}}_{r \in R[X]} L(r) = \mathop{\mathbb{B}}_{r \in R[Y]} L(r)$

2. *L ist in $\mathbb{B}\Sigma_1[<]$ definierbar*

3. *$L = \mathbb{B}(\Gamma^* a_1 \Gamma^* \dots a_n \Gamma^*, a_i \in \Gamma)$*

4. *$Synt(L)$ ist $\mathcal{J}$-trivial*

Hier wird nun eine mehr algebraische Formulierung dieses Beweises gegeben, wobei die Ranker Charakterisierung vermutlich neu ist. Der Übersicht wegen ist der Beweis in vier Lemmata aufgeteilt.

Beweis:
$1 \Rightarrow 2$: Lemma 3.2
$2 \Rightarrow 3$: Σ_1-Normalform (Lemma 3.3)
$3 \Rightarrow 4$: Lemma 3.4
$4 \Rightarrow 1$: Kombinatorischer Beweis (Lemma 3.5) □

Lemma 3.2: *Sei r ein Ranker mit ausschließlich neXt-Anweisungen, dann existiert eine Formel $\varphi_r \in \Sigma_1[<]$ mit $L(r) = L(\varphi_r)$.*

Beweis: Ohne Einschränkung haben die Ranker nur neXt-Anweisungen $r = X_{a_1} \dots X_{a_n}$. Hierfür lässt sich einfach eine Formel $\varphi_r \in \Sigma_1[<]$ angeben.

$$\varphi_r := \exists x_1, \dots, x_n \bigwedge_{i=1}^{n} \lambda(x_i) = a_i \ \wedge \ \bigwedge_{i=2}^{n} <(x_{i-1}, x_i)$$

Falls $r(w)$ $definiert$, sind insbesondere alle Präfixe von r auf w definiert. Diese Positionen erfüllen das Geforderte: $w \models \varphi_r$. Falls $w \models \varphi_r$, gibt es Positionen mit korrekter Beschriftung, welche linear angeordnet sind. Wähle von den möglichen erfüllenden Belegungen der x_i induktiv die am weitesten links liegende Positionen. Diese sind genau die Positionen, an welchen die Präfixe von r definiert sind. Also ist $r(w)$ $definiert$. □

Lemma 3.3: *Eine Sprache $L(\varphi) \subseteq \Gamma^*$ mit $\varphi \in \mathbb{B}\Sigma_1[<]$ hat Level 1 der Straubing-Thérien Hierarchie, sprich die Form $\mathbb{B}(\Gamma^* a_1 \Gamma^* \dots a_n \Gamma^*, a_i \in \Gamma)$.*

Beweis: Da beide Charakterisierungen endliche boolsche Kombinationen erlauben, genügt es für eine Formel $\psi \in \Sigma_1[<]$ eine boolsche Kombination von Sprachen der gewünschten From anzugeben. Nach Lemma 1.1 können wir $\psi = \bigvee_n \varphi_n$ annehmen, wobei die Klauseln φ_n nur positive Prädikate verwenden.

$$\varphi_n = \exists x_1, \dots, x_{k_n} : \bigwedge_{i \in I_n} \lambda(x_i) = a_{n_i} \wedge \bigwedge_{(j_1, j_2) \in J_n} <(x_{j_1}, x_{j_2})$$

Wir geben nun die gewünschte Sprachcharakterisierung K_n für so eine Klausel an. Die Relation $<$ definiert auf der Menge der Variablen X einen gerichteten Graph $G = (X, <)$. Falls unterschiedliche Beschriftungen λ für die gleiche Variable gefordert sind oder dieser Graph einen Kreis enthält, setzen wir $K_n := \emptyset = \overline{\Gamma^*}$ und haben damit die gewünschte Sprache angegeben. Anderenfalls gibt es eine endliche Anzahl gerichtete Pfade γ_l maximaler Länge. Diese Pfade sind wieder eine lineare Anordnung unserer Variablen. Mit $a_{l_{i'}} \in \Gamma$ bezeichnen wir das Label von der Variable i' auf dem Pfad γ_l. x_{l_m} sei die letzte Variable auf diesem Pfad. Wir setzen die Sprache K_n folglich auf:

$$K_n := \bigcap_{\text{Pfade } \gamma_l} (\Gamma^* a_{l_1} \dots a_{l_m} \Gamma^*).$$

□

Lemma 3.4: *Das syntaktische Monoid von Sprachen der Form* $L = \mathbb{B}(\Gamma^* a_1 \Gamma^* \dots a_n \Gamma^*, a_i \in \Gamma)$ *ist* $\mathcal{J}$*-trivial.*

Die Idee des folgenden Beweises entstammt einem Ehrenfeucht-Fraïsse Spiel. Jedoch ist er in der folgenden Notation auch ohne Grundlagen dieser Beweismethode nachvollziehbar.

Beweis: Sei h der syntaktische Homomorphismus, der L erkennt. Es gilt nun zu zeigen, dass für beliebige Elemente $a', b', c', d' \in Synt(L)$ die $\mathcal{J}$-Identität gilt (1.3).

$$(a'b')^\pi a'(c'd')^\pi = (a'b')^\pi d'(c'd')^\pi \tag{3.1}$$

Wir wählen nun jeweils beliebige Urbilder dieser Elemente $a, b, c, d \in \Gamma^*$ die im Folgenden aber fest gehalten werden.

$$a' = h(a)\ ,\ b' = h(b)\ ,\ c' = h(c)\ ,\ d' = h(d)$$

Da die Urbilder der beiden Idempotenten beliebig groß gewählt werden können, werden wir uns dies zu Nutze machen. Uns genügt bereits ein n, welches größer ist, als die größte Anzahl der a_i welche in einem Teilausdruck von L vorkommen. Wir betrachten nun die folgenden zusammengesetzten Wörter $u, v \in \Gamma^*$,

$$u := (ab)^n a(cd)^n \qquad v := (ab)^n d(cd)^n$$

Im Folgenden sind nun $p, q \in \Gamma^*$ beliebig. Wir zeigen nun, dass $puq \in K$ genau dann, wenn auch $pvq \in K$, wobei K ein Teilausdruck der boolschen Kombinationen von L ist. Um das zu gewährleisten, geben wir eine Taktik an, mit der für beliebige, von K forderbaren, Subwörtern $a_1 \dots a_n$ diese auch in pvq enthalten sind.

1. Wähle exakt die gleichen Positionen für p oder q auf pvq.

2. Falls das letzte a unbelegt ist, wähle exakt die gleichen Positionen auf v und wir sind fertig.

3. Falls das letzte a belegt ist, können davor maximal noch $n-1$ Positionen in $(ab)^n$ belegt sein.

   a) Im schlimmsten Fall sind nur a's belegt. Selbst dann lassen sich auf v einfach die n a-Positionen hintereinander belegen.

   b) Falls auch mindestens ein b belegt wurde, ist sogar ein ganzer ab bzw. ba Faktor unbelegt, wodurch auf v genügend a-Positionen zur Verfügung stehen, wovon wir jeweils die kleinsten Positionen wählen.

   Wähle gegebenenfalls restliche Positionen in $(cd)^n$ auf v identlisch.

Wendet man zusätzlich die duale Strategie für pvq an, erhält man:

$$\begin{aligned} puq \in K &\Leftrightarrow pvq \in K \\ puq \in L &\Leftrightarrow pvq \in L \\ &\Rightarrow u \sim_L v \\ &\Rightarrow h((ab)^n a(cd)^n) = h((ab)^n d(cd)^n). \end{aligned}$$

Und die $\mathcal{J}$-Gleichung gilt für beliebige Elemente $a', b', c', d' \in Synt(L)$. □

Das folgende kombinatorische Lemma ist zentrale Grundlage dieser Arbeit.

Lemma 3.5: *Sei $L \subseteq \Gamma^*$ und $Synt(L)$ $\mathcal{J}$-trivial, dann ist L endliche boolsche Kombination von Rankern, die ausschließlich neXt-Anweisungen benützen.*

Beweis: Betrachte folgende Relation auf Wörtern $u, v \in \Gamma^*$

$$u \sim_k v \;:\Leftrightarrow\; \forall r = X_{a_1} \dots X_{a_k}, a_i \in \Gamma : r(u) \text{ def. gdw. } r(v) \text{ def.}$$

Dies ist offensichtlich eine Äquivalenzrelation. Sei $h : \Gamma^* \to Synt(L) =: M$ der erkennende Homomorphismus, wobei M die $\mathcal{J}$-Gleichung erfülle. Wir werden zeigen, dass diese Relation eine Verfeinerung der syntaktischen Kongruenz ist $\sim_k \subseteq \sim_L$. Da $L = \bigcup_u [u]_{\sim_L}$ eine Vereinigung solcher Klassen ist, folgt die Definierbarkeit durch Ranker.

