Ausdrucksstärke von Stickersystemen

Untersuchung der Ausdrucksstärke von Stickersystemen durch Vergleich mit Chomskygrammatiken und Mehrkopfautomaten

Diplom-Informatiker
Peter Weigel

Dezember 2009

Peter Weigel. Ausdrucksstärke von Stickersystemen. Untersuchung der Ausdrucksstärke von Stickersystemen durch Vergleich mit Chomskygrammatiken und Mehrkopfautomaten. Diplomarbeit, Martin-Luther-Universität Halle-Wittenberg, Institut für Informatik, Halle/Saale, 12.10.2004.

Peter Weigel. Ausdrucksstärke von Stickersystemen. Untersuchung der Ausdrucksstärke von Stickersystemen durch Vergleich mit Chomskygrammatiken und Mehrkopfautomaten. VDM Verlag Dr. Müller, Saarbrücken, Juli 2008, ISBN 978-3-639-00803-6.

Peter Weigel. Ausdrucksstärke von Stickersystemen. Untersuchung der Ausdrucksstärke von Stickersystemen durch Vergleich mit Chomskygrammatiken und Mehrkopfautomaten. www.stickersysteme.de, Halle/Saale, Dezember 2009.

Die Bausteine des Lebens sind Grundlage des theoretischen Konzeptes der Stickersysteme, das im Buch „DNA-Computing. New Computing Paradigms.“ von G. Paun, G. Rozenberg und A. Salomaa ausführlich untersucht wird. Die dort durchgeführten Komplexitätsuntersuchungen zum Vergleich von Stickersystemen und Chomskygrammatiken sind jedoch unvollständig.

In der vorliegenden Arbeit werden die genannten Untersuchungen auf Mehrkopfautomaten ausgedehnt und dadurch alle noch offenen Fragen zu Beziehungen zwischen Stickersprachfamilien und Chomskysprachfamilien beantwortet. Nebenbei werden dabei auch viele Fragen zu Beziehungen zwischen Stickersystemen und Mehrkopfautomaten, zu Beziehungen der Stickersprachfamilien untereinander und zu Abschlusseigenschaften der Stickersprachfamilien geklärt.

Diese Arbeit stellt gewissermaßen eine Ergänzung zu „DNA-Computing. New Computing Paradigms.“ dar und richtet sich hauptsächlich an Forscher, Dozenten und Studenten der Theoretischen Informatik und/oder Bioinformatik.

Hiermit danke ich Herrn Prof. Dr. Ludwig Staiger und Doz. Dr. Dietrich Kuske für Ihre hilfreichen Tipps und Anregungen bei der Erstellung der Diplomarbeit [Wei04]. Ein besonderer Dank geht außerdem an Herrn Jens Keilwagen für seine Anregungen zu Satz 4.12, Idee zu Anmerkung 4.13 und seinem Mitwirken an den Folgerungen 4.14 und 4.15.

Inhaltsverzeichnis

Kapitel 1

Einleitung

In [Adl94] beschrieb L. M. Adleman ein Verfahren zur Lösung des *Hamiltonia Path Problem (HPP)* auf der Grundlage von DNA-Strängen. Dieses unter der Bezeichnung *Adleman's Experiment* bekannte Verfahren bildet die Grundlage zu dem Konzept der *Stickersysteme*, das in [KPG98] unter dem Namen *regular sticker systems* eingeführt wurde. Stickersysteme mit der Fähigkeit zur Verlängerung auf der linken und rechten Seite wurden dagegen erstmals in [FPR98] mit der Bezeichnung *bidirectional sticker systems* erwähnt. In [PR98] wurden diese beiden Konzepte unter der Bezeichnung *sticker systems* zusammengeführt. Die Definition der Stickersysteme aus [PR98] und ein Großteil der Resultate und Beweise aus [KPG98], [FPR98] und [PR98] wurden anschließend in [PRS98] zusammengefasst und ergänzt.

Stickersysteme stellen eine Möglichkeit zur theoretischen Untersuchung der für die Sprach-, Berechenbarkeits- und Komplexitätstheorie interessanten Eigenschaften und Fähigkeiten von DNA-Strängen dar. Da sich die Konstrukte der Stickersysteme, die durchaus als Grammatiken angesehen werden können, grundlegend von den Chomskygrammatiken unterscheiden, bedürfen sie einer ausführlicheren Analyse. In [KPG98], [FPR98], [PR98] und [PRS98] werden bereits umfangreiche Untersuchungen zu Stickersystemen bzw. Stickersprachen durchgeführt und viele Beziehungen zwischen den Stickersprachfamilien untereinander und den Chomskysprachfamilien aufgedeckt.

Die vorliegende Arbeit ist eine aktualisierte Version der Diplomarbeit [Wei04] und stellt gewissermaßen eine Ergänzung zu [PRS98] dar.

Im Folgenden werden wir das in [PRS98] definierte Konzept der Stickersysteme geringfügig erweitern und einen Großteil der aus [PRS98] bekannten Ergebnisse über Stickersysteme auf das erweiterte Konzept übertragen. Außerdem werden wir zeigen, dass die Komplementarität $\rho = \{(x, x) : x \in V\}$ ausreicht. Wir werden die Beziehungen $REG \not\subseteq SSL(n)$, $LIN \not\subseteq OSL(n)$ und $CF \not\subseteq ASL(n)$ (inkl. $ASL(n) \subset CS$) beweisen, und damit alle noch offenen Fragen bezüglich den Beziehungen zwischen Stickersprachfamilien und Chomskysprachfamilien beantworten. Darüber hinaus werden wir belegen, dass Stickersysteme durch Mehrkopfautomaten mit höchstens 4 Köpfen simuliert werden können. Wir werden untersuchen, in welchem Maße eine Reduktion der Anzahl der Köpfe durch Einschränkung der Stickersysteme möglich ist. Als „Abfallprodukt“ der in dieser Arbeit durchgeführten Untersuchungen beantworten wir fast alle Fragen zu Beziehungen der Stickersprachfamilien untereinander und zu Abschlusseigenschaften der Stickersprachfamilien.

Neben dem Konzept der *Stickersysteme* werden in [PRS98] auch *Watson-Crick-Automaten*, *Insertion-Deletion-Systeme* und verschiedene Formen von *Splicingsystemen* vorgestellt (siehe [PRS98, Chapter 4, 5, 6, 7-11]). Alle diese Konzepte besitzen ihren Ursprung auf dem Gebiet des *DNA-Computing*, lassen aber Äquivalenzen oder Ähnlichkeiten zu bereits bekannten „klassischen“ Konzepten erkennen. So handelt es sich bei Insertion-Deletion-Systemen im Prinzip um Chomskygrammatiken $G = (T, N, S, P)$ mit Produktionen der Gestalt (S, w), (uv, uwv) und (uwv, uv) mit $u, v, w \in (T \cup (N \setminus \{S\}))^*$. Außerdem sind gemäß [PRS98, Lemma 5.8] Watson-Crick-Automaten äquivalent zu einfache (0.2)-Einwegmehrkopfautomaten. Es stellt sich zwangsläufig die Frage, ob es zu Stickersystemen ebenfalls ein äquivalentes oder ähnliches „klassisches“ Konzept gibt. Diese Frage werden wir mit Anmerkung 5.2 beantworten.

Im nächsten Kapitel werden die für die nachfolgenden Kapitel notwendigen Grundlagen geschaffen. Neben Definitionen und Vereinbarungen zu Stickersystemen, Chomskygrammatiken und Mehrkopfautomaten werden auch einige Basisresultate aufgeführt, auf die wir im weiteren Verlauf öfters zurückgreifen werden.

Im dritten Kapitel untersuchen wir die Beziehungen zwischen den Mehrkopfsprachfamilien und den Chomskysprachfamilien. Diese Analyse dient hauptsächlich der Vermeidung von Redundanzen in den darauf folgenden Kapiteln.

Den Untersuchungen zu Beziehungen der Stickersprachfamilien und Chomskysprachfamilien zueinander widmen wir uns im vierten Kapitel. Neben bereits bekannten Resultaten sind hier auch einige neue Ergebnisse zu finden, die nicht nur aus der Erweiterung des klassischen Konzeptes der Stickersysteme resultieren.

Das fünfte Kapitel widmet sich dem Zusammenhang zwischen Stickersystemen und Mehrkopfautomaten. Hier werden wir u.a. untersuchen, wie viele Köpfe ein (einfacher) Mehrkopfautomat besitzen muss, um Stickersprachen der verschiedenen Stickersprachfamilien akzeptieren zu können.

Die Komplexitätsuntersuchungen in Kapitel drei, vier und fünf gliedern sich jeweils in *Obere Schranken*, *Untere Schranken*, *Optimalität der oberen Schranken* und *Optimalität der unteren Schranken*.

Kapitel sechs dient der Zusammenfassung dieser Arbeit hinsichtlich beantworteter und weiterhin offener Fragen. Außerdem beschäftigen wir uns hier abschließend, auf der Grundlage der vorangehenden Untersuchungen, mit der Frage nach dem Nutzen des Konzeptes der Stickersysteme für die Theoretische Informatik.

Kapitel 2

Grundlagen

In diesem Kapitel wollen wir uns mit den für diese Arbeit notwendigen Definitionen und Vereinbarungen vertraut machen.

$\mathbb{N}$ ist die Menge der natürlichen Zahlen inklusive der 0. Die Menge aller Teilmengen einer Menge A bezeichnen wir mit $P(A)$. Die *leere Menge* wird mit $\emptyset$ bezeichnet.

Ein *Alphabet* ist eine nichtleere endliche Menge abstrakter Symbole. Die Elemente eines Alphabets werden *Buchstaben* genannt. Es sei Σ ein Alphabet. Ein *Wort* über Σ ist eine endliche geordnete Sequenz von Buchstaben aus Σ. Σ^* ist die Menge aller Wörter über dem Alphabet Σ inklusive dem *leeren Wort* ε. Es gelte $\Sigma^+ := \Sigma^* \setminus \{\varepsilon\}$. Eine *Sprache* L über dem Alphabet Σ ist eine Teilmenge von Σ^*. Das *Komplement* einer Sprache L wird mit L^{co} bezeichnet, es gilt $L^{co} := \{w \in \Sigma^* : w \notin L\}$. Seien $k, i \in \mathbb{N}$ und $w = a_1 a_2 \dots a_k \in \Sigma^*$ ein Wort, dann bezeichnet $w^R := a_k \dots a_2 a_1$ die *Umkehrung*, $|w| := k$ die *Länge* und $w[i] := a_i$ den i-ten *Buchstaben* von w für $1 \leq i \leq k$. Außerdem gelte $w[i] := \varepsilon$ für $i < 1$ oder $i > k$. Die *Konkatenation* zweier Wörter u und v mit $u = a_1 \dots a_k$ und $v = b_1 \dots b_m$ ist definiert durch $u \cdot v := a_1 \dots a_k b_1 \dots b_m$.

Es sei $k \in \mathbb{N}$. Dann ist Σ^k das k-fache Kreuzprodukt einer beliebigen nichtleeren Menge Σ. Die Elemente von Σ^k werden *Vektoren* genannt. Sei $\Sigma_1, \Sigma_2, \dots, \Sigma_k$ eine endliche Folge beliebiger nichtleerer Mengen und sei $x = (x_1, x_2, \dots, x_k) \in \Sigma_1 \times \Sigma_2 \times \dots \times \Sigma_k$, dann bezeichnet $x[i] := x_i$ die i-te *Komponente* von x für $1 \leq i \leq k$.

Sei $k \in \mathbb{N}$ und $\lambda : A^k \to B$ eine partielle Abbildung aus A^k in B. Falls keine abweichenden Vereinbarungen getroffen werden, ist die Fortsetzung $\lambda : P(A)^k \to P(B)$ definiert durch

$$\lambda(X_1, X_2, \dots, X_k) := \left\{ \lambda(x_1, x_2, \dots, x_k) : \begin{array}{l} x_i \in X_i \text{ für } 1 \leq i \leq k, \\ \lambda(x_1, x_2, \dots, x_k) \text{ ist definiert} \end{array} \right\}.$$

Die *Umkehrung* A^R und die *Konkatenation* $A \cdot B$ zweier Sprachen A und B sind damit eindeutig definiert.

2.1 Stickersysteme

In diesem Abschnitt wollen wir uns mit den für die Arbeit mit Stickersystemen und Stickersprachen notwendigen Definitionen und Vereinbarungen vertraut machen.

Es seien V ein Alphabet und $\rho \subseteq V \times V$ eine symmetrische binäre Relation. Dann sind $\binom{V^*}{V^*} := \left\{\binom{u}{v} : u, v \in V^*\right\}$ die Menge aller *Paare von Wörtern*[1] aus V^* und $\left[\begin{smallmatrix}V^*\\V^*\end{smallmatrix}\right]_\rho := \left\{\binom{u}{v} \in \binom{V^*}{V^*} : |u| = |v|, (u[i], v[i]) \in \rho\right\}$ die Menge aller *Paare zueinander komplementärer Wörter* aus V^*. Für $\binom{u}{v} \in \left[\begin{smallmatrix}V^*\\V^*\end{smallmatrix}\right]_\rho$ schreiben wir $\left[\begin{smallmatrix}u\\v\end{smallmatrix}\right]_\rho$. Die Konkatenation $\binom{x_1}{x_2} \cdot \binom{y_1}{y_2}$ lautet $\binom{x_1 \cdot y_1}{x_2 \cdot y_2}$. Analog schreiben wir für $\left[\begin{smallmatrix}x_1\\x_2\end{smallmatrix}\right]_\rho \cdot \left[\begin{smallmatrix}y_1\\y_2\end{smallmatrix}\right]_\rho$ auch $\left[\begin{smallmatrix}x_1 \cdot y_1\\x_2 \cdot y_2\end{smallmatrix}\right]_\rho$.

Die Menge der *Dominos* $W_\rho(V)$ ist definiert durch $W_\rho(V) := S_\rho(V) \cup LR_\rho(V)$ mit der Menge der *einfachen Dominos* $S_\rho(V) := \binom{V^*}{V^*}$, der Menge der *einsträngigen Dominos* $O_\rho(V) := \left\{\binom{u}{v} \in S_\rho(V) : u = \varepsilon \text{ oder } v = \varepsilon\right\}$, der Menge der *erweiterten Dominos* $E_\rho(V) := S_\rho(V) \setminus O_\rho(V)$, der Menge der *nicht-einfachen Dominos* $LR_\rho(V) := O_\rho(V) \times \left(\left[\begin{smallmatrix}V^*\\V^*\end{smallmatrix}\right]_\rho \setminus \left\{\left[\begin{smallmatrix}\varepsilon\\\varepsilon\end{smallmatrix}\right]_\rho\right\}\right) \times O_\rho(V)$ und der Menge der *vollständigen Dominos* $WK_\rho(V) := \left\{\binom{\varepsilon}{\varepsilon}\right\} \times \left(\left[\begin{smallmatrix}V^*\\V^*\end{smallmatrix}\right]_\rho \setminus \left\{\left[\begin{smallmatrix}\varepsilon\\\varepsilon\end{smallmatrix}\right]_\rho\right\}\right) \times \left\{\binom{\varepsilon}{\varepsilon}\right\}$. Das Domino $\binom{\varepsilon}{\varepsilon}$ identifizieren wir mit dem Symbol ε. Folglich können wir das Domino $\binom{\varepsilon}{\varepsilon}\left[\begin{smallmatrix}x\\y\end{smallmatrix}\right]_\rho\binom{\varepsilon}{\varepsilon}$ mit $\left[\begin{smallmatrix}x\\y\end{smallmatrix}\right]_\rho$ identifizieren.

Seien $x \in LR_\rho(V)$ und $y \in S_\rho(V)$, so bezeichnen wir die einzelnen Komponenten in der Regel mit x_1^t, x_1^b, x_2^t, x_2^b, x_3^t, x_3^b, y^t, y^b für $x = \left(\binom{x_1^t}{x_1^b}, \left[\begin{smallmatrix}x_2^t\\x_2^b\end{smallmatrix}\right]_\rho, \binom{x_3^t}{x_3^b}\right)$ und $y = \binom{y^t}{y^b}$. Außerdem ist $x_2 := \left[\begin{smallmatrix}x_2^t\\x_2^b\end{smallmatrix}\right]_\rho$ das *Mittelstück*, $x_1 := \binom{x_1^t}{x_1^b}$ der *linke* und $x_3 := \binom{x_3^t}{x_3^b}$ der *rechte Überhang*. Statt $x = (x_1, x_2, x_3)$ schreiben wir meistens $x = x_1x_2x_3$. Die Wörter y^t und $x^t := x_1^t \cdot x_2^t \cdot x_3^t$ bezeichnen wir als *oberen Strang*. Analog heißen y^b und $x^b := x_1^b \cdot x_2^b \cdot x_3^b$ *unterer Strang*. Die Buchstaben der einzelnen Komponenten nennen wir *Basen*.

Die *Struktur* eines Dominos x ist definiert durch die Abbildung $struct : W_\rho(V) \to W_{\{(\sharp,\sharp)\}}(\{\sharp\})$, wobei $struct(x)$ aus x hervorgeht, indem alle Basen durch die Base $\sharp$ ersetzt werden.

[1] $\binom{V^*}{V^*}$ ist somit lediglich eine andere Schreibweise für $V^* \times V^*$.