Sei also $u \sim_k v$. Betrachte die Indexmengen $blau := R(u)$ mit den Abstiegsstellen der $\mathcal{R}$-Faktorisierung von $u = u_0 a_1 u_1 \dots u_{k-1} a_k u_k$ und $rot := L(v)$ mit den Abstiegsstellen der $\mathcal{L}$-Faktorisierung von $v = v_l b_l \dots v_1 b_1 v_0$. Durch das Abstiegslemma (Lemma 2.1) ist sichergestellt, dass $a_{i+1} \notin alph(u_i)$ und $b_{i+1} \notin alph(v_i)$. Außerdem wird so Folgendes sichergestellt:

$$\begin{aligned} h(u) \;\mathcal{R}\; h(u|_{blau}) \;\mathcal{R}\; h(A_0^* a_1 A_1^* \dots a_k A_k^*) &\text{ mit } A_i = alph(u_i) \\ h(v) \;\mathcal{L}\; h(v|_{rot}) \;\mathcal{L}\; h(B_l^* b_l \dots B_1^* b_1 B_0^*) &\text{ mit } B_i = alph(v_i). \end{aligned}$$

Da $x\mathcal{R}y \Rightarrow x\mathcal{J}y$ (entsprechend für $\mathcal{L}$) und M $\mathcal{J}$-trivial ist, gilt sogar Gleichheit.

$$\begin{aligned} h(u) = h(u|_{blau}) = h(A_0^* a_1 A_1^* \dots a_k A_k^*) &\text{ mit } A_i = alph(u_i) \\ h(v) = h(v|_{rot}) = h(B_l^* b_l \dots B_1^* b_1 B_0^*) &\text{ mit } B_i = alph(v_i) \end{aligned}$$

Da $u \sim_k v$, lassen sich die blauen Positionen auf v übertragen, wobei das am weitesten links liegende Vorkommen gewählt wird. Die duale Konstruktion liefert die übertragene rote Indexmenge auf u.

$$\begin{aligned} \overline{blau} &:= \{\, X_{a_1} \dots X_{a_i}(v) \;\mid\; 1 \leq i \leq k \,\} \\ \overline{rot} &:= \{\, Y_{b_1} \dots Y_{b_i}(u) \;\mid\; 1 \leq i \leq l \,\} \end{aligned}$$

Essentiell sind nun die reduzierten Wörter $\bar{u} := u\big|_{blau \cup \overline{rot}}$ und $\bar{v} := v\big|_{\overline{blau} \cup rot}$, die nach Konstruktion bereits $alph(\bar{u}) = alph(\bar{v})$ erfüllen und jeweils in ihren blauen und in ihren roten Indizes die richtige Anordnung besitzen. Außerdem sind $|\bar{u}|$ und $|\bar{v}|$ durch $2|M|-2$ begrenzt.

Überraschenderweise gilt sogar $\bar{u}=\bar{v}$. Hierfür betrachtet man, ähnlich dem Boubblesort Algorithmus, alle Paare an Zeichenpositionen a_i, b_j, um auf die lineare Anordnung auf beiden Wörtern zu schließen.

Fall $<$:

$$
\begin{aligned}
&X_{a_1}\dots X_{a_i}(\bar{u}) < Y_{b_1}\dots Y_{b_j}(\bar{u})\\
\Leftrightarrow &X_{a_1}\dots X_{a_i}X_{b_j}\dots X_{b_1}(u) = def.\\
\Leftrightarrow &X_{a_1}\dots X_{a_i}X_{b_j}\dots X_{b_1}(v) = def.\\
\Leftrightarrow &X_{a_1}\dots X_{a_i} < Y_{b_1}\dots Y_{b_j}(\bar{v})
\end{aligned}
$$

Fall $\leq$:

$$
\begin{aligned}
&X_{a_1}\dots X_{a_i}(\bar{u}) \leq Y_{b_1}\dots Y_{b_j}(\bar{u})\\
\Leftrightarrow &X_{a_1}\dots X_{a_{i-1}}X_{a_i=b_j}X_{b_{j-1}}\dots X_{b_1}(u) = def.\\
\Leftrightarrow &X_{a_1}\dots X_{a_{i-1}}X_{a_i=b_j}X_{b_{j-1}}\dots X_{b_1}(v) = def.\\
\Leftrightarrow &X_{a_1}\dots X_{a_i} \leq Y_{b_1}\dots Y_{b_j}(\bar{v})
\end{aligned}
$$

Da alle relativen Positionen von Zeichen a_i und b_j in $\bar{u}$ und $\bar{v}$ gleich sind, haben beide Wörter die gleiche lineare Anordnung $\bar{u}=\bar{v}$. Es gilt:

$$
\begin{aligned}
h(u) = h(\bar{u}) = h(\bar{v}) = h(v)\\
\Rightarrow u \sim_L v\\
\Rightarrow L = \bigcup_{u\in L}[u]_{\sim_L} = \bigcup_{u\in L}[u]_{\sim_k}.
\end{aligned}
$$

Um dies sicherzustellen haben wir nur maximal $k := 2|M|-2$ neXt-Anweisungen benötigt. $\square$

4 Knasts Theorem

In diesem Kapitel wollen wir einen kombinatorischen Beweis für Knasts Theorem geben. Entscheident wird Lemma 4.5 sein, das den Zusammenhang von der algebraischen Charakterisierung zu Wortrankern herstellt. Wie in Bemerkung 1.18 beschrieben, sind Halbgruppen, welche die $\mathcal{B}_1$-Identität erfüllen, in $L\mathcal{J}$ und damit insbesondere in $\mathbb{LDA}$. Wir werden wiederum die generischen $\mathcal{R}$ und $\mathcal{L}$-Faktorisierungen betrachten. Werden um die Abstiegsstellen nun Intervalle ausreichender Länge betrachtet, existieren nach Lemma 1.5 stabilisierende Idempotente. Damit sind die Voraussetzungen für das Faktor-Abstiegslemma für $\mathbb{LDA}$ (Lemma 2.3) erfüllt und wir können mit Wortrankern dieser Faktoren wiederum kombinatorisch argumentieren.

Satz 4.1 (Dot-Depth-1)**:** *Die folgenden Charakterisierungen sind äquivalent:*

1. $L = \underset{r\in R[X_w]}{\mathbb{B}}(L(r) \cap min_r\Gamma^* max_r) = \underset{r\in R[Y_w]}{\mathbb{B}}(L(r) \cap min_r\Gamma^* max_r)$

2. *L ist in* $\mathbb{B}\Sigma_1[<, +1, \min, \max]$ *definierbar*

3. $L = \mathbb{B}(w_0\Gamma^* w_1 \dots \Gamma^* w_k, w_i \in \Gamma^+)$

4. $Synt(L) \in [\![\mathcal{B}_1]\!]$

Beweis: Der Übersicht wegen ist der Beweis in 4 Lammata aufgeteilt.
1 ⇒ 2: Lemma 4.2
2 ⇒ 3: Σ_1-Normalform (Lemma 4.3)
3 ⇒ 4: Lemma 4.4
4 ⇒ 1: Kombinatorischer Beweis (Lemma 4.5) □

Lemma 4.2: *Sei* $L = \underset{r\in R[X_w]}{\mathbb{B}}(L(r) \cap min_r\Gamma^* max_r)$*, dann existiert eine Formel* $\varphi \in \mathbb{B}\Sigma_1[<, +1, \min, \max]$ *so, dass* $L = L(\varphi)$*.*

Beweis: Da in beiden Charakterisierungen endliche boolsche Kombinationen erlaubt sind, genügt es wiederum für einen Wortranker eine Formel in $\varphi_r \in \mathbb{B}\Sigma_1[y, +1, \min, \max]$ anzugeben.