Die *Länge des Überhangs* eines Dominos ist wie folgt durch die Abbildung $d : W_\rho(V) \to \mathbb{N}$ definiert:

$$d(x) := \begin{cases} max\left\{|x_1^t|, |x_1^b|, |x_3^t|, |x_3^b|\right\} & \text{falls } x \in LR_\rho(V), \\ max\left\{|x^t|, |x^b|\right\} & \text{falls } x \in S_\rho(V). \end{cases}$$

Seien $x, y \in W_\rho(V)$, dann ist die *Verklebung* von x und y wie folgt durch die Abbildung $\mu_\rho : W_\rho(V) \times W_\rho(V) \to W_\rho(V)$ definiert:

$$\mu_\rho(x,y) := \begin{cases} x_1\left(x_2 \cdot \left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]_\rho \cdot y_2\right) y_3 & \text{falls } x \in LR_\rho(V), y \in LR_\rho(V), \\ & x_3 \cdot y_1 = \left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]_\rho, \\ x_1\left(x_2 \cdot \left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]_\rho\right) w & \text{falls } x \in LR_\rho(V), y \in S_\rho(V), \\ & x_3 \cdot y = \left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]_\rho \cdot w, w \in O_\rho(V), \\ w\left(\left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]_\rho \cdot x_2\right) x_3 & \text{falls } x \in S_\rho(V), y \in LR_\rho(V), \\ & x \cdot y_1 = w \cdot \left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]_\rho, w \in O_\rho(V), \\ x \cdot y & \text{falls } x \in S_\rho(V), y \in S_\rho(V), \\ \text{undefiniert} & \text{sonst.} \end{cases}$$

Aufgrund der Assoziativität von μ_ρ schreiben wir $x \cdot_\rho y$ statt $\mu_\rho(x, y)$.

Ein *Stickersystem* ist ein Konstrukt

$$\gamma = (V, \rho, A, D)$$

mit einem Alphabet V, einer symmetrischen binären Relation $\rho \subseteq V \times V$, einer endlichen Menge $A \subseteq LR_\rho(V)$ und einer endlichen Menge $D \subseteq W_\rho(V) \times W_\rho(V)$. Die Relation ρ wird *Komplementarität* von V genannt. Die Elemente von A werden als *Axiome* und die Elemente von D als *Regeln* bezeichnet.

Seien $x, y \in W_\rho(V)$, so schreiben wir $x \to_\gamma y$ falls es eine Regel $(u, v) \in D$ gibt mit $y = u \cdot_\rho x \cdot_\rho v$. Außerdem schreiben wir $x \to_\gamma^k y$ für $x = x_0 \to_\gamma x_1 \to_\gamma x_2 \to_\gamma \cdots \to_\gamma x_k = y$ mit $k \in \mathbb{N}$ und $x_i \in W_\rho(V)$ für $0 \leq i \leq k$ bzw. $x \to_\gamma^* y$ falls es ein solches k gibt und nennen dies eine *Ableitung* falls $x \in A$ bzw. eine *vollständig Ableitung* falls zusätzlich noch $y \in WK_\rho(V)$ gilt.

Es gelten $C^0(\gamma) := A$ und $C^k(\gamma) := \left\{y \in W_\rho(V) : \exists x \in C^{k-1}(\gamma) : x \to_\gamma y\right\}$ mit $k \in \mathbb{N}$ und $k \geq 1$. $C^*(\gamma) := \bigcup_{k \in \mathbb{N}} C^k(\gamma)$ ist die Menge der durch γ *erzeugten Dominos*, $LM(\gamma) := C^*(\gamma) \cap WK_\rho(V)$ ist die von γ *erzeugte Molekülsprache* und $L(\gamma) := \{x^t : x \in LM(\gamma)\}$ ist die von γ *erzeugte Sprache*.

Aufgrund der vorangehenden Definitionen gilt stets $\varepsilon \notin L(\gamma)$ für beliebige Stickersysteme $\gamma = (V, \rho, A, D)$. Wir erweitern nun die Definition eines Stickersystems, indem wir $\binom{\varepsilon}{\varepsilon} \in A$ erlauben, wobei dieses Axiom niemals in einer Ableitung verwendet werden darf. Es gilt $\varepsilon \in L(\gamma)$ gdw. $\binom{\varepsilon}{\varepsilon} \in A$.

Eine Regel $(u, v) \in D$ heißt *erweitert* falls mindestens eine der beiden Komponente ein erweitertes Dominos ist, *einfach* falls beide Komponenten einfache Dominos sind, *einsträngig* falls beide Komponenten einsträngige Dominos sind, *linksseitig* falls gilt $v = \varepsilon$, *rechtsseitig* falls gilt $u = \varepsilon$ und *einseitig* falls gilt $u = \varepsilon$ oder $v = \varepsilon$. Eine Ableitung $x_0 \to_\gamma^* x_k$ heißt *überhangbeschränkt durch* $d \in \mathbb{N}$ falls gilt $d(x_i) \leq d$ für $0 \leq i \leq k$. Ein Stickersystem $\gamma = (V, \rho, A, D)$ heißt *erweitert* falls es in D mindestens eine erweiterte Regel gibt, *einfach* falls alle Regeln in D einfach sind, *einseitig* falls alle Regel in D einseitig sind, *regulär* falls alle Regel in D rechtsseitig sind und *überhangbeschränkt* falls eine Konstante $d \in \mathbb{N}$ existiert, so dass es für jedes Domino $x \in LM(\gamma)$ eine durch d überhangbeschränkte Ableitung gibt.

Mit $ASL(n)$ bezeichnen wir die Familie der durch Stickersysteme erzeugten Sprachen. Die Einschränkung auf überhangbeschränkte Stickersysteme kennzeichnen wir durch Ersetzen von n durch b. Den Ausschluss erweiterter Stickersysteme kennzeichnen wir durch Voranstellen des Buchstabens c für *klassisch*. Einschränkungen auf einfache, einseitige, reguläre, einfache und einseitige bzw. einfache und reguläre Stickersysteme kennzeichnen wir durch Ersetzen von A durch S, O, R, SO bzw. SR.

Beobachtung 2.1.

$$XSL(cb) \subseteq XSL(cn), \qquad SRSL(x) \subseteq SOSL(x),$$
$$XSL(cn) \subseteq XSL(n), \qquad SOSL(x) \subseteq SSL(x),$$
$$XSL(cb) \subseteq XSL(b), \qquad SSL(x) \subseteq ASL(x),$$
$$XSL(b) \subseteq XSL(n), \qquad SOSL(x) \subseteq OSL(x),$$
$$OSL(x) \subseteq ASL(x),$$
$$SRSL(x) \subseteq RSL(x),$$
$$RSL(x) \subseteq OSL(x).$$

Dabei sei $X \in \{A, S, O, R, SO, SR\}$ *und* $x \in \{cb, b, cn, n\}$*, wobei* $XSL(cy)$ *in diesem Zusammenhang für* $cXSL(y)$ *steht.*[2]

Lemma 2.2 (vgl. [PRS98, Lemma 5.8])**.** *Für jedes Stickersystem* $\gamma = (V, \rho, A, D)$ *gibt es ein effektiv konstruierbares Stickersystem* $\gamma' = (V, \rho', A', D')$ *mit* $L(\gamma) = L(\gamma')$ *und* $\rho' = \{(x, x) : x \in V\}$*, das mit* γ *in den hier definierten Eigenschaften*[3] *der Regeln und Ableitungen übereinstimmt.*

Beweis. Der Beweis ist eine Übertragung und Verallgemeinerung des Komplementaritätslemmas für Watson-Crick-Automaten [PRS98, Lemma 5.8].

[2] Wir werden auch an anderen Stellen von dieser Notationsänderung gebrauch machen.

[3] Die Transformation von γ zu γ' erhält die Eigenschaften einfach, einseitig, regulär, überhangbeschränkt, ...

Gegeben sei ein beliebiges Stickersystem $\gamma = (V, \rho, A, D)$.

Die Abbbildung $\lambda_\rho : \binom{V^*}{V^*} \to P\left(\binom{V^*}{V^*}\right)$ ist wie folgt definiert:

$$\lambda_\rho\left(\binom{a}{b}\right) := \begin{cases} \left\{\binom{a}{a}\right\} & \text{falls } a \in V, b \in V, (a,b) \in \rho, \\ \left\{\binom{a}{\varepsilon}\right\} & \text{falls } a \in V, b = \varepsilon, \\ \left\{\binom{\varepsilon}{\varepsilon}\right\} & \text{falls } a = \varepsilon, b = \varepsilon, \\ \left\{\binom{\varepsilon}{c} : (c,b) \in \rho\right\} & \text{falls } a = \varepsilon, b \in V, \\ \text{undefiniert} & \text{sonst.} \end{cases}$$

Betrachten wir nun die Fortsetzung von λ_ρ auf $O_\rho(V)$ und $\left[\begin{smallmatrix} V^* \\ V^* \end{smallmatrix}\right]_\rho$ mit $\lambda_\rho(u \cdot v) := \lambda_\rho(u) \cdot \lambda_\rho(v)$, auf $E_\rho(V)$ mit $\lambda_\rho\left(\binom{u}{v}\right) := \lambda_\rho\left(\binom{u}{\varepsilon}\right) \cdot \lambda_\rho\left(\binom{\varepsilon}{v}\right)$, auf $LR_\rho(V)$ mit $\lambda_\rho(x_1x_2x_3) := \lambda_\rho(x_1) \times \lambda_\rho(x_2) \times \lambda_\rho(x_3)$ und auf Mengen von Dominos mit $\lambda_\rho(M) := \bigcup_{w \in M} \lambda_\rho(w)$, dann ist die Transformation λ_ρ von Dominos strukturerhaltend, im oberen Strang werden keine Basenumbenennungen vorgenommen und eine Base α des unteren Stranges geht genau dann in eine Base β über, falls β unter Respektierung der Komplementarität ρ im oberen Strang direkt über α stehen könnte bzw. falls es da bereits steht.

Sei $\gamma' = (V', \rho', A', D')$ das Stickersystem mit

$$\begin{aligned} V' &:= V, \\ \rho' &:= \{(x,x) : x \in V'\}, \\ A' &:= \lambda_\rho(A), \\ D' &:= \bigcup_{(u,v) \in D} \lambda_\rho(u) \times \lambda_\rho(v). \end{aligned}$$

Dann gilt $L(\gamma) = L(\gamma')$.

Man beweist dazu die Beziehung $\lambda_\rho(u \cdot_\rho v) = \lambda_\rho(u) \cdot_{\rho'} \lambda_\rho(v)$ für beliebige $u, v \in W_\rho(V)$ und zeigt damit, dass die Transformation λ_ρ bezüglich der Verklebungen $\cdot_\rho$ und $\cdot_{\rho'}$ ein Homomorphismus ist. Anschließend zeigt man mittels Induktion über $k \in \mathbb{N}$ die Beziehung $\lambda_\rho(C^k(\gamma)) = C^k(\gamma')$ und somit $\lambda_\rho(C^*(\gamma)) = C^*(\gamma')$. Da λ_ρ strukturerhaltend ist und keine Basenumbenennungen im oberen Strang vornimmt, folgt daraus $L(\gamma) = L(\gamma')$. □

Falls wir im weiteren Verlauf bei der Definition eines Stickersystems die Komplementaritätsrelation ρ weglassen, so meinen wir damit $\rho := \{(x,x) : x \in V\}$. In diesem Zusammenhang schreiben wir dann auch $\left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]$ statt $\left[\begin{smallmatrix} u \\ v \end{smallmatrix}\right]_\rho$, $x \cdot y$ statt $x \cdot_\rho y$ und $W(V)$, $S(V)$, $O(V)$, $E(V)$, $LR(V)$ statt $W_\rho(V)$, $S_\rho(V)$, $O_\rho(V)$, $E_\rho(V)$, $LR_\rho(V)$.

Lemma 2.3 ([PRS98, Theorem 4.6])**.**

$$cXSL(b) = cXSL(n). \ (X \in \{O, R, SO, SR\})$$

Beweis. Gemäß [PRS98, Theorem 4.6] ist jedes einseitige, nicht-erweiterte Stickersystem überhangbeschränkt. Unter Hinzunahme der Definition der Stickersprachfamilien folgen daraus $cOSL(b) = cOSL(n)$, $cRSL(b) = cRSL(n)$, $cSOSL(b) = cSOSL(n)$, $cSRSL(b) = cSRSL(n)$. Der Vollständigkeit halber wollen wir den Beweis hier kurz skizzieren.

Sei $\gamma = (V, \rho, A, D)$ ein einseitiges Stickersystem und sei $d \in N$ die maximale Länge der Überhänge aller in den Axiomen oder Regeln vorkommenden Dominos, dann lässt sich jede vollständige Ableitung von γ in eine äquivalente durch die Konstante d überhangbeschränkte Ableitung transformieren.

Sei σ eine vollständige Ableitung mit o.B.d.A. zuerst nur linksseitigen und danach nur rechtsseitigen Regelanwendungen. Jede einfache Regel ist, da es keine erweiterten Regeln gibt, einsträngig. Sei x das erste während diese Ableitung erzeugte Domino mit $d(x) > d$ und s die zuletzt angewendete Regel. Nach Definition von μ_ρ muss s einfach sein. Außerdem können, solange die Länge des Überhangs nicht wieder auf $\leq d$ verkleinert wurde, nur einfache Regeln angewendet werden. Da σ vollständig ist, gibt es Regelanwendungen, die die Länge des Überhangs wieder verkleinern. Sei r die erste die Länge des Überhangs verkleinernde Regel. Man kann sich überzeugen, dass sich die Anwendung der Regel r bis unmittelbar vor die Anwendung der Regel s verlagern lässt, ohne dass dabei das Ergebnis der Ableitung verändert oder die Verklebbarkeit beeinflusst wird. Die Länge des Überhangs wurde aber dadurch verkleinert. Durch mehrfache Anwendung dieses Verfahrens lässt sich die Ableitung σ in eine durch d überhangbeschränkte Ableitung transformieren. Dann ist γ aber überhangbeschränkt. □

2.2 Chomskygrammatiken

In diesem Abschnitt wollen wir kurz einige Definitionen, Vereinbarungen und Basisresultate zu Chomskygrammatiken wiederholen.

Eine *Chomskygrammatik* ist ein Konstrukt

$$G = (T, N, S, P)$$

mit zwei disjunkten Alphabeten T und N, einem Symbol $S \in N$ und einer endlichen Menge $P \subseteq V^+ \times V^*$. T ist die Menge der *Terminalsymbole*, N

die Menge der *Nichtterminalsymbole*, $V := N \cup T$ das *Arbeitsalphabet*, S das *Startsymbol* und P ist die Menge der *Produktionen*.

Seien $x, y \in V^*$, so schreiben wir $x \to_G y$ falls es eine Regel $(u, v) \in P$ gibt mit $x = r \cdot u \cdot s$ und $y = r \cdot v \cdot s$ für geeignete $r, s \in V^*$. Außerdem schreiben wir $x \to_G^k y$ für $x = x_0 \to_G x_1 \to_G x_2 \to_G \cdots \to_G x_k = y$ mit $k \in \mathbb{N}$ und $x_i \in V^*$ für $0 \leq i \leq k$ bzw. $x \to_G^* y$ falls es ein solches k gibt und nennen dies eine *Ableitung* falls $x = S$ bzw. eine *vollständig Ableitung* falls zusätzlich noch $y \in T^*$ gilt.

Es gelten $C^0(G) := \{S\}$ und $C^k(G) := \left\{y \in V^* : \exists x \in C^{k-1}(G) : x \to_G y\right\}$ mit $k \in \mathbb{N}$ und $k \geq 1$. $C^*(G) := \bigcup_{k \in \mathbb{N}} C^k(G)$ ist die Menge der durch G *erzeugten Wörter* und $L(G) := C^*(G) \cap T^*$ ist die von G *erzeugte Sprache*.

Eine Chomskygrammatik $G = (T, N, S, P)$ heißt *regulär* falls gilt $P \subseteq N \times (T^+ \cup (T^* \cdot N))$, *linear* falls gilt $P \subseteq N \times (T^+ \cup (T^* \cdot N \cdot T^*))$, *kontextfrei* falls gilt $P \subseteq N \times V^+$ und *nicht-verkürzend* falls für alle $(u, v) \in P$ gilt $|u| \leq |v|$. Außerdem lassen wir für alle hier definierten Chomskygrammatiken auch die Regel $(S, \varepsilon) \in P$ zu.

Analog zu [WW86, Section 4.1.1] bezeichnen wir mit *CS*, *CF*, *LIN*, *REG* die Familien der durch *nicht-verkürzende* bzw. *kontextsensitive*[4], *kontextfreie*, *lineare* und *reguläre* Chomskygrammatiken erzeugten Sprachen.

Beobachtung 2.4.

$$REG \subseteq LIN \subseteq CF \subseteq CS.$$

Lemma 2.5 ([WW86, Theorem 4.6])**.**

$$REG \subset LIN \subset CF \subset CS.$$

2.3 Mehrkopfautomaten

Analog zu den beiden vorangehenden Abschnitten, wollen wir nun das Konzept der Mehrkopfautomaten einführen bzw. wiederholen.