Betrachte ein $L = L(r = X_{w_1}...X_{w_k}) \cap min_r \Gamma^* max_r$. Wir setzen

$$\begin{aligned}\varphi_r := \exists x_{i,j} & \bigwedge_{i,j} \lambda(x_{i,j}) = (w_i)_j \ \wedge \\ & \bigwedge_{i} +1(x_{i,1}, x_{i,2}) \wedge \ldots \wedge +1(x_{i,|w_i|-1}, x_{i,|w_i|}) \ \wedge \\ & \bigwedge_{1 \leq i < k} < (x_{i,1}, x_{i+1,1}) \ \wedge \min(min_r) \wedge \max(max_r).\end{aligned}$$

Dies modeliert exakt die Definition von Wortrankern und es gilt $u \in L \Leftrightarrow u \in L(\varphi_r)$. □

Lemma 4.3: *Sei $L = L(\varphi) \subseteq \Gamma^+$ mit $\varphi \in \mathbb{B}\Sigma_1[<, +1, \min, \max]$ dann ist L innerhalb von Level 1 der Dot-Depth-Hierarchie; hat also die Form $L = \mathbb{B}(w_0\Gamma^* w_1 \ldots \Gamma^* w_k)$ mit $w_i \in \Gamma^+$.*

Beweis: Da in beiden Charakterisierungen endliche boolsche Kombinationen erlaubt sind, genügt es für eine Formel $\psi \in \Sigma_1[<, +1, \min, \max]$ eine Kombination von Sprachen der gewünschten Form anzugeben. Durch Lemma 1.1 können wir $\psi = \bigvee_n \varphi_n$ annehmen, wobei die Klauseln φ_n nur positive Prädikate besitzen. Wir konstruieren zuerst äquivalente Formeln $\tilde{\varphi}_n$. Seien dazu $x_1, \ldots, x_k$, sowie $y_1, \ldots, y_l$ existenziell gebundene Variablen, wobei $+1(x_i, x_{i+1})$ für $1 \leq i \leq k-1$ und $+1(y_i, y_{i+1})$ für $1 \leq i \leq l-1$ in φ_n enthalten sind, wobei k, l maximal. Ist außerdem das Prädikat $< (x_i, y_j)$ Teil der Formel φ_n für ein $j < i$ mit $1 \neq j$ und $k \neq i$, also

$$\varphi_n =< (x_i, y_j) \wedge \varphi'_n$$

so ersetzen wir diese Klausel durch die folgenden Klauseln

$$(\varphi'_n|_{y_1 = x_{i-j+1}}) \vee (\varphi'_n|_{y_1 = x_{i-j+2}}) \vee \ldots \vee (\varphi'_n|_{y_1 = x_k}) \vee (< (x_k, y_1) \wedge \varphi'_n)$$

Hierbei bezeichnen $|_{y_1 = x}$ jeweils nur textuelle Ersetzungen der Variable x durch Variable y_1. Diesen Prozess setzen wir fort, bis er auch auf den neuen Klauseln nicht mehr angewendet werden kann und erhalten die Formel $\tilde{\varphi}_n$ nach endlich vielen solchen Ersetzungen. Am Ende enthält also $\tilde{\varphi}_n$ keine Klauseln mit Sitationen $x_i < y_j$ mit $1 \neq j$ und $k \neq i$.

Hiervon können wir analog zu Lemma 3.3 nun Sprachen der Form $w_0\Gamma^* w_1 \ldots w_m$ für jeden Pfad im Graphen ablesen, wobei w_0 und w_k die Beschriftungen von min und max-Prädikaten sind, die sich eventuell an Pfadbeginn und Pfadende befinden. Der Schnitt über diese Sprachen stimmt mit der von einer Klausel akzeptierten Sprache überein. □

Lemma 4.4: *Die syntaktische Halbgruppe $Synt(L)$ von Sprachen der Form $L = \mathbb{B}(w_0\Gamma^* w_1 \ldots \Gamma^* w_k, w_i \in \Gamma^+)$ erfüllt die $\mathcal{B}_1$-Identität.*

Beweis: Sei $h : \Gamma^+ \to Synt(L) =: S$ der syntaktische Homomorphismus, der L erkennt. Betrachte die $\mathcal{B}_1$-Identität (1.17), die für alle Elemente $a', b', c', d' \in S$ und Idempotente $e', f' \in E(S)$ gelten muss.

$$(e'a'f'b')^\pi e'a'f'(c'e'd'f')^\pi = (e'a'f'b')^\pi e'd'f'(c'e'd'f')^\pi$$

Wir wählen nun jeweils beliebige Urbilder $a, b, c, d, e, f \in \Gamma^+$ dieser Elemente, die im Folgenden aber fest gehalten werden. Sie existieren, da der syntaktische Homomorphismus surjektiv ist.

$$a' = h(a),\ b' = h(b),\ c' = h(c),\ d' = h(d),\ e' = h(e),\ f' = h(f)$$

Da die Urbilder der Idempotenten ($e' = e' \cdot e'$) beliebig groß gewählt werden können, werden wir uns dies wieder zu Nutze machen. Uns genügt ein m, welches größer ist als das Maximum von maximaler Länge eines Wortes w_i und Anzahl von Subfaktoren in einem Teilausdruck von L. Wir betrachten als Grundlage nun die folgenden Wörter mit $u, v \in \Gamma^+, p, q \in \Gamma^*$.

$$p\,\underbrace{(e^m a f^m b)^m e^m a f^m (ce^m df^m)^m}_{=:u}\,q \qquad p\,\underbrace{(e^m a f^m b)^m e^m df^m (ce^m df^m)^m}_{=:v}\,q$$

Wir zeigen nun, dass $puq \in K$ genau dann, wenn auch $pvq \in K$, wobei K ein beliebiger Teilausdruck aus der boolschen Kombination für L ist. Um dies zu gewährleisten, geben wir eine Taktik an, mit der beliebige, von K forderbare, Subfaktoren $w_0 \dots w_k$ auch auf pvq gefunden werden können.

1. Wähle w_0 und w_k identisch. Falls p und q kürzer sind, genügt sicherlich e^m bzw. f^m um einen gemeinsamen Präfix und Suffix zu gewärleisten.

2. Wähle exakt die gleichen Positionen für Faktoren die auf p oder q liegen.

3. Falls das komplette letzte a unbedeckt ist, wähle die gleichen Positionen auf v und wir sind fertig.

4. Sei das letzte a bedeckt. Mit Faktoren der Länge maximal m lässt sich nicht der ganze Faktor $e^m a f^m$ mit einem w_i geschlossen überdecken. Daher genügt es bereits sicherzustellen, dass diese in genügender Anzahl und in richtiger Reihenfolge vorkommen.

   a) Falls kein $f^m be^m$-Faktor mit einem w_i belegt ist, könnten nicht mehr als $m-1$ $e^m a f^m$-Faktoren belegt sein. Diese kommen auch in v in der richtigen Anordnung vor. Diese wählen wir.

   b) Falls mindestens ein $f^m be^m$-Faktor mit einem w_i überdeckt ist, können sogar nicht mehr als $n-2$ $e^m a f^m$-Faktoren überdeckt sein. Auch diese Situation erfüllt bereits der Anfangsteil von v. Wir wählen einfach hierin von links nach rechts die jeweils kleinst möglichen Positionen.

   Wähle die restlichen Positionen in $(ce^m df^m)$ auf v identisch.

Wendet man zusätzlich die duale Strategie für pvq an, erhält man:

$$\begin{aligned} puq \in K &\Leftrightarrow pvq \in K \\ puq \in L &\Leftrightarrow pvq \in L \\ \Rightarrow \quad & u \sim_L v \\ \Rightarrow \quad & h(u) = h(v). \end{aligned}$$

Wodurch die $\mathcal{B}_1$-Identität für beliebige Elemente $a', b', c', d' \in S$ und $e', f' \in E(S)$ gilt. □

Lemma 4.5: *Sei L regulär und die syntaktische Halbgruppe $Synt(L)$ erfülle die $\mathcal{B}_1$-Identität, dann ist* $L = \underset{r \in R[X_w]}{\mathbb{B}} (L(r) \cap min_r \Gamma^* max_r)$

Beweis: Betrachte folgende Relation auf Wörtern $u, v \in \Gamma^+$

$$\begin{aligned} u \sim_{(l,n)} v :\Leftrightarrow & first_l(u) = first_l(v) \wedge \\ & last_l(u) = last_l(v) \wedge \\ & \forall r = X_{w_1} \dots X_{w_n} : r(u) \; def. \Leftrightarrow r(v) \; def. \quad \text{mit } 1 \leq |w_i| \leq l \end{aligned}$$

Dies ist eine Äquivalenzrelation. Sei $h : \Gamma^+ \to Synt(L) =: S$ der erkennende Homomorphismus, wobei die Halbgruppe S die $\mathcal{B}_1$-Gleichung erfülle. Wir werden analog zum Beweis von Satz 3.1 zeigen, dass diese Relation eine Verfeinerung der syntaktischen Kongruenz ist $\sim_{(l,n)} \subseteq \sim_L$.

Sei also $u \sim_{(l,n)} v$, wobei l und n, die nur von $|S|$ abhängen werden, noch festzulegen sind. Betrachte wiederum die $\mathcal{R}$-Faktorisierung von u und die $\mathcal{L}$-Faktorisierung von v. Bilde nun um jede Position aus $i \in R(u)$ Intervalle $I_i = [i - |S|, i + |S|]$ der Länge $l'' := 2|S| + 1$, um sicherzustellen dass stabilisierende Idempotente (Lemma 1.5) für den Präfix $u|_{[1,i]}$ und den Suffix $u|_{[i,|u|]}$ im Intervall I_i existieren. Wir definieren außerdem Intervalle $I_\alpha := [1 : \min\{|u|, |S| + 1\}]$ für Wortanfang und $I_\omega := [\max 1, |u| - |S|, |u|]$ für Wortende. Diese Intervalle auf u können sich überlappen. Vereinigt man nun alle überlappenden Intervalle sukzessiv, erhält man sogenannte $\mathcal{R}$-Faktoren, deren Länge im schlimmsten Fall durch $|R(u)| \times 2|S| + 2|S| + 1 \leq 2|S|^2 + 1 =: l'$ beschränkt sind. Dadurch wurde die Menge *blau* von nicht überlappenden $\mathcal{R}$-Faktoren auf u gebildet. Die duale Konstruktion mit der $\mathcal{L}$-Faktorisierung auf v liefert die ebenfalls disjunkte Intervallmenge *rot*. Dadurch ergeben sich die folgenden Faktorisierungen für u in $\mathcal{R}$-Faktoren $x_j := u|_J, J \in blau$ und v in $\mathcal{L}$-Faktoren $y_j := v|_J, J \in rot$, wobei $\alpha, \omega \in \Gamma^l$ gemeinsamer Anfang- und Endteil sind und eventuell durch Zusammenfügen bereits in x_1, x_k bzw. y_k, y_1 enthalten sind.

$$u = \alpha \cdot u_0 \cdot x_1 \cdot u_1 \cdot \ldots \cdot x_k \cdot u_k \cdot \omega \qquad u_i, x_i \in \Gamma^*, \; l'' \leq |x_i| \leq l' \tag{4.1}$$

$$v = \alpha \cdot v_k \cdot y_k \cdot \ldots \cdot v_1 y_1 \cdot v_0 \cdot \omega \qquad v_i, y_i \in \Gamma^*, \; l'' \leq |y_i| \leq l' \tag{4.2}$$

Nach Konstruktion existiert innerhalb eines Wortes u_i kein $\mathcal{R}$-Abstieg und innerhalb eines v_i kein $\mathcal{L}$-Abstieg. Wie bereits zu Beginn dieses Kapitels erwähnt, gilt durch das Faktor-Abstiegslemma (Lemma 2.3), dass diese Stellen der Faktorisierung durch Wortranker definiert sind.

Da $u \sim_{(l,n)} v$, lassen sich die blauen Faktoren x_i auch auf v übertragen.

$$\begin{aligned} \overline{blau} &:= \{\, [j - |x_i| + 1, j] \;\mid\; j = X_{x_1} \dots X_{x_i}(v), 1 \leq i \leq k \,\} \\ \overline{rot} &:= \{\, [j, j + |y_i| - 1] \;\mid\; j = Y_{y_1} \dots Y_{y_i}(u), 1 \leq i \leq k \,\} \end{aligned}$$

Analog zum Beweis von Lemma 3.5 betrachten wir alle paarweisen relativen Lagen von blauen Faktoren x_i und roten Faktoren y_i, um auf ihre lineare Anordnung zu schließen. Es gibt vier mögliche Fälle für die relative Lage von zwei Faktoren.

1. Keine Überlappung:

$$\begin{aligned} &X_{x_1} \dots X_{x_i}^{|x_i|}(u) < Y_{y_1} \dots Y_{y_j}^{|y_j|}(u) \\ \Leftrightarrow &X_{x_1} \dots X_{x_i}^{|x_i|} X_{y_j}^1 \dots X_{y_1}(u) \ definiert \\ \Leftrightarrow &X_{x_1} \dots X_{x_i}^{|x_i|} X_{y_j}^1 \dots X_{y_1}(v) \ definiert \\ \Leftrightarrow &X_{x_1} \dots X_{x_i}^{|x_i|}(v) < Y_{y_1} \dots Y_{y_j}^{|y_j|}(v) \end{aligned}$$

2. Überlappung der Form $x_i\gamma = \beta y_j$ mit $\gamma, \beta \in \Gamma^*, |\beta| \geq 1$: Bezeichne $2 \leq t \leq |x_i|$ die Position des ersten Zeichens von y_j in x_i. Nach Konstruktion liegen die Faktoren $y_j \dots y_1$ möglichst weit rechts, wodurch t maximal gewählt ist.

$$\begin{aligned} &X_{x_1} \dots X_{x_i}^{t}(u) = Y_{y_1} \dots Y_{y_j}^{|y_j|}(u) \\ \Rightarrow &X_{x_1} \dots X_{x_i}^{t-1} X_{y_j} \dots X_{y_1}(u) \ definiert \\ \Leftrightarrow &X_{x_1} \dots X_{x_i}^{t-1} X_{y_j} \dots X_{y_1}(v) \ definiert \\ \Rightarrow &X_{x_1} \dots X_{x_i}^{t}(v) \leq Y_{y_1} \dots Y_{y_j}^{|y_j|}(v) \end{aligned}$$

   Angenommen unerwünschterweise gilt auf v sogar $<$, dann könnte ein y_j Faktor auch an der Position $t+1$ beginnen, wobei diese auch außerhalb von x_i liegen könnte.

$$\begin{aligned} &X_{x_1} \dots X_{x_i}^{t} < Y_{y_1} \dots Y_{y_j}^{|y_j|}(v) \\ \Rightarrow &X_{x_1} \dots X_{x_i}^{t} X_{y_j} \dots X_{y_1}(v) \ definiert \\ \Rightarrow &X_{x_1} \dots X_{x_i}^{t} X_{y_j} \dots X_{y_1}(u) \ definiert \\ \Rightarrow &\text{Widerspruch zur Maximalität von } t \text{ bzw. zur Überlappung auf } u \text{ sonst.} \end{aligned}$$

   Die beiden Faktoren haben also auch auf v die gleiche Überlappung, wodurch wiederum beide Richtungen gelten.

3. Überlappungen der Form $y_j\gamma = \beta x_i =: w_{ij}$ mit $\gamma, \beta \in \Gamma^*$: Die Länge von w_{ij} ist durch $2 \times l' - 1$ beschränkt. Hier ist es möglich, dass vorherige x Faktoren ganz

oder teilweise in β enthalten sind und entsprechend y Faktoren in γ. Sei also $x_{i'}$ der letzten Faktor, der nicht mehr ganz in β enthalten ist($i' < i$) und $y_{j'}$ der ersten Faktor, der nicht mehr ganz in γ enthalten ist($j < j'$). Nach Konstruktion ist die Position der Faktoren $x_1 \dots x_i$ minimal und die von $y_j \dots y_1$ maximal gewählt.

$$\begin{aligned}
& X_{x_1} \dots X_{x_{i'}} X_{w_{ij}}(u) = Y_{y_1} \dots Y_{y_j}^{|y_j|}(u) \\
\Rightarrow & X_{x_1} \dots X_{x_{i'}} X_{w_{ij}} X_{y_{j'}} \dots X_{y_1}(u) \ definiert \\
\Leftrightarrow & X_{x_1} \dots X_{x_{i'}} X_{w_{ij}} X_{y_{j'}} \dots X_{y_1}(v) \ definiert \\
\Rightarrow & X_{x_1} \dots X_{x_{i'}} X_{w_{ij}}(v) \leq Y_{y_1} \dots Y_{y_j}^{|y_j|}(v) \ \wedge \qquad (4.3) \\
& X_{x_1} \dots X_{x_i}^{|x_i|}(v) \leq Y_{y_1} \dots Y_{y_{j'}} Y_{w_{ij}}(v) \qquad (4.4)
\end{aligned}$$

Angenommen (4.3) oder (4.4) erfüllen sogar $<$. Dann lässt sich dies in ähnlicher weise auf einen Widerspruch führen, dass y_j maximal auf u lag bzw. x_i minimal auf u lag. Dadurch gilt Gleichheit der Überlappung auf v und in gleicher Weise die Implikationen rückwärts.

4. Einschluss: Diese Situationen lassen sich analog zu Fall 3 lösen, wobei jeweils der umschließende Faktor als w_{ij} gewählt wird.