Seien $k \in \mathbb{N}$, $K := \{1, \ldots, k\}$ und $x, y \in \mathbb{N}^k$, dann ist y die *Kompression* von x, wir schreiben $y = comp(x)$, falls folgende Bedingungen erfüllt sind:

(1) $\forall i, j \in K : x[i] \leq x[j] \iff y[i] \leq y[j]$,

(2) $\forall i \in K, y[i] \neq 0 : \exists j \in K : y[j] = y[i] - 1$.

[4]Gemäß [WW86, Theorem 4.2] sind die durch *nicht-verkürzende* bzw. *kontextsensitive* Grammatiken erzeugte Sprachfamilien identisch.

Außerdem ist für $k \in \mathbb{N}$ die Menge aller k-Kompressionen definiert durch $COMP_k := \{comp(x) : x \in \mathbb{N}^k\}$.

Ein *Mehrkopfautomat* ist ein Konstrukt

$$\mathcal{A} = (X, Z, s, F, T)$$

mit einem Alphabet X, einer endlichen Menge Z, einem Symbol $s \in Z$, einer endlichen Menge $F \subseteq Z$ und einer endlichen Menge $T \subseteq Z \times V^k \times COMP_k \times Z \times M^k$ mit $V := X \cup \{\varepsilon\}$, $M := \{-1, 0, +1\}$ und $k \in \mathbb{N}$. X ist das *Eingabealphabet*, V das *Arbeitsalphabet*, Z die Menge der *Zustände*, s der *Startzustand*, F die Menge der *Finalzustände*, T die Menge der *Transitionen* und k die Anzahl der *Köpfe*. Einen Mehrkopfautomaten mit k Köpfen nennen wir auch *k-Mehrkopfautomat*.

$C(\mathcal{A}) := X^* \times Z \times \mathbb{N}^k$ ist die Menge der *Konfigurationen*[5] von $\mathcal{A}$. Eine Konfiguration $x \in C(\mathcal{A})$ heißt *initial* falls $x \in C_i(\mathcal{A}) := X^* \times \{s\} \times \{(1, \ldots, 1)\}$ bzw. *final* falls $x \in C_f(\mathcal{A}) := X^* \times F \times \mathbb{N}^k$. Seien $x, y \in C(\mathcal{A})$ mit $x = (w, z_x, \vec{h_x})$ und $y = (w, z_y, \vec{h_y})$, so schreiben wir $x \vdash_{\mathcal{A}} y$ und nennen y eine *Nachfolgekonfiguration* von x falls es eine Transition[6] $t = (z, \vec{v}, \vec{c}, q, \vec{m}) \in T$ gibt mit $z_x = z$, $\forall i \in K : w[\vec{h_x}[i]] = \vec{v}[i]$, $comp(\vec{h_x}) = \vec{c}$, $z_y = q$, $\forall i \in K : \vec{h_y}[i] = min\{|w| + 1, max\{0, \vec{h_x}[i] + \vec{m}[i]\}\}$. Außerdem schreiben wir $x \vdash_{\mathcal{A}}^u y$ für $x = x_0 \vdash_{\mathcal{A}} x_1 \vdash_{\mathcal{A}} x_2 \vdash_{\mathcal{A}} \cdots \vdash_{\mathcal{A}} x_u = y$ mit $u \in \mathbb{N}$ und $x_i \in C(\mathcal{A})$ für $0 \leq i \leq k$ bzw. $x \vdash_{\mathcal{A}}^* y$ falls es ein solches u gibt und nennen dies einen *Lauf* falls x initial bzw. einen *erfolgreichen Lauf* falls zusätzlich noch y final ist.

Es gelten $C^0(\mathcal{A}) := C_i(\mathcal{A})$ und $C^k(\mathcal{A}) := \{y \in C(\mathcal{A}) : \exists x \in C^{k-1}(\mathcal{A}) : x \vdash_{\mathcal{A}} y\}$ mit $k \in \mathbb{N}$ und $k \geq 1$. $C^*(\mathcal{A}) := \bigcup_{k \in \mathbb{N}} C^k(\mathcal{A})$ ist die Menge der von $\mathcal{A}$ *erreichbaren Konfigurationen*, $C_f^*(\mathcal{A}) := C^*(\mathcal{A}) \cap C_f(\mathcal{A})$ ist die Menge der von $\mathcal{A}$ *erreichbaren Finalkonfigurationen* und $L(\mathcal{A}) := \{c[1] : c \in C_f^*(\mathcal{A})\}$ ist die von $\mathcal{A}$ akzeptierte *Sprache*.

Ein k-Mehrkopfautomat $\mathcal{A} = (X, Z, s, F, T)$ heißt *einfach* falls für alle Transitionen $(z, \vec{v}, \vec{c}, q, \vec{m}) \in T$ und für alle $\vec{r} \in COMP_k$ gilt $(z, \vec{v}, \vec{r}, q, \vec{m}) \in T$, die relative Kopfpositionserkennung also nicht benötigt wird. Zur Vereinfachung lassen wir in diesem Fall die Komponente $\vec{c}$ bzw. $\vec{r}$ weg und schreiben $(z, \vec{v}, q, \vec{m}) \in T$.

Ein *Einwegmehrkopfautomat* ist ein Mehrkopfautomat mit $s + t$ Köpfen für beliebige $s, t \in \mathbb{N}$, der seine $s + t$ Köpfe gemeinsam an einer Position p posi-

[5] Eine Konfiguration ist gekennzeichnet durch die momentan zu bearbeitende Eingabe (Eingabewort), den aktuelle Zustand und die Positionen der k Köpfe.

[6] Eine Transition dient der Beschreibung von Zustandswechsel und Kopfpositionsänderungen. Ein Mehrkopfautomat wechselt in einen Zustand q und ändert die Kopfpositionen entsprechend $\vec{m}$ falls er sich in einem Zustand z befindet, an den aktuellen Positionen der Köpfe die Buchstaben $\vec{v}$ gelesen werden und die Köpfe die durch $\vec{c}$ beschriebenen relativen Positionen zueinander haben. Das Eingabewort w kann nicht verändert werden und die Köpfe können sich nicht über die Randmarkierung ε hinaus bewegen.

tioniert und anschließend Kopf 1 bis s nach links ($\vec{m}[i] \in \{-1, 0\}$) und Kopf $s+1$ bis $s+t$ nach rechts ($\vec{m}[i] \in \{0, +1\}$) bewegt. Sei w das Eingabewort, dann gelte $p = 1$ für $w = \varepsilon$, $p = 1$ für $s = 0$ und $p = |w|$ für $t = 0$, ansonsten sei p beliebig aus dem Intervall $[1, |w|]$. Wir nennen einen solchen Automaten mit s nach links laufenden Köpfen und t nach rechts laufenden Köpfen auch $(s.t)$-*Einwegmehrkopfautomat*.

Mit AHL_k bezeichnen wir die Familie der durch k-Mehrkopfautomaten akzeptierten Sprachen. Die Einschränkung auf einfache Mehrkopfautomaten kennzeichnen wir durch Ersetzen von A durch S. Für $s, t \in \mathbb{N}$ bezeichnen wir mit $OHL_{s.t}$ die Familie der durch $(s.t)$-Einwegmehrkopfautomaten akzeptierten Sprachen. Die Einschränkung auf einfache Einwegmehrkopfautomaten kennzeichnen wir durch Voranstellung des Buchstabens S. Durch Vereinigung über k, s oder t erhalten wir die Mehrkopfsprachfamilien AHL_*, SHL_*, $OHL_{s.*}$, $SOHL_{s.*}$, $OHL_{*.t}$, $SOHL_{*.t}$, $OHL_{*.*}$ und $SOHL_{*.*}$.[7]

Anmerkung 2.6. *Das hier eingeführte Konzept der Mehrkopfautomaten mit relativer Kopfpositionserkennung ist äquivalent zu dem bereits bekannten Konzept der Mehrkopfautomaten mit der Fähigkeit zur Erkennung des Zusammentreffens von Köpfen (sensing, coincidence detection).*

Beobachtung 2.7.

$$\begin{aligned} XHL_k &\subseteq XHL_{k+1}, & SOHL_{s.t} &\subseteq SHL_{s+t}, \\ XHL_k &\subseteq XHL_*, & SHL_s &\subseteq AHL_s, \\ YHL_{s.t} &\subseteq YHL_{s+1.t}, & SOHL_{s.t} &\subseteq OHL_{s.t}, \\ YHL_{s.t} &\subseteq YHL_{s.t+1}, & OHL_{s.t} &\subseteq AHL_{s+t}. \\ YHL_{s.t} &\subseteq YHL_{*.t}, \\ YHL_{s.t} &\subseteq YHL_{s.*}, \end{aligned}$$

Dabei sei $k \in \mathbb{N}$; $s, t \in \mathbb{N} \cup \{*\}$; $X \in \{A, S\}$ und $Y \in \{O, SO\}$. Außerdem gelte in diesem Zusammenhang $* + x = x + * = *$ für $x \in \mathbb{N} \cup \{*\}$.

Lemma 2.8. $YHL_{0.k} = YHL_{k.0}$. $(k \in \mathbb{N},\ Y \in \{O, SO\})$

(Ohne Beweis. Die genannten Beziehungen können als allgemein bekannt angenommen werden. Der Beweis erfolgt durch Rückwärtssimulation.)

Auf der Grundlage von Lemma 2.8 definieren wir $YHL_t := YHL_{0.t}$ für $Y \in \{O, SO\}$ und $t \in \mathbb{N} \cup \{*\}$.

[7] Die Sprachfamilie SHL_k wird in [WW86] mit *2:k−NFA* (oder *2−NFA* für $k = 1$) bezeichnet. SHL_* entspricht der Sprachfamilie *2:multi−NFA*. Analog ersetze man $SOHL_k$ durch *1:k−NFA* (bzw. *1−NFA* für $k = 1$) und $SOHL_*$ durch *1:multi−NFA*. Für [Mon80] ersetze man SHL_k durch $NH(k)$ und für [YR78] $SOHL_k$ durch R_k.

Lemma 2.9 ([Mon80], [YR78])**.**

$$XHL_k \subset XHL_{k+1} \subset XHL_*. \ (k \in \mathbb{N},\ X \in \{A, S, O, SO\})$$

Beweis. Zum Beweis von $SHL_k \subset SHL_{k+1}$ und $SOHL_k \subset SOHL_{k+1}$ sei auf [Mon80] und [YR78] verwiesen. Der Beweis der Inklusionen $AHL_k \subset AHL_{k+1}$ und $OHL_k \subset OHL_{k+1}$ erfolgt analog.

Die Inklusion $XHL_{k+1} \subset XHL_*$ folgt aus $XHL_{k+1} \subset XHL_{k+2} \subseteq XHL_*$. □

Lemma 2.10 ([YR78, Introduction])**.** $OHL_3 \not\subseteq SOHL_*$.

Aus Lemma 2.10 folgt $SOHL_k \subset OHL_k$ für $k \geq 3$. Vermutlich gilt auch $SHL_k \subset AHL_k$ für $k \geq 2$ (und $SOHL_k \subset OHL_k$ für $k = 2$). Mit einem zusätzlichen Kopf lässt sich dieser Rückstand aber wieder aufholen.

Satz 2.11. *$AHL_k \subseteq SHL_{k+1}$. ($k \in \mathbb{N}$)*

Beweis. Dieses Resultat kann als allgemein bekannt angenommen werden. Der Vollständigkeit halber geben wir hier einen Beweis an.

Sei $\mathcal{A} = (X, Z, s, F, T)$ ein k-Mehrkopfautomat, dann arbeitet der einfache $(k+1)$-Mehrkopfautomat $\mathcal{B} = (X, Z', s', F', T')$ wie folgt:

$\mathcal{B}$ besitzt einen *realen* und k *virtuelle Köpfe*.[8] Kopf $k+1$ realisiert den realen Kopf. Die Köpfe 1 bis k realisieren Zähler für den Bereich $-(|w|+1), \ldots, +(|w|+1)$ mit w als Eingabewort. [Der Betrag wird durch die Kopfposition kodiert. Das Vorzeichen wird im Zustand gemerkt.] Diese k Zählerstände geben stets die relativen Positionen der virtuellen Köpfe 1 bis k zu dem realen Kopf an. Hat zum Beispiel der Zähler 3 den Wert -7, so befindet sich der virtuelle Kopf 3 sieben Zeichen links von der aktuellen Position des realen Kopfes. Zu jedem Zeitpunkt kann der Automat durch Lesen der linken Randmarkierung ε mit den Köpfen 1 bis k bzw. durch Auswerten der gemerkten Vorzeicheninformationen erkennen, ob sich der reale Kopf an der Position eines virtuellen Kopfes befindet oder welche virtuellen Köpfe sich links bzw. rechts von der aktuellen Position befinden. Außerdem ist er in der Lage, die Kopfpositionsinformationen für beliebige (gültige) Bewegungen des realen oder der virtuellen Köpfe korrekt zu pflegen.

Initialisierung

> Der reale Kopf und die k virtuellen Köpfe befinden sich an der absoluten Position 1. (Kopf 1 bis k befinden sich an Position 0, Kopf $k+1$ an Position 1.) Der gemerkte Zustand des Automaten $\mathcal{A}$ ist der Startzustand s.

[8] Die Idee der realen und virtuellen Köpfe stammt aus [WW86, Theorem 13.2(5,space)].

Am Ende dieser Phase wird in die Akzeptierungsphase gewechselt falls der reale Kopf das Symbol ε liest und $s \in F$ gilt. Ansonsten wird in die Simulationsphase gewechselt.

Simulation

Ein Schritt des Automaten $\mathcal{A}$ wird durch $\mathcal{B}$ simuliert, indem sich der reale Kopf einmal durch das komplette Eingabewort bewegt. Dadurch läuft der reale Kopf genau einmal über jede Position der virtuellen Köpfe und sammelt dabei sowohl die Buchstaben an den entsprechenden Positionen als auch Informationen über die relativen Positionen der virtuellen Köpfe zueinander ein. Auf der Grundlage dieser Informationen, zusammen mit dem zuletzt gemerkten Zustand, wird nun eine passende Transition von $\mathcal{A}$ gewählt. Entsprechend dieser Transition werden die Positionen der virtuellen Köpfe angepasst und der neue Zustand gemerkt. Anschließend wird der reale Kopf wieder an der absoluten Position 1 positioniert. Am Ende dieser Phase wird in die Akzeptierungsphase oder wieder an den Anfang der Simulationsphase gewechselt.

Akzeptierung

Die Eingabe wird akzeptiert, wenn der gemerkte Zustand aus der Menge F ist.

Es sei $\lambda : C(\mathcal{B}) \to C(\mathcal{A})$ eine Funktion, die aus einer Konfiguration des Automaten $\mathcal{B}$ eine Konfiguration des Automaten $\mathcal{A}$ berechnet. Dazu werden das Eingabewort und der gemerkte Zustand extrahiert und die absoluten Kopfpositionen der k virtuellen Köpfe aus dem Zustand und den Kopfpositionen der $k+1$ Köpfe errechnet. Man beweist nun mittels Induktion über die Anzahl der Laufschritte die Beziehung $\lambda(C^*(\mathcal{B})) = C^*(\mathcal{A})$ und zeigt damit, dass jeder Lauf des Automaten $\mathcal{A}$ durch den Automaten $\mathcal{B}$ simuliert werden kann und andererseits jeder Lauf des Automaten $\mathcal{B}$ die Simulation eines Laufes des Automaten $\mathcal{A}$ darstellt. Unter Einbeziehung der Akzeptierungsbedingung folgt daraus die Beziehung $L(\mathcal{B}) = L(\mathcal{A})$. □

Kapitel 3

Mehrkopfautomaten und Chomskygrammatiken

In den nachfolgenden Kapiteln untersuchen wir die Beziehungen zwischen Stickersystemen und Chomskygrammatiken bzw. Stickersystemen und Mehrkopfautomaten. Zuvor ist es aber nötig, einige Beziehungen zwischen Mehrkopfautomaten und Chomskygrammatiken zu wiederholen.

Tabelle 3.1 auf Seite 30 gibt einen Überblick über die zwischen den Mehrkopfsprachfamilien und den Chomskysprachfamilien geltenden Beziehungen auf der Grundlage von Abschnitt 2.2, 2.3 und 3.x. Vermutlich gilt zusätzlich die Beziehung $CF \not\subseteq AHL_*$. Damit könnte in der Übersicht jedes Vorkommen von $\not\subseteq$ durch $\not\subseteq, \not\supseteq$ ersetzt werden.

Ein möglicher Kandidat für eine Sprache L mit $L \in CF$ und $L \notin AHL_*$ ist die zweiklammerige Dycksprache $D_2 := L(G)$ mit $G = (\{[,], <, >\}, \{S\}, S, P)$ und den Produktionen $(S, [S])$, $(S, <S>)$, (S, SS) und (S, ε). Ein Beweis dieser Vermutung steht jedoch noch aus.

3.1 Obere Schranken

In diesem Abschnitt wiederholen bzw. folgern wir die Inklusionen $AHL_1 \subseteq REG$, $SOHL_{1.1} \subseteq LIN$ und $AHL_* \subset CS$ und geben somit obere Schranken für die Komplexität ausgewählter Mehrkopfsprachfamilien an.

Lemma 3.1. $AHL_1 \subseteq REG$.