Die Abstiegsfaktoren sind also in u und v in gleicher Weise angeordnet. Um dies zu gewährleisten benötigten wir für $\sim_{(l,n)}$ nur

$$\begin{aligned}
n &:= 2|S|^2 + 2|S| - 1 \qquad && \geq |R(u)| + |L(v)| + l' \\
l &:= 4|S|^2 + 1 \qquad && \geq 2 \times l' - 1
\end{aligned}$$

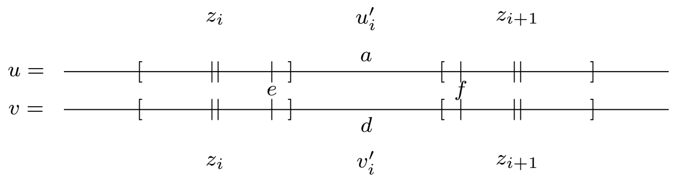
Nun sind wir in der Lage grüne Intervalle auf u zu definieren, die durch Erweiterung der blauen Intervalle mit roten Überlappungen entstanden sind. Dies bedeutet, dass innerhalb der Zwischenstellen noch immer keine $\mathcal{R}$-Abstiege existieren. Analog ist $\overline{gr\ddot{u}n}$ auf v durch Erweiterung roter Intervalle mit blauen Überlappungen entstanden. Somit existieren noch immer keine $\mathcal{L}$-Abstiege innerhalb der Zwischenstellen. Durch obige Fallunterscheidung ist klar, dass diese Intervalle in u und v gleiche Beschriftungen z_i haben.

$$\begin{aligned}
gr\ddot{u}n &:= \left\{ \bigcup_{I \text{ überlappend}} I \;\middle|\; I \in blau \cup \overline{rot} \right\} \text{ auf } u \\
\overline{gr\ddot{u}n} &:= \left\{ \bigcup_{I \text{ überlappend}} I \;\middle|\; I \in \overline{blau} \cup rot \right\} \text{ auf } v
\end{aligned}$$

Entscheidend sind die folgenden Faktorisierungen der Wörter u, v, die sich nun annehmen lassen.

$$\begin{aligned}
u &= \alpha u'_0 z_1 u'_1 \dots z_k u'_k \omega \qquad && u'_i, z_i \in \Gamma^*, l'' \leq |z_i| \leq l \\
v &= \alpha v'_0 z_1 v'_1 \dots z_k v'_k \omega \qquad && v'_i \in \Gamma^*
\end{aligned}$$

Wir betrachten nun induktiv von links nach rechts die Faktoren, welche die Zwischenstellen u'_i bzw. v'_i umschließen, für $0 \leq i \leq k$. Da diese mit ausreichnder Länge konstruiert sind, existiert nach Lemma 1.5 für den linken Faktor, der insbesondere α sein kann, das Idempotente e welches wir einfügen können. Analoges gilt für den rechten Faktor.



Da die u'_i keine $\mathcal{R}$-Abstiege und die v'_i keine $\mathcal{L}$-Abstiege enthalten, lässt sich damit folgende Klassenzugehörigkeit für Präfix p bis zum Idempotenten e und Suffix q ab dem Idempotenten f ablesen.

$$pe \; \mathcal{R} \; peaf \qquad edfq \; \mathcal{L} \; fq$$

Also existieren Elemente $\bar{a}$ und $\bar{d}$ so, dass

$$pe = p(eaf\bar{a})^\pi e \qquad fq = f(\bar{d}edf)^\pi q.$$

Betrachtet man nun zusätzlich die $\mathcal{B}_1$-Identität, dann gilt:

$$\begin{aligned} p(eaf\bar{a})^\pi eaf(\bar{d}edf)^\pi q &= p(eaf\bar{a})^\pi edf(\bar{d}edf)^\pi q && \mathcal{B}_1(1.17) \\ peafq &= p(eaf\bar{a})^\pi edf(\bar{d}edf)^\pi q && \text{links abpumpen} \\ peafq &= pedfq && \text{rechts abpumpen} \end{aligned}$$

Dieses Argument lässt sich induktiv entlang den Faktoren z_i (beispielsweise von links nach rechts) anwenden. Dadurch ist $h(v) = h(u)$ und somit $u \sim_L v$. $\square$

Algorithmus zur Entscheidbarkeit

Da sich algebraische Eigenschaften leicht überprüfen lassen, legt dies folgenden Algorithmus zur Entscheidung der Definierbarkeit in $\mathbb{B}\Sigma_1[<, +1, \min, \max]$ nahe. Die Implementierung ist in der Sprache GAP für eine bestimmte reguläre Sprache, die wir in Kapitel 6 näher untersuchen werden, gegeben. [4]

```
LoadPackage("automata");

L := RationalExpression("(011*U022*)*011*3(22*3U11*3)*");
Automat := RatExpToAut(L);
S := TransitionSemigroup(Automat);
Display(Automat);

ELEM := Elements(S);
I := Idempotents(S);

for e in I do
  for a in ELEM do
   for b in ELEM do
    for f in I do
      Banfang := e*a*f*b;
      while Banfang <> (Banfang^2) do
       Banfang := Banfang * (e*a*f*b);
      od;
      for c in ELEM do
       for d in ELEM do
        Bende := c*e*d*f;
        while Bende <> (Bende^2) do
          Bende := Bende *(c*e*d*f);
        od;
        Left := Banfang * e*a*f * Bende;
        Right := Banfang * e*d*f *Bende;
        if Left <> Right then
          Print("--------------------------------\n");
          Print("GEGENBEISPIEL:\n");
          Print("e:"); Print(e); Print("\n");
          Print("f:"); Print(f); Print("\n");
          Print("a:"); Print(a); Print("\n");
          Print("b:"); Print(b); Print("\n");
          Print("c:"); Print(c); Print("\n");
          Print("d:"); Print(d); Print("\n");
          Print("Banfang:"); Print(Banfang); Print("\n");
          Print("Bende:"); Print(Bende); Print("\n");
          Print("Links:"); Print(Left); Print("\n");
          Print("Rechts:"); Print(Right); Print("\n");
          Print("--------------------------------\n");
        fi;
       od;
      od;
    od;
   od;
  od;
od;
```

5 Weitere entscheidbare Logikfragmente

Im vorherigen Kapitel haben wir gezeigt, dass Sprachen $L \in \mathbb{B}\Sigma_1[<, +1, \min, \max]$ genau denen entsprechen, deren syntaktische Halbgruppe $Synt(L)$ die $\mathcal{B}_1$-Identität erfüllen. Da die syntaktische Halbgruppe regulärer Sprachen endlich ist, war es möglich durch einfaches Nachrechnen dieser Identität für alle Elemente der Halbgruppe die Definierbarkeit im Logikfragment zu entscheiden. Die Verwendung der min und max Prädikate waren in den Beweisen von Kapitel 4 jedoch nicht von elementarer Bedeutung. Dadurch haben wir die Möglichkeit auf ähnliche Weise entscheidbare Charakterisierungen für die Fragmente $\mathbb{B}\Sigma_1[<, +1]$, $\mathbb{B}\Sigma_1[<, +1, \min]$ und $\mathbb{B}\Sigma_1[<, +1, \max]$ anzugeben, wobei das Fehlen eines der Prädikate sich durch die Hinzunahme einer weiteren, einfach entscheidbaren Bedingung an die syntaktische Halbgruppe äußert. Diese entscheidbaren Charakterisierungen sind neue Ergebnisse. Die folgenden Beweise sind alle sehr ähnlich zu den vier Lemmata aus Kapitel 4, wobei wir auf die Vollständigkeit hier verzichten und nur die Stellen beschreiben, die abgewandelt werden müssen.

5.1 Das Fragment $\mathbb{B}\Sigma_1[<, +1, \min]$

Satz 5.1: *Die folgenden Charakterisierungen sind äquivalent:*

1. $L = \underset{r \in R[X_w]}{\mathbb{B}} (L(r) \cap min_r \Gamma^*) = \underset{r \in R[Y_w]}{\mathbb{B}} (L(r) \cap min_r \Gamma^*)$
2. *L ist in $\mathbb{B}\Sigma_1[<, +1, \min]$ definierbar*
3. $L = \mathbb{B}(w_0 \Gamma^* w_1 \ldots w_k \Gamma^*, w_i \in \Gamma^+)$
4. *$Synt(L) \in [\![\mathcal{B}_1]\!]$ und $L = \bigcup_{u \in L} [u]_{\mathcal{R}}$*

Beweis:
"1 ⇒ 2": Beweis analog zu Lemma 4.2.
"2 ⇒ 3": Beweis analog zu Lemma 4.3.
"3 ⇒ 4": Die $\mathcal{B}_1$-Identität lässt sich analog zu Lemma 4.4 zeigen.
Wir zeigen $L = \bigcup_{u \in L} [u]_{\mathcal{R}}$. Seien $u', v' \in Synt(L)$ mit $u' \mathcal{R} v'$. Dann existieren $x', y' \in Synt(L)$ mit $u' = v'x'$, $v' = u'y'$. Für diese Elemente lassen sich wiederum nichtleere Urbilder des syntaktischen Homomorphismus $u, v, x, y \in \Gamma^+$ wählen. Sei K ein Teilausdruck von L der Form $K = w_0 \Gamma^* \ldots w_k \Gamma^*$. Falls $u \in K$, ist auch $v \in K$, da $v = uy$. Die zweite Faktorisierung liefert: $v \in K \Rightarrow u \in K$.

"$4 \Rightarrow 1$": Wir konstruieren in exakt der gleichen Weise wie in Lemma 4.5 die Intervalle *grün* auf Wort u und $\overline{grün}$ auf v wobei $u \sim_{(l,n)} v$ mit

$$\begin{aligned} u \sim_{(l,n)} v :\Leftrightarrow & first_l(u) = first_l(v) \wedge \\ & \forall r = X_{w_1} \dots X_{w_n} : r(u) def. \Leftrightarrow r(v) def. \text{ mit } 1 \leq w_i \leq l. \end{aligned}$$

Indem wir nur einen Wortranker mehr erlauben, können wir zusätzlich sicherstellen, dass die letzten l Zeichen von v auch hinter denen der z_k auf u vorkommen. Falls v_k das leere Wort ist, gilt dies bereits. Anderenfalls ändert ein vorausgehender Wortranker $Y_{last_l(v_k)}$ weder die Definierbarkeit der $\mathcal{L}$-Abstiegsfaktoren auf v noch ein folgender $X_{last_l(v_k)}$ die Positionen der $\mathcal{R}$-Abstiegsfaktoren auf u. Wir erhalten dadurch folgende Faktorisierung der beiden Wörter u und v:

$$\begin{aligned} u =& \alpha u_0 z_1 u_1 \dots z_k u_k last_l(v_k) u_{k+1} && \text{mit } \alpha, u_i, v_i, z_i \in \Gamma^* \\ v =& \alpha v_0 z_1 v_1 \dots z_k v_k && |z_i| > 2|Synt(L)| \end{aligned}$$

Durch Konstruktion ist wiederum innerhalb der u_i kein $\mathcal{R}$-Abstieg und innerhalb der v_i kein $\mathcal{L}$-Abstieg vorhanden. Dadurch lässt sich in gleicher Weise, von links nach rechts, die $\mathcal{B}_1$-Identität anwenden. Mangels fehlendem gemeinsamen Ende erhalten wir diesmal:

$$h(v) = h(\alpha u_0 z_1 \dots last_l(v_k)) \;\mathcal{R}\; h(u).$$

Da unsere Sprache gegebenerweise eine Vereinigung solcher $\mathcal{R}$-Klassen ist, folgt schließlich:

$$L = \bigcup_{u \in L} [u]_{\mathcal{R}} = \bigcup_{u \in L} [u]_{\sim_{(l,n)}}.$$

□

Bemerkung 5.1: In der Tat ist Eigenschaft 4 des obigen Satzes entscheidbar. Zum Algorithmus aus Kapitel 4 müssen lediglich zwei Schleifen der folgenden Form hinzugefügt werden. $P := h(L) \subseteq Synt(L)$ bezeichnet die Menge der akzeptierenden Elemente.

```
foreach a in P do
   foreach b in S do
      if (a*S=b*S) and not(b in P) then
         return FALSE;
      fi
   od
od
```

5.2 Das Fragment $\mathbb{B}\Sigma_1[<, +1, \max]$

Satz 5.2: *Die folgenden Charakterisierungen sind äquivalent:*

1. $L = \underset{r\in R[X_w]}{\mathbb{B}}(L(r)\cap\Gamma^* max_r) = \underset{r\in R[Y_w]}{\mathbb{B}}(L(r)\cap\Gamma^* max_r)$
2. L *ist in* $\mathbb{B}\Sigma_1[<,+1,\max]$ *definierbar*
3. $L = \mathbb{B}(\Gamma^* w_0 \dots \Gamma^* w_k, w_i \in \Gamma^+)$
4. $Synt(L) \in [\![\mathcal{B}_1]\!]$ *und* $L = \bigcup_{u\in L}[u]_{\mathcal{L}}$

Beweis:
"$1 \Rightarrow 2$": Beweis analog zu Lemma 4.2.
"$2 \Rightarrow 3$": Beweis analog zu Lemma 4.3.
"$3 \Rightarrow 4$": Die $\mathcal{B}_1$-Identität lässt sich analog zu Lemma 4.4 zeigen.
Wir zeigen $L = \bigcup_{u\in L}[u]_{\mathcal{L}}$. Seien $u', v' \in Synt(L)$ mit $u\mathcal{L}v$. Dann existieren $x', y' \in Synt(L)$ mit $x'u' = v'$, $y'v' = u'$. Für diese Elemente lassen sich wiederum nichtleere Urbilder des syntaktischen Homomorphismus $u, v, x, y \in \Gamma^+$ wählen. Sei K ein Teilausdruck von L der Form $K = \Gamma^* w_0 \dots \Gamma^* w_k$. Falls $u \in K$, ist auch $v \in K$, da $v = xu$. Die zweite Faktorisierung liefert: $v \in K \Rightarrow u \in K$.
"$4 \Rightarrow 1$": Wir konstruieren in exakt der gleichen Weise wie in Lemma 4.5 die Intervalle $grün$ auf Wort u und $\overline{grün}$ auf v wobei $u \sim_{(l,n)} v$ mit

$$\begin{aligned} u \sim_{(l,n)} v :\Leftrightarrow & last_l(u) = last_l(v) \wedge \\ & \forall r = X_{w_1}\dots X_{w_n} : r(u)def. \Leftrightarrow r(v)def. \text{ mit } 1 \le w_i \le l. \end{aligned}$$

Indem wir nur einen Wortranker mehr erlauben, können wir zusätzlich sicherstellen, dass die ersten l Zeichen von u auch vor denen der z_1 auf v vorkommen. Falls u_0 das leere Wort ist, gilt dies bereits. Anderenfalls ändert weder der nachfolgende Wortranker $Y_{first_l(u_0)}$ die Definierberkeit der $\mathcal{L}$-Abstiegsfaktoren auf v noch ein vorausgehender $X_{first_l(u_0)}$ die Positionen der $\mathcal{R}$-Abstiegsfaktoren auf u. Wir erhalten dadurch folgende Faktorisierung der beiden Wörter u und v:

$$\begin{aligned} u &= u_0 z_1 u_1 \dots z_k u_k \omega & \text{mit } \omega, u_i, v_i, z_i \in \Gamma^* \\ v &= v_{-1} first_l(u_0) v_0 z_1 v_1 \dots z_k v_k \omega & |z_i| > 2|Synt(L)| \end{aligned}$$

Durch Konstruktion ist wiederum innerhalb der u_i kein $\mathcal{R}$-Abstieg und innerhalb der v_i kein $\mathcal{L}$-Abstieg vorhanden. Dadurch lässt sich, diesmal von rechts nach links, die $\mathcal{B}_1$-Identität anwenden und wir erhalten mangels fehlendem gemeinsamen Anfangs:

$$h(u) = h(first_l(u_0) v_0 z_1 \dots \omega) \ \mathcal{L}\ h(v).$$

Da unsere Sprache gegebenerweise eine Vereinigung solcher $\mathcal{L}$-Klassen ist, folgt schließlich:

$$L = \bigcup_{u\in L}[u]_{\mathcal{L}} = \bigcup_{u\in L}[u]_{\sim_{(l,n)}}.$$

$\square$

5.3 Das Fragment $\mathbb{B}\Sigma_1[<,+1]$

Satz 5.3: *Die folgenden Charakterisierungen sind äquivalent:*

1. $L = \underset{r\in R[X_w]}{\mathbb{B}} L(r) = \underset{r\in R[Y_w]}{\mathbb{B}} L(r)$
2. *L ist in* $\mathbb{B}\Sigma_1[<,+1]$ *definierbar*
3. $L = \mathbb{B}(\Gamma^* w_0 \dots w_k \Gamma^*, w_i \in \Gamma^+)$
4. $Synt(L) \in [\![\mathcal{B}_1]\!]$ *und* $L = \bigcup_{u\in L}[u]_{\mathcal{J}}$

Beweis:
"$1 \Rightarrow 2$": Beweis analog zu Lemma 4.2.
"$2 \Rightarrow 3$": Beweis analog zu Lemma 4.3.
"$3 \Rightarrow 4$": Die $\mathcal{B}_1$-Identität lässt sich analog zu Lemma 4.4 zeigen.
Wir zeigen $L = \bigcup_{u\in L}[u]_{\mathcal{J}}$. Seien $u', v' \in Synt(L)$ mit $u\mathcal{J}v$. Dann existieren $w', x', y', z' \in Synt(L)$ mit $u' = w'v'x'$, $v' = y'u'z'$. Für diese Elemente lassen sich wiederum nichtleere Urbilder des syntaktischen Homomorphismus $u, v, w, x, y, z \in \Gamma^+$ wählen. Sei K ein Teilausdruck von L der Form $K = \Gamma^* w_0 \dots w_k \Gamma^*$. Falls $u \in K$, ist auch $v \in K$, da $v = yuz$. Die zweite Faktorisierung liefert: $v \in K \Rightarrow u \in K$.
"$4 \Rightarrow 1$": Wir konstruieren in exakt der gleichen Weise wie in Lemma 4.5 die Intervalle $grün$ auf Wort u und $\overline{grün}$ auf v wobei $u \sim_{(l,n)} v$ mit

$$u \sim_{(l,n)} v :\Leftrightarrow \forall r = X_{w_1} \dots X_{w_n} : r(u)def. \Leftrightarrow r(v)def. \quad \text{mit } 1 \le w_i \le l.$$