(Ohne Beweis. Wegen $COMP_1 = \{(0)\}$ gilt $AHL_1 \subseteq SHL_1$. Die Inklusion $SHL_1 \subseteq REG$ ist allgemein bekannt.)

Lemma 3.2 (vgl. Lemma 3.5)**.** $OHL_{1.1} \subseteq LIN$.

(Ohne Beweis. Die genannte Inklusion kann als allgemein bekannt angenommen werden. Der Beweis ist im Prinzip eine Umkehrung des Beweises von Lemma 3.5.)

Lemma 3.3 (Satz 2.11 + [WW86])**.** $AHL_* \subset CS$.

Beweis. Es gilt $AHL_* \subseteq SHL_* = NL \subset NLINSPACE = CS$ gemäß Satz 2.11, [WW86, Theorem 13.2(8,space)], [WW86, Section 22.3][1] und [WW86, Theorem 12.15]. □

3.2 Untere Schranken

In diesem Abschnitt wiederholen bzw. zeigen wir die Inklusionen $REG \subseteq SOHL_1$ und $LIN \subseteq SOHL_{1.1}$ und geben somit untere Schranken für die Komplexität ausgewählter Mehrkopfsprachfamilien an.

Lemma 3.4. $REG \subseteq SOHL_1$.

(Ohne Beweis. Das Resultat ist allgemein bekannt.)

Lemma 3.5. $LIN \subseteq SOHL_{1.1}$.

Beweis. Dieses Resultat kann als allgemein bekannt angenommen werden. Der Vollständigkeit halber skizzieren wir hier kurz einen entsprechenden Beweis.

Es sei $G = (N, T, S, P)$ eine lineare Chomskygrammatik mit o.B.d.A. nur Regeln der Gestalt $N \times (T \cdot N \cdot T)$, $N \times (T \cdot N)$, $N \times (N \cdot T)$, $N \times T$ und $\{(S, \varepsilon)\}$.

[1] Wir werden auf die beiden Sprachfamilien *NL* und *NLINSPACE* in dieser Arbeit nicht genauer eingehen, da wir diese nur hier kurz benötigen. Es handelt sich hierbei um die Familien von Sprachen, die von Turingmaschinen mit einem Zwei-Weg-Eingabeband und einem logarithmisch bzw. linear platzbeschränkten Arbeitsband akzeptiert werden können.

Dann sei $\mathcal{A} = (T, Z, s_0, F, R)$ der einfache (1.1)-Einwegmehrkopfautomat mit[2]

$$
\begin{aligned}
Z &:= N \cup \{s_0, s_1, f\} \text{ mit } s_0, s_1, f \notin N, \\
F &:= \{f\}, \\
R &:= \left\{\left(s_0, \begin{pmatrix}\varepsilon\\ \varepsilon\end{pmatrix}, f, \begin{pmatrix}0\\0\end{pmatrix}\right) : (S, \varepsilon) \in P\right\} \\
&\cup \left\{\left(s_0, \begin{pmatrix}u\\ u\end{pmatrix}, s_1, \begin{pmatrix}0\\0\end{pmatrix}\right) : u \in T\right\} \\
&\cup \left\{\left(s_1, \begin{pmatrix}u\\ u\end{pmatrix}, s_1, \begin{pmatrix}+1\\+1\end{pmatrix}\right) : u \in T\right\} \\
&\cup \left\{\left(s_1, \begin{pmatrix}u\\ u\end{pmatrix}, X, \begin{pmatrix}-1\\+1\end{pmatrix}\right) : \exists (X, u) \in P; X \in N; u \in T\right\} \\
&\cup \left\{\left(Y, \begin{pmatrix}u\\ v\end{pmatrix}, X, \begin{pmatrix}-1\\+1\end{pmatrix}\right) : \exists (X, uYv) \in P; X, Y \in N; u, v \in T\right\} \\
&\cup \left\{\left(Y, \begin{pmatrix}u\\ v\end{pmatrix}, X, \begin{pmatrix}-1\\0\end{pmatrix}\right) : \exists (X, uY) \in P; X, Y \in N; u, v \in T\right\} \\
&\cup \left\{\left(Y, \begin{pmatrix}u\\ v\end{pmatrix}, X, \begin{pmatrix}0\\+1\end{pmatrix}\right) : \exists (X, Yv) \in P; X, Y \in N; u, v \in T\right\} \\
&\cup \left\{\left(S, \begin{pmatrix}\varepsilon\\ \varepsilon\end{pmatrix}, f, \begin{pmatrix}0\\0\end{pmatrix}\right)\right\}.
\end{aligned}
$$

Man kann aus jedem erfolgreichen Lauf von $\mathcal{A}$ eine äquivalente vollständige Ableitung von G extrahieren. Außerdem kann man aus jeder vollständigen Ableitung von G einen äquivalenten erfolgreichen Lauf von $\mathcal{A}$ konstruieren. Daraus folgt $L(\mathcal{A}) = L(G)$ und somit $LIN \subseteq SOHL_{1.1}$. □

3.3 Optimalität der oberen Schranken

In diesem Abschnitt wiederholen wir die Beziehung $SOHL_2 \not\subseteq CF$ und belegen somit für fast alle Mehrkopfsprachfamilien die Optimalität der oberen Schranke CS.

Lemma 3.6. $SOHL_2 \not\subseteq CF$.

Beweis. Dieses Resultat ist allgemein bekannt. Der Vollständigkeit halber skizzieren wir hier kurz einen entsprechenden Beweis.

[2]Hinweis: $\binom{x_1}{x_2}$ steht hier für den Vektor (x_1, x_2) und nicht für ein einfaches Domino.

Es sei $\mathcal{A} = (X, Z, z_a, \{f\}, T)$ der einfache (0.2)-Einwegmehrkopfautomat mit

$$\begin{aligned}
X &:= \{a, b, c\}, \\
Z &:= \{z_a, z_b, z_c, f\}, \\
T &:= \left\{\left(z_a, \begin{pmatrix} a \\ a \end{pmatrix}, z_a, \begin{pmatrix} +1 \\ 0 \end{pmatrix}\right), \left(z_a, \begin{pmatrix} b \\ a \end{pmatrix}, z_b, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)\right\} \\
&\cup \left\{\left(z_b, \begin{pmatrix} b \\ a \end{pmatrix}, z_b, \begin{pmatrix} +1 \\ +1 \end{pmatrix}\right), \left(z_b, \begin{pmatrix} c \\ b \end{pmatrix}, z_c, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)\right\} \\
&\cup \left\{\left(z_c, \begin{pmatrix} c \\ b \end{pmatrix}, z_c, \begin{pmatrix} +1 \\ +1 \end{pmatrix}\right), \left(z_c, \begin{pmatrix} \varepsilon \\ c \end{pmatrix}, f, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)\right\} \\
&\cup \left\{\left(z_a, \begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, f, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)\right\}.
\end{aligned}$$

Es sei $L_{abc} := \{a^n b^n c^n : n \in \mathbb{N}\}$. Zu jedem Wort $w \in L_{abc}$ lässt sich ein passender erfolgreicher Lauf von $\mathcal{A}$ angeben. Man kann außerdem zeigen, dass jeder erfolgreiche Lauf von $\mathcal{A}$ ein Lauf auf einem Wort $w \in L_{abc}$ ist. Somit gilt $L(\mathcal{A}) = L_{abc}$. Man kann sich leicht mittels des Pumpinglemmas für kontextfreie Sprachen (siehe [HU79, Lemma 6.1]) überzeugen, dass gilt $L_{abc} \notin CF$. Wegen $L(\mathcal{A}) \in SOHL_2$ gilt somit $SOHL_2 \not\subseteq CF$. □

3.4 Optimalität der unteren Schranken

In diesem Abschnitt wiederholen bzw. zeigen wir die Beziehungen $LIN \not\subseteq OHL_*$ und $CF \not\subseteq OHL_{*,*}$ und belegen somit für ausgewählte Mehrkopfsprachfamilien, dass die angegebenen unteren Schranken optimal sind.

Lemma 3.7 ([WW86, Theorem 13.5(3)])**.** $LIN \not\subseteq OHL_*$.

Beweis. Es gilt $S_1 := \{w \in \{a, b\}^* : w = w^R\} \in LIN$. Gemäß [WW86, Theorem 13.5] gilt $S_1 \notin SOHL_*$. Der Beweis der Beziehung $S_1 \notin OHL_*$ erfolgt analog. □

Lemma 3.8 (vgl. Lemma 3.7)**.** $CF \not\subseteq OHL_{*,*}$.

Beweis. Es gilt $S_2 := S_1 \cdot S_1 = \left\{v \cdot w \in \{a, b\}^* : v = v^R, w = w^R\right\} \in CF$. Analog zu $S_1 \notin OHL_*$ gemäß Lemma 3.7 gilt $S_2 \notin OHL_{*,*}$. Dabei wird hauptsächlich ausgenutzt, dass sich bei jedem Lauf auf einem Wort $v \cdot w$ eines der beiden Teilwörter v und w vollständig auf einer Seite befindet. □

Lemma 3.9. $SHL_2 \not\subseteq OHL_{*,*}$.

Beweis. Dieses Resultat kann unter Verwendung von Lemma 3.8 als allgemein bekannt angenommen werden. Der Vollständigkeit halber skizzieren wir hier kurz einen entsprechenden Beweis.

Es sei $\mathcal{A} = (X, Z, z_a, \{f\}, T)$ der einfache 2-Mehrkopfautomat mit

$$\begin{aligned}
X &:= \{a, b\}, \\
Z &:= \{z_a, z_b, z_c, z_d, f\}, \\
T &:= \left\{\left(z_a, \begin{pmatrix} x \\ y \end{pmatrix}, z_a, \begin{pmatrix} 0 \\ +1 \end{pmatrix}\right) : x, y \in X\right\} \\
&\cup \left\{\left(z_a, \begin{pmatrix} x \\ y \end{pmatrix}, z_b, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right) : x, y \in X\right\} \\
&\cup \left\{\left(z_b, \begin{pmatrix} x \\ x \end{pmatrix}, z_b, \begin{pmatrix} +1 \\ -1 \end{pmatrix}\right) : x \in X\right\} \\
&\cup \left\{\left(z_b, \begin{pmatrix} x \\ \varepsilon \end{pmatrix}, z_c, \begin{pmatrix} 0 \\ +1 \end{pmatrix}\right) : x \in X\right\} \\
&\cup \left\{\left(z_c, \begin{pmatrix} x \\ y \end{pmatrix}, z_c, \begin{pmatrix} 0 \\ +1 \end{pmatrix}\right) : x, y \in X\right\} \\
&\cup \left\{\left(z_c, \begin{pmatrix} x \\ \varepsilon \end{pmatrix}, z_d, \begin{pmatrix} 0 \\ -1 \end{pmatrix}\right) : x \in X\right\} \\
&\cup \left\{\left(z_d, \begin{pmatrix} x \\ x \end{pmatrix}, z_d, \begin{pmatrix} +1 \\ -1 \end{pmatrix}\right) : x \in X\right\} \\
&\cup \left\{\left(z_d, \begin{pmatrix} \varepsilon \\ x \end{pmatrix}, f, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right) : x \in X\right\} \\
&\cup \left\{\left(z_a, \begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, f, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right), \left(z_b, \begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, f, \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)\right\}.
\end{aligned}$$

Es sei $S_2 := \left\{v \cdot w \in \{a, b\}^* : v = v^R, w = w^R\right\}$. Zu jedem Wort $w \in S_2$ lässt sich ein passender erfolgreicher Lauf von $\mathcal{A}$ angeben. Man kann außerdem zeigen, dass jeder erfolgreiche Lauf von $\mathcal{A}$ ein Lauf auf einem Wort $w \in S_2$ ist. Somit gilt $S_2 = L(\mathcal{A}) \in \mathit{SHL}_2$. Unter Hinzunahme von $S_2 \notin \mathit{OHL}_{*,*}$ gemäß Lemma 3.8 folgt $\mathit{SHL}_2 \not\subseteq \mathit{OHL}_{*,*}$. □

	REG	LIN	CF	CS
AHL_*	$\supset$	$\supset$	$\not\subseteq$	$\subset$[3]
AHL_2	$\supset$	$\supset$	$\not\subseteq$	$\subset$
AHL_1	$=$	$\subset$	$\subset$	$\subset$
SHL_*	$\supset$	$\supset$	$\not\subseteq$	$\subset$
SHL_2	$\supset$	$\supset$	$\not\subseteq$	$\subset$
SHL_1	$=$	$\subset$	$\subset$	$\subset$
$OHL_{*.*}$	$\supset$	$\supset$	$\not\subseteq, \not\supseteq$	$\subset$
$OHL_{0.*}$	$\supset$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$OHL_{0.2}$	$\supset$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$OHL_{1.1}$	$\supset$	$=$	$\subset$	$\subset$
$OHL_{0.1}$	$=$	$\subset$	$\subset$	$\subset$
$SOHL_{*.*}$	$\supset$	$\supset$	$\not\subseteq, \not\supseteq$	$\subset$
$SOHL_{0.*}$	$\supset$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$SOHL_{0.2}$	$\supset$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$SOHL_{1.1}$	$\supset$	$=$	$\subset$	$\subset$
$SOHL_{0.1}$	$=$	$\subset$	$\subset$	$\subset$

Tabelle 3.1: Beziehungen zwischen Mehrkopfsprachfamilien und Chomskysprachfamilien

[3]Dieser Eintrag steht für $AHL_* \subset CS$.

Kapitel 4

Stickersysteme und Chomskygrammatiken

Die Fähigkeiten der Stickersysteme lassen sich am besten durch Vergleiche der Stickersprachfamilien untereinander und mit anderen Sprachfamilien beurteilen. Da es sich bei Stickersystemen um eine spezielle Form von Grammatiken handelt, werden in anderen Arbeiten vorrangig die Beziehungen von $cASL(n)$, $cASL(b)$, $cSSL(n)$, $cSSL(b)$, $cOSL(n)$, $cOSL(b)$, $cRSL(n)$, $cRSL(b)$, $cSOSL(n)$, $cSOSL(b)$, $cSRSL(n)$, $cSRSL(b)$ untereinander und mit den Chomskysprachfamilien CS, CF, LIN und REG untersucht. In diesem Kapitel wollen wir neben der Wiederholung bereits bekannter Ergebnisse unseren Beitrag zu diesen Untersuchungen leisten.

Tabelle 4.1 auf Seite 41 gibt einen Überblick über die zwischen den Stickersprachfamilien und den Chomskysprachfamilien geltenden Beziehungen.

Die Tabellen 4.2, 4.3, 4.4 und 4.5 auf Seite 42ff geben einen Überblick über die zwischen den Stickersprachfamilien untereinander geltenden Beziehungen. Vermutlich gelten zusätzlich die Beziehungen $SRSL(b) \not\subseteq cSSL(n)$, $SRSL(n) \not\subseteq cASL(n)$ und $SOSL(n) \not\subseteq RSL(n)$. Dadurch könnte in der Übersicht jedes Vorkommen von $\subseteq$ durch $\subset$ und jedes Vorkommen von $\not\supseteq$ durch $\not\subseteq, \not\supseteq$ ersetzt werden. Ein Beweis dieser Vermutung steht jedoch noch aus.

Tabelle 4.6 auf Seite 46 gibt einen Überblick über die Abschlusseigenschaften der Stickersprachfamilien bezüglich Vereinigung, Durchschnitt, Komplementbildung, Umkehrung, Konkatenation und Schnitt mit regulären Chomskysprachen. Diese Aussagen sind lediglich Nebenprodukte der in diesem Kapitel durchgeführten Untersuchungen und daher unvollständig.

Die Tabellen 4.1, 4.2, 4.3 und 4.6 basieren auf Abschnitt 2.1, 2.2, 4.x und den Abschlusseigenschaften der Chomskysprachfamilien (siehe z.B. [PRS98, Table 3.1]).

4.1 Obere Schranken

In diesem Abschnitt wiederholen bzw. zeigen wir die Inklusionen $OSL(b) \subseteq REG$, $ASL(b) \subseteq LIN$ und $ASL(n) \subset CS$ und geben somit obere Schranken für die Komplexität ausgewählter Stickersprachfamilien an.

Lemma 4.1 ([PRS98, Theorem 4.1])**.** *$OSL(b) \subseteq REG$.*

Beweis. Zum Beweis von $cOSL(b) \subseteq REG$ sei auf [PRS98, Theorem 4.1] verwiesen. Der Beweis kann ohne jegliche Modifikation auf $OSL(b) \subseteq REG$ übertragen werden. □

Lemma 4.2 ([PRS98, Theorem 4.3])**.** *$ASL(b) \subseteq LIN$.*

Beweis. Zum Beweis von $cASL(b) \subseteq LIN$ sei auf [PRS98, Theorem 4.3] verwiesen. Der Beweis kann ohne jegliche Modifikation auf $ASL(b) \subseteq LIN$ übertragen werden. □

Anmerkung 4.3 ([PRS98, Lemma 4.3])**.** *Da Stickersysteme keinerlei Löschoperationen besitzen, also einmal erzeugte Basen nicht wieder entfernt werden können, ist es laut [PRS98, Lemma 4.3] leicht zu zeigen, dass sich Stickersysteme (ohne erweitere Regeln) durch kontextsensitive Chomskygrammatiken simulieren lassen. Somit gilt $cASL(n) \subseteq CS$ und offensichtlich auch $ASL(n) \subseteq CS$.*

Korollar 4.4. *$ASL(n) \subset CS$.*

Beweis. Es gelten die Inklusionen $ASL(n) \subseteq OHL_{2.2} \subset AHL_4 \subset AHL_* \subset CS$ gemäß Satz 5.1, Beobachtung 2.7, Lemma 3.9, Lemma 2.9 und Lemma 3.3. □

Mit Korollar 4.4 haben wir eine offenen Fragen aus [PRS98], nämlich die Frage nach der Echtheit der Inklusion $cASL(n) \subseteq CS$, beantwortet.