Indem wir nur zwei Wortranker mehr erlauben, können wir, analog zu den Beweisen der beiden vorherigen Sätze, die folgende Faktorisierung sicherstellen:

$$\begin{aligned} u &= \quad u_0 \underbrace{z_1 u_1 \dots z_k}_{=:\bar{u}} u_k last_l(v_k) u_{k+1} && \text{mit } u_i, v_i, z_i \in \Gamma^* \\ v &= \quad v_{-1} first_l(u_0) v_0 \underbrace{z_1 v_1 \dots z_k}_{=:\bar{v}} v_k && |z_i| > 2|Synt(L)| \end{aligned}$$

Durch sukzessive Anwendung der $\mathcal{B}_1$-Identität erhalten wir:

$$h(v) \;\mathcal{L}\; h(\bar{v}) = h(\bar{u}) \;\mathcal{R}\; h(u).$$

Betrachten wir noch die $\mathcal{J}$-Klasse von $h(u)$

$$S\Big(h(u)S\Big) = Sh(\bar{u})S = \Big(Sh(\bar{v})\Big)S = Sh(v)S$$

Da unsere Sprache gegebenerweise eine Vereinigung solcher $\mathcal{J}$-Klassen ist, folgt schließlich:

$$L = \bigcup_{u\in L}[u]_{\mathcal{J}} = \bigcup_{u\in L}[u]_{\sim_{(l,n)}}.$$

□

6 Die Beziehung zwischen den Klassen $\mathcal{B}_1$ und $L\mathcal{J}$

Von Knast wurde eine Sprache $L \subseteq \Gamma^+$ gegeben, deren syntaktische Halbgruppe $Synt(L)$ in $L\mathcal{J}$ liegt, aber nicht die $\mathcal{B}_1$-Identität erfüllt.[8] Diest zeigt, dass es sich um eine echte Teilmenge handelt. Dieses Beispiel wollen wir im Folgenden näher untersuchen, um allgemeinere Eigenschaften zur Unterscheidung der Klassen zu gewinnen.

Betrachte die folgende Sprache $L \subseteq \Gamma^+$ über dem Alphabet $\Gamma = \{0, 1, 2, 3\}$.

$$L := (01^+|02^+)^*01^+3(1^+3|2^+3)^*$$

Die Sprache L wird von folgendem Automat akzeptiert:

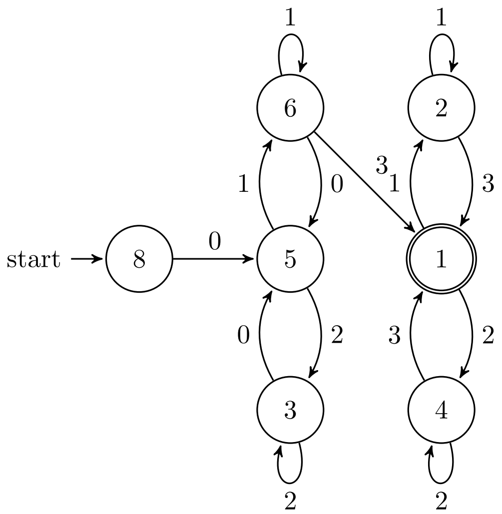


Abbildung 6.1: Deterministischer endlicher Automat für die Sprache $L \notin \mathcal{B}_1$, wobei der Fehlerzustand 7 nicht eingezeichnet ist.

Um $L \notin \mathcal{B}_1$ zu zeigen, genügt es Elemente aus $Synt(L)$, sprich Transformationen auf dem Automaten, anzugeben, welche die $\mathcal{B}_1$-Identität nicht erfüllen. Wir geben

Vertreter und die zugehörigen Transformationen auf dem Automaten an:

$$\begin{aligned}
e = [1] =& [2,2,7,7,6,6,7,7]\\
f = [2] =& [4,7,3,4,3,7,7,7]\\
a = [02] =& [7,7,3,7,7,3,7,3]\\
b = [20] =& [7,7,5,7,5,7,7,7]\\
c = [23] =& [1,7,7,1,7,7,7,7]\\
d = [13] =& [1,1,7,7,1,1,7,7]
\end{aligned}$$

Durch zweimaliges Ausführen der Transformationen e bzw. f kann man sich überzeugen, dass diese Elemente idempotent (z.B. $e^2 = e$) sind. Wir betrachten nun Anfangs- und Endteil der beiden Seiten der $\mathcal{B}_1$-Identität.

$$\begin{aligned}
(eafb)^\pi &= ([1][02][2][20])^\pi = [10]^\pi = [10]\\
(cedf)^\pi &= ([23][1][13][2])^\pi = [232]^\pi = [232]
\end{aligned}$$

Nun können wir Wörter angeben, die laut der $\mathcal{B}_1$-Identität äquivalent sein müssten, jedoch nicht gleichzeitig in L liegen.

$$\begin{gathered}
u := 10 \cdot 1 \cdot 02 \cdot 2 \cdot 232 \text{ und } v := 10 \cdot 1 \cdot 13 \cdot 2 \cdot 232\\
0u3 \notin L \text{ aber } 0v3 \in L\\
\Rightarrow L \notin [\![\mathcal{B}_1]\!]
\end{gathered}$$

Für diese Sprache lässt sich die folgende äquivalente Beschreibung angeben. Da $Synt(L)$ nicht die $\mathcal{B}_1$-Identität erfüllt, lässt sie sich nicht ganz in dennen von Dot-Depth-1 erlaubten Konstrukten darstellen.

$$\begin{aligned}
L :=& (01^+|02^+)^*01^+3(1^+3|2^+3)^* \\
=& \overline{\Gamma^*00\Gamma^*} \cap \overline{\Gamma^*33\Gamma^*} \cap \overline{\Gamma^*12\Gamma^*} \cap \overline{\Gamma^*21\Gamma^*} \cap && \text{Übergänge}\\
& \overline{\Gamma^*3\Gamma^*0\Gamma^*} \cap && \text{nach letzter 3 keine 0}\\
& 0\Gamma^* \cap \Gamma^*3 \cap && \text{Anfang und Endteile}\\
& \Gamma^*01(\Gamma \setminus \{0\})^* && \text{nicht in } \mathcal{B}_1 \text{ erlaubt}
\end{aligned}$$

Verbotene Muster für $\mathcal{LJ}$ und $\mathcal{B}_1$

Obwohl eine exakte Unterscheidung, durch die Angabe einer Logikmodalität, der beiden Klassen nicht möglich war, lassen sich einfach Bedingungen an die endlichen Automaten solcher Sprachen definieren.[1] Von den Identitäten für $\mathcal{LJ}$ und $\mathcal{B}_1$ lassen sich leicht Muster ablesen, die nicht in den Automaten von Sprachen in diesen Klassen enthalten sein drüfen. Das folgende Muster $M_{\mathcal{B}_1}$ darf nicht in Automaten von Sprachen, deren syntaktische Halbgruppe die $\mathcal{B}_1$-Identität erfüllt, liegen.

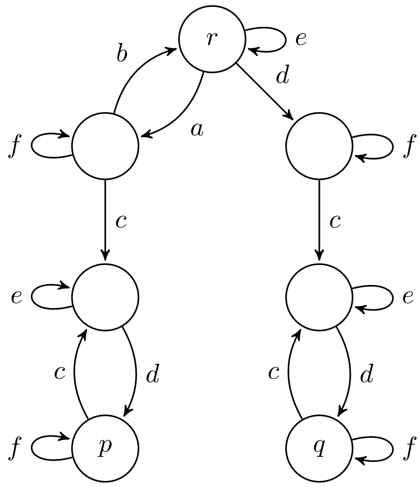


Abbildung 6.2: Verbotenes Muster $M_{\mathcal{B}_1}$ für Sprachen, deren syntaktische Halbgruppe die $\mathcal{B}_1$-Identität erfüllt. Automaten solcher Sprachen dürfen dieses Muster nicht enthalten, falls $p \neq q$.

Diese Eigenschaft lässt sich durch das folgende Lemma zeigen.