Als Folgerung aus der bekannten Beziehung $ASL(n) \subseteq CS$ in Kombination mit Korollar 4.19 erhalten wir ebenfalls $ASL(n) \subset CS$. Demnach ist $S_2 := \{v \cdot w \in \{a,b\}^* : v = v^R, w = w^R\}$ eine der vielen Sprachen, die zwar eine Chomskysprache, aber keine Stickersprache ist.

4.2 Untere Schranken

In diesem Abschnitt wiederholen wir die Inklusionen $REG \subseteq cRSL(b)$ und $LIN \subseteq cASL(b)$.

Lemma 4.5 ([PRS98, Theorem 4.4])**.** $REG \subseteq cRSL(b)$.

Lemma 4.6 ([PRS98, Theorem 4.5])**.** $LIN \subseteq cASL(b)$.

Aus $REG \subseteq cRSL(b) \subseteq OSL(b) \subseteq REG$ gemäß Lemma 4.5, Beobachtung 2.1 und Lemma 4.1 folgen $cOSL(b) = OSL(b) = RSL(b) = cRSL(b)$. An dieser Stelle wollen wir einen alternativen Beweis für $cOSL(b) = OSL(b)$ (bzw. $cRSL(b) = RSL(b)$) skizzieren, der nicht auf die genannten Lemmata zurückgreifen muss.

Beweis. Gemäß Beobachtung 2.1 gilt $cOSL(b) \subseteq OSL(b)$. Somit genügt es $OSL(b) \subseteq cOSL(b)$ zu zeigen: Sei $\gamma = (V, A, D)$ ein einseitiges, überhangbeschränktes Stickersystem mit der Überhangslängenbeschränkung $d \in \mathbb{N}$ und sei σ eine vollständige Ableitung. Da γ einseitig ist und somit der Aufbau der linken und rechten Seite unabhängig voneinander erfolgt, können wir o.B.d.A. annehmen, dass zuerst nur linksseitige und dannach nur rechtseitige Regeln zum Einsatz kommen. Kommt in dieser Ableitung eine erweiterte Regel zum Einsatz, so können die beiden Stränge des erweiterten Dominos wegen der Längenbeschränkung des Überhangs nur in beschränktem Maße gegeneinander „verschoben“ werden. Entscheidend dabei ist, dass es höchstens $2d + 1$ verschiedene Möglichkeiten der gegenseitigen Verschiebung gibt, wobei die Anfänge der beiden Stränge höchstens einen Abstand von d Basen zueinander haben. Nach spätestens $2d - 1$ Regelanwendungen überlappen sich die beiden Stränge. Wir fügen γ nun eine neue einseitige Regel hinzu, die entsprechend der gerade geführten Betrachtung aus der Verklebung bereits existierender Regeln und, falls das noch möglich/nötig ist, mittels geeigneter Verschiebung der beiden Stränge und anschließender Umwandlung in eine nicht-einfache Regel (der überlappende Teil bildet das Mittelstück) entsteht. Offensichtlich ändert das Hinzufügen dieser Regel die erzeugte Sprache nicht. Offensichtlich gibt es nur endlich viele solche neuen Regeln. Offensichtlich kann jede vollständige Ableitung durch Ersetzung von Regelsequenzen durch geeignete neue Regeln in eine vollständige Ableitung mit gleichem Ergebnis aber ohne erweiterten Regeln transformiert werden. Somit können also alle erweiterten Regeln aus γ entfernt werden, ohne dass dies' Auswirkungen auf die erzeugte Sprache hätte. □

4.3 Optimalität der oberen Schranken

In diesem Abschnitt wiederholen bzw. zeigen wir die Beziehungen $cSSL(b) \not\subseteq REG$, $cSSL(n) \not\subseteq CF$ und $SRSL(n) \not\subseteq CF$.

Lemma 4.7. $cSSL(b) \not\subseteq REG$.

Beweis. Dieses Resultat ist allgemein bekannt, in [PRS98] jedoch nicht aufgeführt. Der Vollständigkeit halber geben wir hier einen entsprechenden Beweis an.

Es sei $\gamma = (V, A, D)$ das Stickersystem mit

$$\begin{aligned} V &:= \{a, b\}, \\ A &:= \left\{ \begin{bmatrix} a \\ a \end{bmatrix}, \begin{bmatrix} b \\ b \end{bmatrix}, \begin{bmatrix} aa \\ aa \end{bmatrix}, \begin{bmatrix} bb \\ bb \end{bmatrix}, \begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix} \right\}, \\ D &:= \left\{ \left(\begin{pmatrix} a \\ \varepsilon \end{pmatrix}, \begin{pmatrix} a \\ \varepsilon \end{pmatrix} \right), \left(\begin{pmatrix} b \\ \varepsilon \end{pmatrix}, \begin{pmatrix} b \\ \varepsilon \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ a \end{pmatrix}, \begin{pmatrix} \varepsilon \\ a \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ b \end{pmatrix}, \begin{pmatrix} \varepsilon \\ b \end{pmatrix} \right) \right\}. \end{aligned}$$

Dann gilt $L(\gamma) \in cSSL(b) \setminus REG$.

Offensichtlich ist γ ein einfaches Stickersystem ohne erweiterte Regeln. Es ist zudem noch überhangbeschränkt, da der Aufbau des oberen Stranges unabhängig vom unteren Strang erfolgt und somit jede Ableitung durch Umsortierung der Regelanwendungen in eine äquivalente Ableitung mit der Überhangslängenbeschränkung $d = 1$ überführt werden kann. Somit gilt $L(\gamma) \in cSSL(b)$.

Es sei $S_1 := \left\{ w \in \{a, b\}^* : w = w^R \right\}$. Zu jedem Wort $w \in S_1$ lässt sich eine passende vollständige Ableitung von γ angeben. Man kann außerdem zeigen, dass jede vollständige Ableitung von γ eine Ableitung eines Wortes $w \in S_1$ ist. Somit gilt $L(\gamma) = S_1$. Man kann sich relativ leicht mittels des Pumpinglemmas für reguläre Sprachen (siehe [HU79, Lemma 3.1]) überzeugen, dass gilt $S_1 \notin REG$. □

Lemma 4.8 ([PRS98, Theorem 4.2])**.** $cSSL(n) \not\subseteq CF$.

Beweis. Dieses Resultat gilt gemäß [PRS98, Theorem 4.2]. Der Vollständigkeit halber skizzieren wir hier kurz einen entsprechenden Beweis.

Es sei $\gamma = (V, A, D)$ das Stickersystem mit

$$\begin{aligned} V &:= \{a, b, c, z\}, \\ A &:= \left\{ \begin{bmatrix} z \\ z \end{bmatrix} \right\}, \\ D &:= \left\{ \left(\begin{pmatrix} a \\ \varepsilon \end{pmatrix}, \begin{pmatrix} b \\ \varepsilon \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ a \end{pmatrix}, \begin{pmatrix} c \\ \varepsilon \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} \varepsilon \\ b \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} \varepsilon \\ c \end{pmatrix} \right) \right\}. \end{aligned}$$

Dann gilt gemäß [PRS98, Theorem 4.2] $L(\gamma) \in cSSL(n) \setminus CF$.

Offensichtlich gilt $L(\gamma) \in cSSL(n)$, da γ ein einfaches Stickersystem ohne erweiterte Regeln ist.

Angenommen es gälte $L(\gamma) \in CF$. Da die Familie der kontextfreien Sprachen unter Schnitt mit regulären Sprachen abgeschlossen ist (siehe [HU79, Satz 6.5]) und es sich bei $L_{abc} := \{a^m z b^n c^p : m, n, p \in \mathbb{N}\}$ offensichtlich um eine reguläre Sprache handelt, wäre auch $L := L(\gamma) \cap L_{abc}$ kontextfrei. Man kann sich leicht überzeugen, dass gilt $L = \{a^n z b^n c^n : n \in \mathbb{N}\}$. Man kann sich ebenfalls relativ leicht mittels des Pumpinglemmas für kontextfreie Sprachen (siehe [HU79, Lemma 6.1]) überzeugen, dass gilt $L \notin CF$. Widerspruch. □

Lemma 4.9 (vgl. Lemma 4.8)**.** *$SRSL(n) \not\subseteq CF$.*

Beweis. Es sei $\gamma = (V, A, D)$ das Stickersystem mit

$$\begin{aligned} V &:= \{a, b, c, z\}, \\ A &:= \left\{ \begin{bmatrix} z \\ z \end{bmatrix} \right\}, \\ D &:= \left\{ \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} a \\ \varepsilon \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} b \\ a \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} c \\ b \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} \varepsilon \\ c \end{pmatrix} \right) \right\}. \end{aligned}$$

Dann gilt analog zu Lemma 4.8 $L(\gamma) \in SRSL(n) \setminus CF$. Es sei dabei $L_{abc} := \{z a^m b^n c^p : m, n, p \in \mathbb{N}\}$. □

4.4 Optimalität der unteren Schranken

Wir kommen nun zum ersten Hauptteil der vorliegenden Arbeit. In diesem Abschnitt beweisen wir die Beziehungen $REG \not\subseteq SSL(n)$, $LIN \not\subseteq OSL(n)$, $CF \not\subseteq ASL(n)$ und $cSOSL(b) \not\subseteq SRSL(n)$ und beantworten damit drei offenen Fragen aus [PRS98] bezüglich den Beziehungen zwischen REG und $SSL(n)$, CF und $ASL(n)$ sowie $SRSL(x)$ und $SOSL(x)$ mit $x \in \{cb, b, cn, n\}$.

Lemma 4.10 (vgl. Satz 4.12)**.** *$REG \not\subseteq SSL(b)$.*

Beweis. Sei $L := \{a\}^* \cup \{b\}$, dann gilt offensichtlich $L \in REG$.

Angenommen es gälte $L \in SSL(b)$. Dann gäbe es ein einfaches, überhangbeschränktes Stickersystem $\gamma = (V, A, D)$ mit $L(\gamma) = L$ und der Überhangslängenbeschränkung $h \in \mathbb{N}$.

Das Wort b bzw. das Domino $\begin{bmatrix} b \\ b \end{bmatrix}$ lässt sich nur durch ein Axiom erzeugen. Da alle anderen erzeugbaren vollständigen Dominos nur a's enthalten, bestehen die Regeln und alle anderen Axiome o.B.d.A. nur aus a's.

Damit eine Regel d auf ein Domino x anwendbar ist, müssen bestimmte Struktur- und Komplementaritätsbedingungen erfüllt sein. Da es nur einfache Regeln gibt, existieren keine Strukturbedingungen. Da alle Regeln nur

aus a's bestehen und das einzige Axiom mit anderen Basen keine Überhänge besitzt, existieren auch keine Komplementaritätsbedingungen. Somit ist stets jede Regel anwendbar.

Da es beliebig lange Wörter in $L(\gamma)$ gibt, A und D endlich sind und Dominos nur eine endliche Anzahl an Basen besitzen, gibt es beliebig lange überhangbeschränkte Ableitungen.

Betrachten wir nun eine überhangbeschränkte Ableitung der Länge $\geq (2h+1)^2+1$. Jedes Domino dieser Ableitung besitzt eine durch h beschränkten Überhangslänge. Diese Überhänge lassen sich aufgrund der Beschränktheit und der Tatsache, dass sie nur aus a's bestehen, beschreiben durch Tupel der Form (a,b) mit $a,b \in \{-h,\ldots,+h\}$. Das Tupel $(+5,-3)$ beschreibt zum Beispiel den Überhang $\binom{a^5}{\varepsilon}\binom{?}{?}\binom{\varepsilon}{a^3}$. Somit gibt es nur $(2h+1)^2$ paarweise verschiedene Überhänge. Dann gibt es aber in der hier betrachteten Ableitung zwei verschiedene Dominos x und y mit gleichen Überhängen. Sei o.B.d.A. $x \to_\gamma^* y$, dann gilt $y = \binom{a^r}{a^r} \cdot x \cdot \binom{a^s}{a^s}$ mit $r,s \in \mathbb{N}$ und $r \neq 0$ oder $s \neq 0$.

Die dabei verwendeten Regeln angewendet auf das Axiom $\binom{b}{b}$ bilden eine Ableitung des Dominos $\binom{a^rba^s}{a^rba^s}$. Somit gilt $w = a^rba^s \in L(\gamma)$. Das bildet jedoch einen Widerspruch zu $w \notin L$ und $L(\gamma) = L$. □

Mit etwas mehr Aufwand können wir das Ergebnis verbessern. Dabei wird Teil 3 des nun folgenden Lemmas eine zentrale Rolle spielen.

Sei $d \in \mathbb{N}$, dann sind die Vergleichsoperatoren $\leq, <$ auf $\mathbb{N}^d$ definiert durch

$$\begin{aligned} x \leq y &\iff x[1] \leq y[1] \text{ und } \ldots \text{ und } x[d] \leq y[d], \\ x < y &\iff x \leq y \text{ und } x \neq y. \end{aligned}$$

Lemma 4.11 ([Dic13], [Hig52])**.** *Sei $d \in \mathbb{N}$, dann gilt für $(\mathbb{N}^d, <)$:*

(1) Für $d \geq 2$ und $x \in \mathbb{N}^d$ mit $x \neq (0,\ldots,0)$ gibt es unendlich viele mit x unvergleichbare[1] Elemente aus $\mathbb{N}^d$.

(2) Für $d \geq 2$ gibt es beliebig große endliche Mengen paarweise unvergleichbarer Elemente aus $\mathbb{N}^d$.

(3) Es gibt keine unendliche Menge paarweise unvergleichbarer Elemente aus $\mathbb{N}^d$.

[1] Zwei Elemente x und y sind bezüglich der Vergleichsoperation $<$ unvergleichbar, falls weder $x < y$ noch $y < x$ gilt.

Beweis. Aussage (1) und (2) sind allgemein bekannt. Aussage (3) wurde erstmalig in [Dic13] bewiesen und in [Hig52] verallgemeinert. Wir geben dennoch einen Beweis aller drei Aussagen an, da es sich um das Kernstück von Satz 4.12 handelt.[2]

(1) Sei $d \geq 2$ und $x \in \mathbb{N}^d$ mit $x \neq (0, \ldots, 0)$, dann gibt es ein $i \in \{1, \ldots, d\}$ mit $x[i] > 0$ und ein $k \in \{1, \ldots, d\}$ mit $i \neq k$.

Offensichtlich ist dann $M_x := \{y \in \mathbb{N}^d : y[i] < x[i], y[k] > x[k]\}$ eine unendliche Menge von mit x unvergleichbaren Elementen aus $\mathbb{N}^d$.

(2) Sei $n \in \mathbb{N}$, $d \geq 2$ und $M_n := \{(a, b, 0, ..., 0) \in \mathbb{N}^d : a + b = n\}$, dann ist M_n eine Menge paarweise unvergleichbarer Elemente aus $\mathbb{N}^d$ mit $|M_n| = n + 1$.

Angenommen es gäbe zwei Elemente $x, y \in M_n$ mit $x < y$. Dann gälten $x[i] = y[i] = 0$ für $3 \leq i \leq d$, $x[1] \leq y[1]$, $x[2] \leq y[2]$ und $x[1] \neq y[1]$ oder $x[2] \neq y[2]$. Somit gälte aber $x[1] + x[2] < y[1] + y[2]$. Das bildet jedoch einen Widerspruch zur Definition von M_n.

Es gibt genau $n + 1$ verschiedene Elemente in M_n, da $x[1]$ beliebig aus der Menge $\{0, \ldots, n\}$ gewählt werden kann, $x[2]$ sich entsprechend der Formel $x[2] := n - x[1]$ errechnet und $x[i] = 0$ für alle $3 \leq i \leq d$ gilt.

(3) Wir zeigen induktiv über die Dimension d, dass jede Menge paarweise unvergleichbarer Elemente aus $\mathbb{N}^d$ endlich ist.

Induktionsanfang

> Sei $d = 0$, dann gilt $\mathbb{N}^d = \{\varepsilon\}$. Somit ist jede Teilmenge von $\mathbb{N}^d$ höchstens einelementig und somit endlich.
> (Sei $d = 1$, dann gilt $\mathbb{N}^d = \mathbb{N}$. Da zwei verschiedene natürliche Zahlen stets vergleichbar sind, ist jede Menge paarweise unvergleichbarer Elemente höchstens einelementig und somit endlich.)

Induktionsvoraussetzung

> Es sei $d > 0$. Die Behauptung gelte für die Dimensionen $\leq d - 1$.

Induktionsbehauptung

> Die Behauptung gilt auch für die Dimensionen d.

Induktionsschritt

[2] Dieses Lemma wurde von mir im Rahmen der Diplomarbeit [Wei04] formuliert und der zugehörige Beweis selbständig entwickelt. Erst im Nachgang stellte sich heraus, dass ich nicht der Erste war. Der hier angegebene Beweis ist meine Variante, Abweichungen zum Original sind demnach möglich und sogar sehr wahrscheinlich.

Die leere Menge ist endlich. Sei M eine nichtleere Menge paarweise unvergleichbarer Elemente aus $\mathbb{N}^d$. Da M nicht leer ist, gibt es ein $x \in M$.

Sei $T(i,k) := \{y \in M : y[i] = k\}$ mit $i \in \{1,...,d\}$ und $k \in \{0,...,x[i]\}$ und sei $T = \bigcup_{i,k} T(i,k)$. Offensichtlich gilt $T \subseteq M$. Sei $y \in M$, dann gibt es ein i mit $y[i] \leq x[i]$, da ansonsten $x < y$ gelten würde, was wegen der paarweisen Unvergleichbarkeit der Elemente aus M nicht sein kann. Somit gilt aber $y \in T(i, y[i]) \subseteq T$. Daraus folgt nun $M = T$.

Es gilt $y[i] = k$ für alle Elemente y der Menge $T(i,k)$. Somit befinden sich alle diese Elemente in einem Raum der Dimension $\mathbb{N}^{d-1}$ (die Komponente i wegprojizieren). Außerdem sind die Elemente wegen $T(i,k) \subseteq T = M$ paarweise unvergleichbar. Nach Induktionsvoraussetzung ist $T(i,k)$ endlich.

Die Menge T ist eine endliche Vereinigung endlicher Mengen, da es nur endlich viele Mengen $T(i,k)$ gibt und jede dieser Mengen endlich ist. Dann ist aber T und somit auch M endlich.

□

Satz 4.12. *REG* $\not\subseteq$ *SSL*(n).

Beweis. Wir betrachten dazu wieder die Sprache $L := \{a\}^* \cup \{b\}$ aus dem Beweis von Lemma 4.10.

Angenommen es gälte $L \in SSL(n)$. Dann gäbe es ein einfaches Stickersystem $\gamma = (V, A, D)$ mit $L(\gamma) = L$.

Analog zu Lemma 4.10 können wir o.B.d.A. annehmen, dass die Regeln nur aus a's bestehen und $\left[\begin{smallmatrix} b \\ b \end{smallmatrix}\right]$ das einzige nicht nur aus a's bestehende Axiom ist. Dadurch sind auch hier stets alle Regeln anwendbar und das erzeugte Domino ist unabhängig von der Reihenfolge der Regelanwendungen.

Sei $A = \{a_1, ..., a_n\}$, $D = \{d_1, ..., d_m\}$ und $\gamma_i := (V, \{a_i\}, D)$ mit $1 \leq i \leq n$, dann gilt $L(\gamma) = \bigcup_i L(\gamma_i)$. Da es sich hierbei um eine endliche Vereinigung handelt und $|L(\gamma)| = \infty$ gilt, gibt es ein γ_k mit $|L(\gamma_k)| = \infty$.

Sei P die Menge aller vollständigen Ableitungen von γ_k mit zuerst nur Anwendungen der Regel d_1, dann Regel d_2, dann .... Eine solche Ableitung lässt sich eineindeutig beschreiben durch ein Tupel $(c_1, \ldots, c_m) \in \mathbb{N}^{|D|}$. Dabei gibt c_i die Anzahl der Anwendungen der Regel d_i an. Somit gelte o.B.d.A. $P \subseteq \mathbb{N}^{|D|}$. Mit $mol(x)$ bezeichnen wir das durch die Ableitung x aus dem Axiom a_k erzeugte vollständige Domino. Sind x und y zwei Ableitungen aus P mit $x < y$, so lässt sich $mol(y)$ aus $mol(x)$ ableiten.

Für jedes vollständige Domino $x \in L(\gamma_k)$ gibt es eine entsprechende Ableitung in P. Wegen $|L(\gamma_k)| = \infty$ gilt $|P| = \infty$. Gemäß Lemma 4.11 gibt es keine unendliche Menge paarweise unvergleichbarer Elemente aus $\mathbb{N}^{|D|}$.[3] Somit gibt es zwei vergleichbare Elemente $x, y \in P$ mit $x < y$. Es gilt $mol(y) = \binom{a^r}{a^r} \cdot mol(x) \cdot \binom{a^s}{a^s}$ mit $r, s \in \mathbb{N}$, $r \neq 0$ oder $s \neq 0$ und $mol(x) \rightarrow_{\gamma_k}^* mol(y)$.

Die dabei verwendeten Regeln angewendet auf das Axiom $\left[\begin{smallmatrix} b \\ b \end{smallmatrix}\right]$ bilden in γ eine Ableitung des Dominos $\left[\begin{smallmatrix} a^r b a^s \\ a^r b a^s \end{smallmatrix}\right]$. Somit gilt $w = a^r b a^s \in L(\gamma)$. Das bildet jedoch einen Widerspruch zu $w \notin L$ und $L(\gamma) = L$. □

Anmerkung 4.13. *In [PRS98, Theorem 4.7] wurde ein Beweis für $L_{bab} \in REG \setminus cSOSL(n)$ mit $L_{bab} := \{ba^n b : n \in \mathbb{N}\}$ angegeben. Analog zu Satz 4.12 lässt sich dieses Ergebnis zu $L_{bab} \notin SSL(n)$ verbessern. Im Gegensatz zu Satz 4.12 gibt es nun auch b's enthaltende Regeln. Die (maximal vier) Anwendungen dieser Regeln pro vollständiger Ableitung lassen sich aber problemlos an das Ableitungsende verlagern und wegen $|D| < \infty$ analog zur Einschränkung auf ein Axiom aus der Betrachtung ausblenden.*

Folgerung 4.14. *Die Sprachfamilien $SSL(x)$, $SOSL(x)$ und $SRSL(x)$ mit $x \in \{cb, b, cn, n\}$ sind nicht abgeschlossen unter der Vereinigung, nicht abgeschlossen unter der Komplementbildung, nicht abgeschlossen unter dem Schnitt mit regulären Chomskysprachen und nicht abgeschlossen unter Konkatenation (mit einem einzelnen Buchstaben).*

Beweis. Es seien $L_0 := \{a\}^* \cup \{b\}$, $L_1 := \{ba^n b : n \in \mathbb{N}\}$, $L_2 := \{a, b\}^*$, $L_3 := \{a\}^*$, $L_4 := \{b\}$, $L_5 := \{u \cdot b \cdot v : u, v \in \{a, b\}^*, |u \cdot v| \geq 1\}$, $L_6 := \{ba^n : n \in \mathbb{N}\}$, $L_7 := L_3 \cup L_4$, $L_8 := L_5^{co}$, $L_9 := L_2 \cap L_7$ und $L_{10} := L_6 \cdot L_4$. Die genannten Sprachfamilien enthalten offensichtlich die Sprachen L_2, L_3, L_4, L_5 und L_6. (Der Beweis von $L_5 \in cSRSL(b)$ erfolgt durch Konstruktion eines entsprechenden klassischen, einfachen, regulären, überhangbeschränkten Stickersystems. Diese Konstruktion ist ähnlich der Konstruktion im Beweis von Folgerung 4.15.) Wegen Satz 4.12 enthalten die genannten Sprachfamilien nicht $L_0 = L_7 = L_8 = L_9$ und wegen Anmerkung 4.13 auch nicht $L_1 = L_{10}$. □

Folgerung 4.15. *Die Sprachfamilien $SSL(x)$ und $SOSL(x)$ mit $x \in \{cb, b, cn, n\}$ sind nicht abgeschlossen unter dem Schnitt.*

Beweis. Es seien $L_{baba} := \{ba^n ba^{2m} : n, m \in \mathbb{N}\}$ und $L_{bab} := \{ba^n b : n \in \mathbb{N}\}$. Es sei $\gamma = (V, A, D)$ das klassiche, einfache, reguläre, überhangbeschränkte

[3] Analog zu $(\mathbb{N}^{|D|}, <)$ ist auch für $(D^*, <)$ bekannt, dass es keine unendlichen Mengen paarweise unvergleichbarer Elemente gibt. (Es sei $x \leq y$, falls x aus y durch Löschen von Buchstaben an beliebigen Stellen hervorgeht.) Somit hätten wir o.B.d.A. auch $P \subseteq D^*$ annehmen können.

Stickersystem mit

$$\begin{aligned} V &:= \{a, b\}, \\ A &:= \left\{ \begin{bmatrix} bb \\ bb \end{bmatrix}, \begin{bmatrix} b \\ b \end{bmatrix} \begin{pmatrix} a \\ \varepsilon \end{pmatrix}, \begin{bmatrix} ba \\ ba \end{bmatrix} \begin{pmatrix} a \\ \varepsilon \end{pmatrix} \right\}, \\ D &:= \left\{ \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} \varepsilon \\ aa \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} aa \\ \varepsilon \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} \varepsilon \\ ab \end{pmatrix} \right), \left(\begin{pmatrix} \varepsilon \\ \varepsilon \end{pmatrix}, \begin{pmatrix} b \\ \varepsilon \end{pmatrix} \right) \right\}. \end{aligned}$$

Dann ist $L_{baba} = L(\gamma)$. Somit enthalten die genannten Sprachfamilien die Sprache L_{baba}. Die genannten Sprachfamilien sind offensichtlich abgeschlossen unter der Umkehrung (einfach alle Axiome und Regeln spiegeln). Somit enthalten sie auch L_{baba}^R. Aufgrund von Anmerkung 4.13 enthalten sie jedoch nicht $L_{bab} = L_{baba} \cap L_{baba}^R$. □

Lemma 4.16. *$cSOSL(b) \not\subseteq SRSL(n)$.*

Beweis. Offensichtlich gilt $L_{ab} := \{a^n b : n \in \mathbb{N}\} \in cSOSL(b)$.

Angenommen es gälte $L_{ab} \in SRSL(n)$. Dann gäbe es ein einfaches reguläres Stickersystem γ mit $L(\gamma) = L_{ab}$. Analog zu $L_{bab} \notin SSL(n)$ aus Anmerkung 4.13 lässt sich zeigen, dass das Domino $\left[\begin{smallmatrix} ba^s \\ ba^s \end{smallmatrix}\right]$ bzw. das Wort $w = ba^s$ für ein gewisses $s \geq 1$ ableitbar wäre. Widerspruch. □

Folgerung 4.17. *Die Sprachfamilien $SRSL(n)$, $cSRSL(n)$, $SRSL(b)$ und $cSRSL(b)$ sind nicht abgeschlossen unter der Umkehrung.*

Beweis. Offensichtlich gilt $L_{ba} := \{ba^n : n \in \mathbb{N}\} \in cSRSL(b)$. Gemäß Lemma 4.16 gilt jedoch $L_{ba}^R = L_{ab} \notin SRSL(n)$. □

Korollar 4.18. *$LIN \not\subseteq OSL(n)$.*

Beweis. Gemäß Lemma 3.7, Beobachtung 2.7 und Satz 5.4 gelten $LIN \not\subseteq OHL_* \supseteq OHL_2 \supseteq OSL(n)$. □

Korollar 4.19. *$CF \not\subseteq ASL(n)$.*

Beweis. Gemäß Lemma 3.8, Beobachtung 2.7 und Satz 5.1 gelten $CF \not\subseteq OHL_{*,*} \supseteq OHL_{2,2} \supseteq ASL(n)$. □

Folgerung 4.20. *Die Stickersprachfamilien $cSSL(b)$, $SSL(b)$, $cSSL(n)$, $SSL(n)$, $cASL(b)$, $ASL(b)$, $cASL(n)$ und $ASL(n)$ sind nicht abgeschlossen unter der Konkatenation.*

Beweis. Aufgrund von Lemma 4.7 und Korollar 4.19 enthalten die genannten Stickersprachfamilien $S_1 := \{w \in \{a, b\}^* : w = w^R\}$ aber nicht $S_2 := S_1 \cdot S_1 = \{v \cdot w \in \{a, b\}^* : v = v^R, w = w^R\}$. □

	REG	LIN	CF	CS
$ASL(n)$	$\supset$	$\supset$	$\not\subseteq, \not\supseteq$	$\subset$
$cASL(n)$	$\supset$	$\supset$	$\not\subseteq, \not\supseteq$	$\subset$
$ASL(b)$	$\supset$	$=$	$\subset$	$\subset$
$cASL(b)$	$\supset$	$=$	$\subset$	$\subset$
$SSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$cSSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$SSL(b)$	$\not\subseteq, \not\supseteq$	$\subset$	$\subset$	$\subset$
$cSSL(b)$	$\not\subseteq, \not\supseteq$	$\subset$	$\subset$	$\subset$
$OSL(n)$	$\supset$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$cOSL(n)$	$=$	$\subset$	$\subset$	$\subset$
$OSL(b)$	$=$	$\subset$	$\subset$	$\subset$
$cOSL(b)$	$=$	$\subset$	$\subset$	$\subset$
$RSL(n)$	$\supset$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$cRSL(n)$	$=$	$\subset$	$\subset$	$\subset$
$RSL(b)$	$=$	$\subset$	$\subset$	$\subset$
$cRSL(b)$	$=$	$\subset$	$\subset$	$\subset$
$SOSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$cSOSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cSOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$cSRSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cSRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$

Tabelle 4.1: Beziehungen zwischen Stickersprachfamilien und Chomskysprachfamilien

	$cASL(b)$	$ASL(b)$	$cASL(n)$	$ASL(n)$
$cASL(n)$				$\subseteq$
$ASL(b)$			$\subset$	$\subset$
$cASL(b)$		$=$	$\subset$	$\subset$
$SSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\subset$
$cSSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$	$\subset$
$SSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cSSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$OSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\subset$
$cOSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$OSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$RSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\subset$
$cRSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$RSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$SOSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\subset$
$cSOSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cSOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\subset$
$cSRSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cSRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$

Tabelle 4.2: Beziehungen der Stickersprachfamilien untereinander

	$cSSL(b)$	$SSL(b)$	$cSSL(n)$	$SSL(n)$
$cSSL(n)$				$\subseteq$
$SSL(b)$			$\not\supseteq$	$\subset$
$cSSL(b)$		$\subseteq$	$\subset$	$\subset$
$OSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$
$cOSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$
$OSL(b)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$
$cOSL(b)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$
$RSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$
$cRSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$
$RSL(b)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$
$cRSL(b)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$
$SOSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\subset$
$cSOSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SOSL(b)$	$\not\supseteq$	$\subset$	$\not\supseteq$	$\subset$
$cSOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\subset$
$cSRSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(b)$	$\not\supseteq$	$\subset$	$\not\supseteq$	$\subset$
$cSRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$

Tabelle 4.3: Beziehungen der Stickersprachfamilien untereinander

	$cOSL(b)$	$OSL(b)$	$cOSL(n)$	$OSL(n)$
$cOSL(n)$				$\subset$
$OSL(b)$			$=$	$\subset$
$cOSL(b)$		$=$	$=$	$\subset$
$RSL(n)$	$\supset$	$\supset$	$\supset$	$\subseteq$
$cRSL(n)$	$=$	$=$	$=$	$\subset$
$RSL(b)$	$=$	$=$	$=$	$\subset$
$cRSL(b)$	$=$	$=$	$=$	$\subset$
$SOSL(n)$	$\not\subset, \not\supset$	$\not\subset, \not\supset$	$\not\subset, \not\supset$	$\subset$
$cSOSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cSOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(n)$	$\not\subset, \not\supset$	$\not\subset, \not\supset$	$\not\subset, \not\supset$	$\subset$
$cSRSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cSRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
	$cRSL(b)$	$RSL(b)$	$cRSL(n)$	$RSL(n)$
$cRSL(n)$				$\subset$
$RSL(b)$			$=$	$\subset$
$cRSL(b)$		$=$	$=$	$\subset$
$SOSL(n)$	$\not\subset, \not\supset$	$\not\subset, \not\supset$	$\not\subset, \not\supset$	$\subset$
$cSOSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cSOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(n)$	$\not\subset, \not\supset$	$\not\subset, \not\supset$	$\not\subset, \not\supset$	$\subset$
$cSRSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
$cSRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$

Tabelle 4.4: Beziehungen der Stickersprachfamilien untereinander

	$cSOSL(b)$	$SOSL(b)$	$cSOSL(n)$	$SOSL(n)$
$cSOSL(n)$				$\subset$
$SOSL(b)$			$\supseteq$	$\subset$
$cSOSL(b)$		$\subseteq$	$=$	$\subset$
$SRSL(n)$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\not\subseteq, \not\supseteq$	$\subset$
$cSRSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(b)$	$\not\supseteq$	$\subset$	$\not\supseteq$	$\subset$
$cSRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$
	$cSRSL(b)$	$SRSL(b)$	$cSRSL(n)$	$SRSL(n)$
$cSRSL(n)$				$\subset$
$SRSL(b)$			$\supseteq$	$\subset$
$cSRSL(b)$		$\subseteq$	$=$	$\subset$

Tabelle 4.5: Beziehungen der Stickersprachfamilien untereinander

	$A \cup B$	$A \cap B$	A^{co}	A^R	$A \cdot B$	$A \cap R$
$ASL(n)$	?	?	?	+	−	?
$cASL(n)$	?	?	?	+	−	?
$ASL(b)$	+	−	−	+	−	+
$cASL(b)$	+	−	−	+	−	+
$SSL(n)$	−	−	−	+	−	−
$cSSL(n)$	−	−	−	+	−	−
$SSL(b)$	−	−	−	+	−	−
$cSSL(b)$	−	−	−	+	−	−
$OSL(n)$	?	?	?	+	?	?
$cOSL(n)$	+	+	+	+	+	+
$OSL(b)$	+	+	+	+	+	+
$cOSL(b)$	+	+	+	+	+	+
$RSL(n)$	?	?	?	?	?	?
$cRSL(n)$	+	+	+	+	+	+
$RSL(b)$	+	+	+	+	+	+
$cRSL(b)$	+	+	+	+	+	+
$SOSL(n)$	−	−	−	+	−	−
$cSOSL(n)$	−	−	−	+	−	−
$SOSL(b)$	−	−	−	+	−	−
$cSOSL(b)$	−	−	−	+	−	−
$SRSL(n)$	−	?	−	−	−	−
$cSRSL(n)$	−	?	−	−	−	−
$SRSL(b)$	−	?	−	−	−	−
$cSRSL(b)$	−	?	−	−	−	−

Tabelle 4.6: Abschlusseigenschaften der Stickersprachfamilien

Kapitel 5

Stickersysteme und Mehrkopfautomaten

Während wir im vorangehenden Kapitel die Beziehungen der Stickersprachfamilien untereinander und mit den Chomskysprachfamilien analysiert haben, wollen wir nun einen komplett anderen Weg gehen und die Untersuchung des Zusammenhangs zwischen Stickersystemen und Mehrkopfautomaten in den Vordergrund stellen. Dabei werden wir unter anderem untersuchen, ob und an welcher Stelle sich die einzelnen Stickersprachfamilien in die Hierarchie der einfachen Mehrkopfsprachfamilien ($SHL_1 \subset SHL_2 \subset \ldots$) einordnen lassen.