Lemma 6.1: *Sei $L \subseteq \Gamma^*$ regulär. $Synt(L) \in [\![\mathcal{B}_1]\!] \Rightarrow M_{\mathcal{B}_1}$ ist nicht im Automat von L enthalten.*

Beweis: "$\Rightarrow$" Sei $S := Synt(L)$ und erfülle die $\mathcal{B}_1$-Identität. Angenommen der Automat von L enthält das Muster $M_{\mathcal{B}_1}$ mit $p \neq q$. Dann ist

$$\begin{aligned} p =& r \cdot (e^\pi a f^\pi b)^\pi e^\pi a f^\pi (c e^\pi d f^\pi)^\pi \\ =& r \cdot (e^\pi a f^\pi b)^\pi e^\pi d f^\pi (c e^\pi d f^\pi)^\pi \\ =& q \end{aligned}$$

Dies ist ein Widerspruch zu $p \neq q$, wodurch der Automat das Muster $M_{\mathcal{B}_1}$ nicht enthält. □

Ein ähnliches verbotenes Muster $M_{L\mathcal{J}}$ für die Klasse $L\mathcal{J}$ lässt sich auf ähnliche Weise angeben und beweisen.

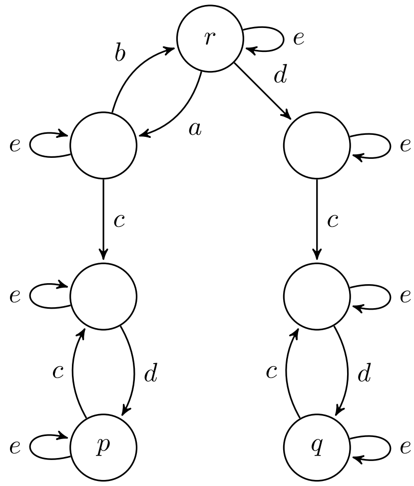


Abbildung 6.3: Verbotenes Muster $M_{L\mathcal{J}}$ für Sprachen, deren syntaktische Halbgruppe in $L\mathcal{J}$ ist. Automaten solcher Sprachen dürfen dieses Muster nicht enthalten, falls $p \neq q$.

7 Zusammenfassung

Das Abstiegslemma für $\mathbb{DA}$ und $\mathbb{LDA}$ aus Kapitel 2 hat sich, in Verbindung mit Rankern und Wortranker, als sehr wichtiges Hilfsmittel erwiesen. Dadurch war es nicht nur möglich auf die Positionen eines Wortes, welche für dessen Akzeptanz entscheidend sind, generisch zuzugreifen, sondern auch diese kombinatorisch zu vergleichen. Wie wir gesehen haben, lassen sich Äquivalenz von Rankersprachen und Logikfragmenten oft leicht nachweisen.

In Kapitel 3 haben wir gesehen, dass sich damit die Identität der $\mathcal{J}$-trivialen Monoide als Ersetzungsregel zwischen diesen Positionen auffassen lässt. Einen Beweis für die bekannte Äquivalenz zwischen $\mathbb{B}\Sigma_1[<]$ und $\mathcal{J}$-trivialen Monoiden konnten wir so einfach angeben. Auch für Knasts Theorem haben wir, unter Verwendung des Faktorabstiegslemma und Faktorrankern, einen neuen, kombinatorischen Beweis in Kapitel 4 angeben können.

In Kapitel 5 ist es uns sogar gelungen durch diese Beweismethode für die Logikfragmente mit Nachfolgerprädikat $+1$, jedoch ohne min bzw. max Prädikate, ein neues Entscheidungsverfahren anzugeben. Eine Übersicht, über alle in dieser Arbeit bewiesenen, entscheidbaren Logikfragmenten, ist in nachfolgender Tabelle gegeben.

Logikfragment	Sprachen	Entscheidbarkeit	Satz
$\mathbb{B}\Sigma_1[<]$	$\mathbb{B}(\Gamma^* a_1 \Gamma^* \dots a_k \Gamma^*)$	$\mathcal{J}$-trivial	3.1
$\mathbb{B}\Sigma_1[<,+1]$	$\mathbb{B}(\Gamma^* w_1 \Gamma^* \dots w_k \Gamma^*)$	$\mathcal{B}_1 \;\wedge\; L = \bigcup_u [u]_{\mathcal{J}}$	5.3
$\mathbb{B}\Sigma_1[<,+1,\min]$	$\mathbb{B}(w_1 \Gamma^* \dots w_k \Gamma^*)$	$\mathcal{B}_1 \;\wedge\; L = \bigcup_u [u]_{\mathcal{R}}$	5.1
$\mathbb{B}\Sigma_1[<,+1,\max]$	$\mathbb{B}(\Gamma^* w_1 \Gamma^* \dots w_k \Gamma^*)$	$\mathcal{B}_1 \;\wedge\; L = \bigcup_u [u]_{\mathcal{L}}$	5.2
$\mathbb{B}\Sigma_1[<,+1,\min,\max]$	$\mathbb{B}(w_1 \Gamma^* w_2 \dots \Gamma^* w_k)$	$\mathcal{B}_1$	4.1

In Kapitel 6 wurde der Zusammenhang zur Klasse $L\mathcal{J}$ von Halbgruppen untersucht. Leider ist es uns nicht gelungen eine Logikcharakterisierung hierfür anzugeben.

Alle in obiger Tabelle dargestellen Logikfragmente sind entscheidbar, für einen gegebenem regulären Ausdruck. Eine Beispielimplementierung in der Programmiersprache GAP wurde in Kapitel 4 gegeben.

Literaturverzeichnis

[1] Joëlle Cohen, Dominique Perrin, and Jean eric Pin. On the expressive power of temporal logic. *J. Comput. System Sci*, 46:271–294, 1993.

[2] R. S. Cohen and J. A. Brzozowski. Dot-depth of star-free events. *Journal of Computer System Sciences*, 5(1):1–16, 1971.

[3] V. Diekert, P. Gastin, and M. Kufleitner. A survey on small fragments of first-order logic over finite words. *International Journal of Foundations of Computer Science*, 19(3):513–548, 2008.

[4] The GAP Group. *GAP – Groups, Algorithms, and Programming, Version 4.4.12*, 2008.

[5] Christian Glaßer and Heinz Schmitz. Concatenation hierarchies and forbidden patterns. Technical report, 2000.

[6] John E. Hopcroft, Rajeev Motwani, and Jeffrey D. Ullman. Introduction to automata theory, languages, and computation, 2nd edition. *SIGACT News*, 32(1):60–65, 2001.

[7] Ondřej Klíma. Piecewise testable languages via combinatorics on words. In *Berichtsband der Konferenz Words 2009*, 2009.

[8] R. Knast. A semigroup characterization of dot-depth one languages. *R.A.I.R.O. Informatique théorique*, 17(4):321–330, 1983.

[9] R. Knast and J. A. Brzozowski. The dot-depth hierarchy of star-free languages is infinite. *Journal of Computer System Sciences*, 16(1):37–55, 1978.

[10] Robert McNaughton and Seymour Papert. Counter-free automata. *M.I.T. research monograph*, 65:163, 1971.

[11] J.-É Pin and P. Weil. The wreath product principle for ordered semigroups. *Commun. Algebra*, 30(12):5677–5713, 2002.

[12] Uwe Schöning. *Logik für Informatiker, 3. Auflage*, volume 56 of *Reihe Informatik*. Bibliographisches Institut, 1992.

[13] Uwe Schöning. *Theoretische Informatik - kurzgefasst*, volume 3. Spektrum Akademischer Verlag, 1997.

[14] M.P. Schützenberger. On finite monoids having only trivial subgroups. *Information and Control*, 8(2):190–194, 1965.

[15] Imre Simon. Piecewise testable events. In H. Brakhage, editor, *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975*, volume 33 of *Lecture Notes in Computer Science*, pages 214–222. Springer Berlin / Heidelberg, 1975.

[16] H. Straubing. A generalization of the schützenberger product of finite monoids. *Theoretical Computer Science*, 13(2):137–150, 1981.

[17] H. Straubing. Finite semigroup varieties of the form $\mathbb{V} * \mathbb{D}$. *Journal of Pure and Applied Algebra*, 36(1):53–94, 1985.

[18] Pascal Tesson and Denis Thérien. Logic meets algebra: the case of regular languages. *CoRR*, abs/cs/0701154, 2007.

[19] D. Thérien. Categories et langages de dot-depth un. *ITA*, 22(4):437–445, 1988.

[20] W. Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25(3):360–376, 1982.

Erklärung

Hiermit versichere ich, diese Arbeit selbständig verfasst und nur die angegebenen Quellen benutzt zu haben.

(Martin P. Seybold)