Tabelle 5.1 auf Seite 55 gibt einen Überblick über die zwischen den Stickersprachfamilien und den einfachen Mehrkopfsprachfamilien geltenden Beziehungen auf der Grundlage von Abschnitt 2.1, 2.3 und 5.x. Vermutlich gelten zusätzlich die Beziehungen $cASL(n) \not\subseteq SHL_4$, $cSSL(n) \not\subseteq AHL_3$, $RSL(n) \not\subseteq SHL_2$ und $SOSL(n) \not\subseteq SHL_2$. Dadurch könnte in der Übersicht jedes Vorkommen von $\not\supseteq$ durch $\not\subseteq, \not\supseteq$ ersetzt werden. Ein Beweis dieser Vermutung steht jedoch noch aus.

5.1 Obere Schranken

Wir kommen nun zum zweiten Hauptteil der vorliegenden Arbeit. In diesem Abschnitt zeigen bzw. folgern wir die Inklusionen $ASL(n) \subseteq OHL_{2.2}$, $SSL(n) \subseteq SOHL_{2.2}$, $OSL(n) \subseteq OHL_2$, $SRSL(n) \subseteq SOHL_2$, $ASL(b) \subseteq SOHL_{1.1}$ und $OSL(b) \subseteq SOHL_1$ und liefern damit die Grundlage zur Einordnung der Stickersprachfamilien in die Hierarchien der Mehrkopfsprachfamilien, wobei wir uns in Tabelle 5.1 exemplarisch auf die Hierarchie der einfachen Mehrkopfsprachfamilien beschränken.

Gemäß Anmerkung 4.3 gilt $ASL(n) \subseteq CS$. Diese Aussage kann zu $ASL(n) \subseteq OHL_{2.2}$ verbessert werden, da Stickersysteme starken Einschränkungen unterliegen:

(1) Bereits erzeugte Basen können nicht wieder überschrieben oder gelöscht werden.

(2) Bereits erzeugte Basen können höchstens einmal zu einem späteren Zeitpunkt (durch Schreiben des zweiten Stranges) „gelesen“ werden.

(3) Die Erzeugungs- und Leserichtung ist streng vorgegebene und stark einschränkend (von innen nach außen).

Satz 5.1. $ASL(n) \subseteq OHL_{2.2}$.

Beweis. In [PRS98, Chapter 5] wurde das Konzept der Watson-Crick-Automaten vorgestellt. Durch geeignete Erweiterung dieses Konzeptes können wir Stickersysteme als spezielle erweiterte Watson-Crick-Automaten betrachten. Gemäß [PRS98, Lemma 5.8] sind Watson-Crick-Automaten äquivalent zu (0.2)-Einwegmehrkopfautomaten. Durch Verallgemeinerung von [PRS98, Lemma 5.8] auf erweiterte Watson-Crick-Automaten und Einschränkung auf Stickersysteme erhalten wir den folgenden Beweis.

Sei $\gamma = (V, A, D)$ ein Stickersystem, dann arbeitet der (2.2)-Einwegmehrkopfautomat $\mathcal{A} = (V, Z, s, F, T)$ wie folgt:

Initialisierung

> Alle vier Köpfe befinden sich an der absoluten Position 1. Es wird in die Akzeptierungshase gewechselt, falls alle vier Köpfe das Symbol ε lesen und $\varepsilon \in L(\gamma)$ gilt. Ansonsten wird eine beliebige Position p geraten, die beiden Köpfe *top-left* (1) und *bottom-left* (2) an der Position p positioniert, die beiden Köpfe *top-right* (3) und *bottom-right* (4) an der Position $p + 1$ positioniert und in die Phase der Axiomauswahl gewechselt.

Axiomauswahl

> Es wird ein beliebiges Axiom $x_1x_2x_3 \in A$ ausgewählt und in der Form $(x_1x_2\binom{\varepsilon}{\varepsilon}, x_3)$ gemerkt. Anschließend wird in die Phase des Regel- und Axiomtests gewechselt.

Regelauswahl

Es wird eine beliebige Regel $(x, y) \in D$ ausgewählt und gemerkt. Anschließend wird in die Phase des Regel- und Axiomtests gewechselt.

Regel- und Axiomtest

Die gemerkte Regel oder das gemerkte Axiom sei (x, y). Es wird zuerst das Domino $y \in W(V)$ betrachtet:

(1) Gilt $y = \binom{u}{v} \in S(V)$, so bewegt sich Kopf *top-right* um $|u|$ Schritte nach rechts und stellt sicher, dass dabei das Wort u gelesen wird. Analog bewegt sich Kopf *bottom-right* um $|v|$ Schritte nach rechts und stellt sicher, dass dabei das Wort v gelesen wird. Die relative Position der beiden Köpfe zueinander ist irrelevant.

(2) Gilt $y = y_1y_2y_3 \in LR(V)$, dann wird zuerst y_1, dann y_2 und danach y_3 betrachtet. Für y_1 und y_3 wird analog zu (1) verfahren. Sei $y_2 = \left[\begin{smallmatrix} u \\ u \end{smallmatrix}\right]$, so bewegen sich Kopf *top-right* und *bottom-right* um $|u|$ Schritte nach rechts und stellen sicher, dass dabei das Wort u gelesen wird. Außerdem wird sichergestellt, dass beide Köpfe während des Tests von y_2 direkt übereinander stehen.

Symmetrisch zu y wird nun das Domino x betrachtet, Kopf *top-left* und *bottom-left* werden also auf der Grundlage von x nach links bewegt. Anschließend wird in die Akzeptierungs- oder Regelauswahlphase gewechselt.

Akzeptierung

Die Eingabe wird akzeptiert, falls alle vier Köpfe das Symbol ε lesen.

Es sei $C(\gamma) := \{(x, y, z) : x \in S(V), y \in LR(V), z \in S(V), x \cdot y \cdot z \in WK(V)\}$ die Menge der *Konfigurationen* von γ. Es sei $A := C(\gamma) \cap (S(V) \times C^*(\gamma) \times S(V))$ und $B := C^*(\mathcal{A}) \cap \left(V^* \times \{q\} \times \mathbb{N}^4\right)$ mit q als Austrittspunkt aus der Phase des Regel- und Axiomtests. Dann sei $\lambda : C(\gamma) \to C(\mathcal{A})$ eine Funktion, die aus einer Konfiguration des Stickersystems γ eine Konfiguration des Automaten $\mathcal{A}$ berechnet mit

$$\lambda\left((x, y, z)\right) := \left(x^t \cdot y^t \cdot z^t, q, \left(|x^t|, |x^b|, |x^t \cdot y^t| + 1, |x^b \cdot y^b| + 1\right)\right).$$

Man beweist nun mittels Induktion über die Anzahl der Regelanwendungen die Beziehung $\lambda(A) = B$ und zeigt damit, dass jede Ableitung des Stickersystems γ durch den Automaten $\mathcal{A}$ simuliert werden kann und andererseits jeder Lauf des Automaten $\mathcal{A}$ die Simulation einer Ableitung des Stickersystems γ darstellt. Unter Einbeziehung der Akzeptierungsbedingung und der Behandlung des Falles $\varepsilon \in L(\gamma)$ folgt daraus $L(\mathcal{A}) = L(\gamma)$. □

Anmerkung 5.2. *Vermutlich können* (2.2)*-Einwegmehrkopfautomaten etwas mehr als Stickersysteme, da die Regelanwendbarkeit durch die Zustände besser gesteuert werden kann. Dennoch ist zwischen Stickersystemen und* (2.2)*-Einwegmehrkopfautomaten eine starke Äquivalenz erkennbar.*

Gerade haben wir gezeigt, dass alle Stickersprachen wegen Satz 5.1 und Satz 2.11 durch einfache 5-Mehrkopfautomaten akzeptiert werden können. Wir wollen nun untersuchen, ob und wie stark die Anzahl der Köpfe durch Einschränkung der Stickersysteme reduziert werden kann.

Folgerung 5.3. $SSL(n) \subseteq SOHL_{2.2}$.

Beweis. Satz 5.1 liefert einen Beweis der Inklusion $ASL(n) \subseteq OHL_{2.2}$. Durch Einschränkung auf einfache Stickersprachen entfällt die Notwendigkeit zur relativen Kopfpositionserkennung, so dass es sich dann bei dem dort angegebenen (2.2)-Einwegmehrkopfautomaten um einen einfachen (2.2)-Einwegmehrkopfautomaten handelt. Somit gilt $SSL(n) \subseteq SOHL_{2.2}$. □

Satz 5.4. $OSL(n) \subseteq OHL_2$.

Beweis. Sei $\gamma = (V, A, D)$ ein einseitiges Stickersystem, dann arbeitet der (0.2)-Einwegmehrkopfautomat $\mathcal{A} = (V, Z, s, F, T)$ wie folgt:

Wegen $OSL(n) \subseteq ASL(n)$ gibt es gemäß Satz 5.1 für jedes einseitige Stickersystem einen äquivalenten (2.2)-Einwegmehrkopfautomaten. Da die Regeln einseitig sind und somit der Aufbau der linken und rechten Seite unabhängig voneinander erfolgt, ist es keine Einschränkung diese (2.2)-Einwegmehrkopfautomaten so abzuändern, dass nach der Wahl des Axioms zuerst nur linksseitigen und danach nur rechtseitigen Regeln zum Einsatz kommen. In der Phase der linksseitigen Regelanwendungen bewegen sich die Köpfe *top-left* und *bottom-left* vom Axiom zum linken Rand. Die Köpfe *top-right* und *bottom-right* bewegen sich dagegen nicht, werden also in dieser Phase nicht benötigt. Analog bewegen sich in der Phase der rechtsseitigen Regelanwendungen die Köpfe *top-right* und *bottom-right* vom Axiom zum rechten Rand. Die Köpfe *top-left* und *bottom-left* bewegen sich dagegen nicht, werden also in dieser Phase nicht benötigt. Es ist offensichtlich möglich, den Simulationsprozess der Phase der linksseitigen Regelanwendungen inklusive der Axiomphase so abzuändern, dass sich die Köpfe nicht von innen nach links außen, sondern von links außen nach innen bewegen, wobei die Regeln dann natürlich in der umgekehrten Reihenfolge ausgewählt werden müssen. (Siehe auch Lemma 3.5) Erst im Anschluss an die Phase der linksseitigen Regelanwendungen wird das Axiom ausgewählt und überprüft. Danach beginnt die Phase der rechtseitigen Regelanwendungen. Offensichtlich können nun die Köpfe *top-left* und

bottom-left die Aufgaben von *top-right* und *bottom-right* übernehmen, so dass eine Reduktion auf 2 Köpfe möglich ist.

Initialisierung

> Beide Köpfe befinden sich an der absoluten Position 1. Es wird in die Akzeptierungsphase gewechselt, falls beide Köpfe das Symbol ε lesen und $\varepsilon \in L(\gamma)$ gilt. Ansonsten wird in die Phase der linksseitigen Regelauswahl oder die Phase der Axiomauswahl gewechselt.

Linksseitige Regelauswahl

> Es wird eine beliebige linksseitige Regel $(x, \varepsilon) \in D$ ausgewählt und gemerkt. Anschließend wird in die Phase des Regel- und Axiomtests gewechselt.

Rechtsseitige Regelauswahl

> Es wird eine beliebige rechtsseitige Regel $(\varepsilon, x) \in D$ ausgewählt und gemerkt. Anschließend wird in die Phase des Regel- und Axiomtests gewechselt.

Axiomauswahl

> Es wird ein beliebiges Axiom $x \in A$ ausgewählt und in der Form (ε, x) gemerkt. Anschließend wird in die Phase des Regel- und Axiomtests gewechselt.

Regel- und Axiomtest

> Die gemerkte Regel oder das gemerkte Axiom sei (x, ε) oder (ε, x) mit $x \in W(V)$:
>
> (1) Gilt $x = \binom{u}{v} \in S(V)$, so bewegt sich Kopf *top* um $|u|$ Schritte nach rechts und stellt sicher, dass dabei das Wort u gelesen wird. Analog bewegt sich Kopf *bottom* um $|v|$ Schritte nach rechts und stellt sicher, dass dabei das Wort v gelesen wird. Die relative Position der beiden Köpfe zueinander ist irrelevant.
>
> (2) Gilt $x = x_1x_2x_3 \in LR(V)$, dann wird zuerst x_1, dann x_2 und danach x_3 betrachtet. Für x_1 und x_3 wird analog zu (1) verfahren. Sei $x_2 = \left[\begin{smallmatrix} u \\ u \end{smallmatrix}\right]$, so bewegen sich Kopf *top* und *bottom* um $|u|$ Schritte nach rechts und stellen sicher, dass dabei das Wort u gelesen wird. Außerdem wird sichergestellt, dass beide Köpfe während des Tests von x_2 direkt übereinander stehen.

Anschließend wird in die linksseitige Regelauswahl-, Axiomauswahl-, rechtseitige Regelauswahl- oder Akzeptierungsphase gewechselt. Ein Wechsel in die Phase der linksseitigen Regelauswahl oder die Phase der Axiomauswahl ist allerdings nur zulässig, wenn die gemerkte Regel eine linksseitige Regel war. Analog darf nur in die Phase der rechtseitigen Regelauswahl oder die Akzeptierungsphase gewechselt werden, wenn die gemerkte Regel eine rechtsseitige Regel oder ein Axiom war.

Akzeptierung

Die Eingabe wird akzeptiert, falls beide Köpfe das Symbol ε lesen.

Ähnlich zu Satz 5.1 kann nun die Beziehung $L(\gamma) = L(\mathcal{A})$ und somit $OSL(n) \subseteq OHL_2$ gezeigt werden. □

Folgerung 5.5. $SRSL(n) \subseteq SOHL_2$.

Beweis. Satz 5.4 liefert einen Beweis der Inklusion $OSL(n) \subseteq OHL_2$. Durch die Einschränkung auf einfache, einseitige Stickersprachen wird die relative Kopfpositionserkennung ausschließlich für den Axiomtest benötigt. Durch Einschränkung auf einfache, rechtseitige Stickersysteme entfällt die Notwendigkeit zur relativen Kopfpositionserkennung komplett, da Axiomauswahl und -test direkt nach der Initialisierung durchgeführt werden. Somit gilt $SRSL(n) \subseteq SOHL_2$. □

Lemma 5.6. $ASL(b) \subseteq SOHL_{1.1}$.

Beweis. Es sei $\gamma = (V, A, D)$ ein überhangbeschränktes Stickersystem mit der Überhangslängenbeschränkung $d \in \mathbb{N}$. Wegen $L(\gamma) \in ASL(b) \subseteq ASL(n)$ gibt es gemäß Satz 5.1 einen (2.2)-Einwegmehrkopfautomaten $\mathcal{A}$ mit $L(\mathcal{A}) = L(\gamma)$.

Wir konstruieren nun aus dem (2.2)-Einwegmehrkopfautomaten $\mathcal{A} = (X, Z, s, F, T)$ einen einfachen 2-Mehrkopfautomaten $\mathcal{B} = (X, Z', s', F', T')$ wie folgt:

Wir wandeln analog zu Satz 2.11 alle vier Köpfe in virtuelle Köpfe um und erzeugen zwei neue reale Köpfe *left* und *right*. Es gibt auch hier vier Zähler, wobei Zähler $1, 2$ die relativen Kopfpositionen der virtuellen Köpfe *top-left*, *bottom-left* zu dem realen Kopf *left* und Zähler $3, 4$ die relativen Kopfpositionen der virtuellen Köpfe *top-right*, *bottom-right* zu dem realen Kopf *right* angeben. Wir beschränken die Zähler durch die Konstante d $(-d \ldots + d)$ und realisieren diese im Gegensatz zu Satz 2.11 komplett durch die Zustände. Da γ durch d überhangbeschränkt ist, kann o.B.d.A. angenommen werden, dass die Köpfe *top-left* und *bottom-left* sowie *top-right* und *bottom-right* höchstens

den Abstand d zueinander haben. Somit ist es möglich die beiden realen Köpfe so zu steuern, dass niemals ein Über- oder Unterlauf der Zähler stattfindet und es trotzdem zu keiner Beeinträchtigung der Simulation gemäß Satz 2.11 kommt.

Offensichtlich ist $\mathcal{B}$ ein einfacher 2-Einwegmehrkopfautomat. Aufgrund von Satz 2.11 und Satz 5.1 gilt $L(\mathcal{B}) = L(\gamma)$. Mit geringfügigen Optimierungen lässt sich der einfache 2-Mehrkopfautomat in einen einfachen (1.1)-Einwegmehrkopfautomaten transformieren. □

Es existiert noch ein Alternativbeweis.

Beweis. Gemäß Lemma 4.2 und Lemma 3.5 gelten die Inklusionen $ASL(b) \subseteq LIN \subseteq SOHL_{1.1}$. □

Lemma 5.7. $OSL(b) \subseteq SOHL_1$.

Beweis. Betrachten wir nun die gemäß Satz 5.1 konstruierten Mehrkopfautomaten. Wegen $OSL(b) \subseteq OSL(n)$ wäre eine Verschmelzung der Köpfe *top-left*, *top-right* zu *top* und *bottom-left*, *bottom-right* zu *bottom* möglich. Wegen $OSL(b) \subseteq ASL(b)$ wäre auch eine Verschmelzung der Köpfe *top-left*, *bottom-left* zu *left* bzw. *top-right*, *bottom-right* zu *right* machbar. Die Kombination beider Ideen führt zu einer Verschmelzung aller vier Köpfe zu einem Kopf. Somit gilt $OSL(b) \subseteq SOHL_1$. □

Analog zu Lemma 5.6 existiert auch hier ein Alternativbeweis.

Beweis. Gemäß Lemma 4.1 und Lemma 3.4 gelten die Inklusionen $OSL(b) \subseteq REG \subseteq SOHL_1$. □

5.2 Untere Schranken

Korollar 5.8. $AHL_1 \subseteq cRSL(b)$.

Beweis. Gemäß Lemma 3.1 und Lemma 4.5 gelten die Inklusionen $AHL_1 \subseteq REG \subseteq cRSL(b)$. □

Korollar 5.9. $OHL_{1.1} \subseteq cASL(b)$.

Beweis. Gemäß Lemma 3.2 und Lemma 4.6 gelten die Inklusionen $OHL_{1.1} \subseteq LIN \subseteq cASL(b)$. □

5.3 Optimalität der oberen Schranken

Korollar 5.10. $cSSL(b) \not\subseteq AHL_1$.

Beweis. Gemäß Lemma 4.7 und Lemma 3.1 gelten die Beziehungen $cSSL(b) \not\subseteq REG \supseteq AHL_1$. □

Korollar 5.11. $cSSL(n) \not\subseteq OHL_{1.1}$.

Beweis. Gemäß Lemma 4.8, Beobachtung 2.4 und Lemma 3.2 gelten die Beziehungen $cSSL(n) \not\subseteq CF \supseteq LIN \supseteq OHL_{1.1}$. □

Korollar 5.12. $SRSL(n) \not\subseteq OHL_{1.1}$.

Beweis. Gemäß Lemma 4.9, Beobachtung 2.4 und Lemma 3.2 gelten die Beziehungen $SRSL(n) \not\subseteq CF \supseteq LIN \supseteq OHL_{1.1}$. □

5.4 Optimalität der unteren Schranken

Korollar 5.13. $SOHL_1 \not\subseteq SSL(n)$.

Beweis. Gemäß Lemma 3.4 und Satz 4.12 gelten $SOHL_1 \supseteq REG \not\subseteq SSL(n)$. □

Korollar 5.14. $SOHL_{1.1} \not\subseteq OSL(n)$.

Beweis. Gemäß Lemma 3.5 und Korollar 4.18 gelten $SOHL_{1.1} \supseteq LIN \not\subseteq OSL(n)$. □

Korollar 5.15. $SHL_2 \not\subseteq ASL(n)$.

Beweis. Gemäß Lemma 3.9, Beobachtung 2.7 und Satz 5.1 gelten $SHL_2 \not\subseteq OHL_{*.*} \supseteq OHL_{2.2} \supseteq ASL(n)$. □

	SHL_1	SHL_2	SHL_3	SHL_4	SHL_5
$ASL(n)$	$\supset$	$\not\supseteq$	$\not\supseteq$	$\not\supseteq$	$\subset$
$cASL(n)$	$\supset$	$\not\supseteq$	$\not\supseteq$	$\not\supseteq$	$\subset$
$ASL(b)$	$\supset$	$\subset$	$\subset$	$\subset$	$\subset$
$cASL(b)$	$\supset$	$\subset$	$\subset$	$\subset$	$\subset$
$SSL(n)$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\not\supseteq$	$\subset$	$\subset$
$cSSL(n)$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\not\supseteq$	$\subset$	$\subset$
$SSL(b)$	$\not\subseteq, \not\supseteq$	$\subset$	$\subset$	$\subset$	$\subset$
$cSSL(b)$	$\not\subseteq, \not\supseteq$	$\subset$	$\subset$	$\subset$	$\subset$
$OSL(n)$	$\supset$	$\not\supseteq$	$\subset$	$\subset$	$\subset$
$cOSL(n)$	$=$	$\subset$	$\subset$	$\subset$	$\subset$
$OSL(b)$	$=$	$\subset$	$\subset$	$\subset$	$\subset$
$cOSL(b)$	$=$	$\subset$	$\subset$	$\subset$	$\subset$
$RSL(n)$	$\supset$	$\not\supseteq$	$\subset$	$\subset$	$\subset$
$cRSL(n)$	$=$	$\subset$	$\subset$	$\subset$	$\subset$
$RSL(b)$	$=$	$\subset$	$\subset$	$\subset$	$\subset$
$cRSL(b)$	$=$	$\subset$	$\subset$	$\subset$	$\subset$
$SOSL(n)$	$\not\subseteq, \not\supseteq$	$\not\supseteq$	$\subset$	$\subset$	$\subset$
$cSOSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$	$\subset$
$SOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$	$\subset$
$cSOSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(n)$	$\not\subseteq, \not\supseteq$	$\subset$	$\subset$	$\subset$	$\subset$
$cSRSL(n)$	$\subset$	$\subset$	$\subset$	$\subset$	$\subset$
$SRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$	$\subset$
$cSRSL(b)$	$\subset$	$\subset$	$\subset$	$\subset$	$\subset$

Tabelle 5.1: Beziehungen zwischen Stickersprachfamilien und Mehrkopfsprachfamilien

Kapitel 6

Ausleitung

In dieser Arbeit haben wir eine optimierte (vgl. Definition der Verklebung in Abschnitt 2.1 und [PRS98, Chapter 4]) und geringfügig erweiterte Definition des Konzeptes der Stickersysteme angegeben und die aus [PRS98] bekannten Resultate über Stickersysteme auf unsere Definition übertragen. Außerdem haben wir den Zusammenhang zwischen Stickersystemen und Mehrkopfautomaten bzw. Mehrkopfautomaten und Chomskygrammatiken untersucht und dadurch alle in [PRS98] nicht beantworteten Fragen zu den Beziehungen zwischen den Stickersprachfamilien und den Chomskysprachfamilien beantwortet. Darüber hinaus haben wir viele offene Fragen zu den Beziehungen der Stickersprachfamilien untereinander (inklusive den Abgeschlossenheitseigenschaften) und zu den Beziehungen zwischen den Stickersprachfamilien und den Mehrkopfsprachfamilien beantwortet.

Wie wir sehen, sind aber noch einige Beziehungen offen, so dass hier noch Forschungsbedarf besteht. Auch die Untersuchungen zu den Abgeschlossenheitseigenschaften sind unvollständig, so dass hier ebenfalls noch Potential für interessante Beweise verborgen sein könnte. Und natürlich können die Untersuchungen auf hier nicht berücksichtigte Stickersprachfamilien ausgedehnt werden.[1] Der einfachste Weg, neue Stickersprachfamilien zu erhalten, ist die Vereinigung oder der Schnitt bereits definierter Stickersprachfamilien. Am Ende dieser Arbeit wollen wir noch einen kleinen Beweis zu diesem Thema führen.

Lemma 6.1. *$SSL(x) \cup OSL(x) \subset ASL(x)$. $(x \in \{cb, b, cn, n\})$*

Beweis. Sei $L_1 := \{a\}^* \cup \{b\}$, $L_2 := \left\{w \in \{c,d\}^* : w = w^R\right\}$ und $L_3 := L_1 \cup L_2$, dann gilt $L_1 \notin SSL(x)$ und $L_1 \in OSL(x)$. Außerdem gilt $L_2 \in SSL(x)$ und $L_2 \notin OSL(x)$. Man kann sich nun leicht überzeugen, dass somit $L_3 \notin$

[1] Siehe dazu [KPG98] bzw. [PRS98] für z.B. primitive, balancierte, kohärente oder faire Stickersysteme.

$SSL(x) \cup OSL(x)$ gilt. Im Gegensatz dazu gilt aber $L_3 \in LIN \subseteq cASL(b) \subseteq ASL(x)$. □

Anmerkung des Autors

Die Entstehung von Stickersystemen war nicht theoretisch sondern hauptsächlich biologisch motiviert (DNA-Stränge → Dominos; Annealing, Ligation → Verklebung). Aus diesem Grund ist eine besonders kritische Hinterfragung des Nutzens dieses Konzeptes für die Sprach-, Berechenbarkeits- und Komplexitätstheorie angebracht.

Adleman's Experiment (siehe [Adl94]) zeigt, dass sich bestimmte Unterklassen der Stickersysteme sehr gut praktisch realisieren lassen, und dass durch bestimmte Erweiterungen eine starke und für die theoretische Informatik interessante Erhöhung der Ausdrucksstärke möglich ist. Die in dieser Arbeit durchgeführten Untersuchungen lassen jedoch für das hier bzw. in [PRS98] definierte Konzept der Stickersysteme folgende Kritikpunkte erkennen:

(1) Bei den Sprachfamilien $ASL(b)$, $cASL(b)$, $OSL(b)$, $cOSL(b)$, $cOSL(n)$, $RSL(b)$, $cRSL(b)$ und $cRSL(n)$ handelt es sich um alternative Charakterisierungen von LIN und REG. Der Nutzen dieser Alternativen gegenüber den regulären und den linearen Chomskygrammatiken ist jedoch fraglich.

(2) Alle nicht unter (1) genannten Stickersprachfamilien sind (vermutlich) nicht abgeschlossen unter Vereinigung, Durchschnitt, Komplement, Konkatenation, Schnitt mit regulären Chomskysprachen, ...

(3) Die Beziehungen $SSL(n) \not\subseteq CF$ und $REG \not\subseteq SSL(n)$ sind nur ein Beleg dafür, dass einige der Stickersprachenfamilien ziemlich „unausgewogen" hinsichtlich ihrer Fähigkeiten sind.

(4) Es gilt $ASL(n) \subseteq OHL_{2.2} \subset CS$. Außerdem besteht zwischen Stickersystemen und (2.2)-Einwegmehrkopfautomaten eine starke Äquivalenz.

(5) Aufgrund von Lemma 2.2, ist die Bedeutung des Konzeptes der Komplementarität zur Erhöhung der Ausdrucksstärke eher gering.

Satz 5.1 zeigt, dass es sich bei Stickersystemen im Prinzip um spezielle 4-Mehrkopfautomaten handelt. Eine Ausdehnung der Untersuchungen zu Stickersystemen auf das Konzept der Mehrkopfautomaten würde einige der genannten

Kritikpunkte beseitigen. Da einfache und allgemeine[2] Mehrkopfautomaten in der Vergangenheit bereits ausgiebig analysiert wurden und sich viele Beweise für einfache Mehrkopfautomaten auf allgemeine Mehrkopfautomaten übertragen lassen, ist auch der Nutzen dieser Untersuchung kritisch zu prüfen, zumal einige der wichtigsten Untersuchungen bereits in Abschnitt 2.3 bzw. Kapitel 3 durchgeführt bzw. wiederholt wurden.

Fazit: Das Konzept der Stickersysteme ist interessant, war ein dankbares Diplomthema und hat auch einige schöne Beweise hervorgebracht. Die Bedeutung des Konzepts für die Theoretische Informatik ist jedoch gering, da die Erkenntnisse kaum Nutzen außerhalb der Theorie der Stickersysteme haben. Ob Stickersysteme dennoch eine Daseinsberechtigung besitzen, z.B. im Rahmen des DNA-Computing, wird die Zukunft zeigen. Nach aktuellem Stand wird das Konzept der Stickersysteme jedoch zwangsläufig in der Bedeutungslosigkeit versinken und sehr schnell in Vergessenheit geraten. Möge dieses Buch dazu beitragen, den unaufhaltsamen Prozess etwas zu verlangsamen...

[2]Siehe zum Beispiel [Har72] oder [YR78, Introduction].

Literaturverzeichnis

[Wei04] Peter Weigel. *Ausdrucksstärke von Stickersystemen. Untersuchung der Ausdrucksstärke von Stickersystemen durch Vergleich mit Chomskygrammatiken und Mehrkopfautomaten.* Diplomarbeit, Martin-Luther-Universität Halle-Wittenberg, Institut für Informatik, Halle/Saale, 2004.

[KW04] D. Kuske, P. Weigel. *The rôle of the complementarity relation in Watson-Crick automata and sticker systems.* Developments in Language Theory: 8th International Conference, DLT 2004. Auckland, New Zealand, December 13-17. Proceedings. / Lecture Notes in Computer Science, Springer, Heidelberg, 3340, 2004, 272 – 283.

[PRS98] G. Păun, G. Rozenberg, A. Salomaa. *DNA-Computing. New Computing Paradigms.* Springer, Berlin Heidelberg, 1998.

[KPG98] L. Kari, G. Păun, G. Rozenberg, A. Salomaa, S. Yu. *DNA-Computing, sticker systems and universality.* Acta Informatica, 35, 5, 1998, 401–420.

[FPR98] R. Freund, G. Păun, G. Rozenberg, A. Salomaa. *Bidirectional sticker systems.* Third Annual Pacific Conference on Biocomputing, Hawaii, 1998 / World Scientific, Sigapore, 1998, 535–546.

[PR98] G. Păun, G. Rozenberg. *Sticker systems.* Theoretical Computer Science, 204, 1998, 183–203.

[Adl94] L. M. Adleman. *Molecular computation of solutions to combinatorial problems.* Science, Vol. 226, November 1994, 1021–1024.

[WW86] K. Wagner, G. Wechsung. *Computational Complexity.* Deutscher Verlag der Wissenschaften, Berlin, 1986.

[Hro83] J. Hromkovic. *One-way multihead deterministic finite automata.* Acta Informatica, 19, 4, 1983, 377–384.

[DH83] P. Duris, J. Hromkovic. *One-way simple multihead finite automata are not closed under concatenation.* Theoretical Computer Science, 27, 1983, 121–125.

[Mon80] B. Monien. *Two-Way Multihead Automata Over A One-Letter Alphabet.* R.A.I.R.O. Theoretical Informatics, 14, 1, 1980, 67–82.

[HU79] J. E. Hopcroft, J. D. Ullman. *Introduction to automata theory, languages and computation.* Addison-Wesley, 1979.

[YR78] A. C. Yao, R. L. Rivest. *$k + 1$ heads are better then k.* Journal of then ACM, 25, 2, April 1978, 337–340.

[IK75] O. H. Ibarra, C. E. Kim. *On 3-head vesus 2-head finite automata.* Inform. Control, 4, 1975, 193–200.

[Iba73] O. H. Ibarra. *On Two-Way Multihead Automata.* Journal of Computer and System Sciences, 7, 1, February 1973, 28–36.

[Har72] J. Hartmanis. *On Non-Determinancy in Simple Computing Devices.* Acta Informatica, 1, 1972, 336-344.

[Ros66] A. L. Rosenberg. *On multihead finite automata.* IBM J.R. and D., 10, 1966, 388–394.

[Hig52] G. Higman. *Ordering by divisibility in abstract algebras.* Proceedings of the London Mathematical Society, (3) 2(7), 1952, 326–336.

[Dic13] L. Dickson. *Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors.* American Journal of Mathematics, 35, 1913, 413–426.

Zum Autor: Peter Weigel wurde am 27. August 1979 in Halle an der Saale geboren. Er studierte von 1999 bis 2004 Informatik an der Martin-Luther-Universität Halle-Wittenberg (Vertiefungsrichtung Theoretische Informatik) und ist seit 2005 bei der GISA GmbH als SAP-Anwendungsentwickler bzw. seit 2010 als Prozess- und IT-Berater für den SAP Solution Manager tätig.