

Institut für Formale Methoden der Informatik

Universität Stuttgart
Universitätsstraße 38
D-70569 Stuttgart

Bachelorarbeit Nr.

**Effiziente Algorithmen für das
Separierbarkeitsproblem der
alternierungsfreien Logik über
unendlichen Wörtern**

Manuel Göggelmann

Studiengang: Informatik

Prüfer: PD Dr. Manfred Kufleitner

Betreuer: PD Dr. Manfred Kufleitner

Beginn am: 20. April 2015

Beendet am: 10. November 2015

CR-Klassifikation: F.4.1; F.4.3

Kurzfassung

Das Separierbarkeitsproblem befasst sich mit der Frage, gegeben zwei Mengen aus einer Klasse, ob es möglich ist, sie durch eine weitere Menge aus einer kleineren Klasse zu separieren. Für den Fall der Separierbarkeit von regulären Sprachen durch eine piecewise testable Sprache über unendlichen Wörtern wird in dieser Arbeit ein Entscheidungsalgorithmus mit polynomialer Laufzeit vorgestellt. Der Beweis orientiert sich an einer Arbeit über den entsprechenden Fall der Separierbarkeit über endlichen Wörtern von L. van Rooijen und M. Zeitoun.

Inhaltsverzeichnis

1 Einleitung	7
2 Grundlagen	9
2.1 Sprachen und Automaten	9
2.2 Piecewise testable Sprachen und Separierbarkeit	9
2.3 Faktorisierungsmuster und (\vec{u}, \vec{B}) -Pfade	10
2.4 ω -Faktorisierungsmuster und ω -(\vec{u}, \vec{B})-Pfade	12
2.5 Faktorisierungswälder	13
3 Separierbarkeit durch eine piecewise testable Sprache über endlichen Wörtern	15
3.1 Ein Algorithmus in Polynomialzeit	15
3.2 Charakterisierung der PT-Separierbarkeit	16
4 Separierbarkeit durch eine piecewise testable Sprache über unendlichen Wörtern	23
4.1 Ein Algorithmus in Polynomialzeit	23
4.2 Charakterisierung der PT-Separierbarkeit	24
5 Zusammenfassung	29
Literaturverzeichnis	31

1 Einleitung

Allgemein sind zwei Mengen L und K *separierbar* durch eine Menge S , falls $L \subseteq S$ und $S \cap K = \emptyset$. Dann wird S ein *Separator* genannt. Das Separierbarkeitsproblem ist die folgende Problemstellung. Betrachtet wird eine Klasse C von Mengen und eine Unterklasse C_0 von C . Zu zwei gegebenen Elementen X, Y aus C wird gefragt, ob es immer ein Element S aus der Unterklasse C_0 gibt, welches X und Y separiert. Insbesondere sind wir an der Entscheidbarkeit dieses Problems und der auftretenden Komplexität dieses Entscheidungsproblems interessiert. Der Fokus liegt in dieser Arbeit auf dem Separierbarkeitsproblem für die Klasse der regulären Sprachen und der Unterklasse der piecewise testable Sprachen, in der wir nach Separatoren suchen.

Für den Fall der Separierbarkeit von regulären Sprachen durch eine piecewise testable Sprache über **endlichen** Wörtern haben Rooijen und Zeitoun in [RZ13] einen Entscheidungsalgorithmus mit polynomialer Laufzeit angegeben. Dabei bezieht sich polynomial auf die Größe des Alphabets und die Größe der nichtdeterministischen Automaten (NFA), wobei angenommen wird, dass die zu separierenden regulären Sprachen durch NFAs gegeben sind. Entscheidend bei diesem Vorgehen ist, dass die Nicht-Separierbarkeit durch einen speziellen, gemeinsamen Pfad in beiden Automaten charakterisiert werden kann. Eine wichtige Rolle bei dem Beweis spielt die Anwendung des Faktorisierungswaldtheorems von Simon [Sim90].

Almeida hat mit dem sogenannten profiniten Ansatz in [Alm96] eine allgemeine Beziehung zwischen der profiniten Halbgruppentheorie und dem Separierbarkeitsproblem gefunden. Darauf aufbauend beschreibt [AZGJ97] einen Algorithmus der polynominal bzgl. der Größe der Automaten und exponentiell bzgl. der Größe des Alphabets ist. Das Vorgehen von Rooijen und Zeitoun folgt den selben Mustern, aber enthält die folgenden Verbesserungen: Verbesserte Komplexität, da der Algorithmus polynomial bzgl. der Größe der Automaten und der Größe des Alphabets ist. Zusätzlich wird kein Gebrauch der Theorie der profiniten Halbgruppen gemacht, da nur elementare Konzepte verwendet werden.

Diese Arbeit zeigt, wie die Methoden von Rooijen und Zeitoun auf **unendliche** Wörter verallgemeinert werden können. Das Hauptresultat ist, dass in Polynomzeit das Separierbarkeitsproblem von regulären Sprachen durch eine pieceweise testable Sprache über **unendlichen** Wörtern entschieden werden kann.

Die **Gliederung** dieser Ausarbeitung ist wie folgt. In Kapitel 2 werden die benötigten Grundlagen erklärt. Der Beweis für den entsprechenden Fall über **endlichen** Wörtern von Rooijen und Zeitoun wird in Kapitel 3 zur Orientierung und zum Nachschlagen nochmals wiedergegeben. Kapitel 4 enthält die Hauptleistung dieser Arbeit. Für die Separierbarkeit

1 Einleitung

durch eine piecewise testable Sprache über unendlichen Wörtern wird zunächst ein Polynomialzeitalgorithmus angegeben. Danach wird die Charakterisierung von Separierbarkeit über unendlichen Wörtern diskutiert. Als Abschluss im letzten Kapitel 5 fassen wir die wichtigsten Schritte und Ergebnisse nochmal zusammen.

2 Grundlagen

2.1 Sprachen und Automaten

Im folgenden werden wir stets von einem endlichen Alphabet A ausgehen. Wir schreiben A^* für die Menge aller endlichen Wörter und A^ω für die Menge aller unendlichen Wörter über A . Sei $\inf(\alpha)$ die Menge aller Buchstaben, die unendlich oft in α vorkommen. Eine Einführung in unendliche Wörter bietet z.B. [DK11]. Sei $A^\infty = A^* \cup A^\omega$. Das leere Wort wird mit ϵ bezeichnet. Zu einem Wort $u \in A^*$ wird das kleinste Teilalphabet $B \subseteq A$, so dass $u \in B^*$, das Alphabet von u genannt und $\text{alph}(u)$ geschrieben; d.h. $\text{alph}(u)$ enthält alle in u vorkommenden Zeichen. Für ein Teilalphabet $B \subseteq A$, bezeichnet B^* die Menge von Wörtern, deren Alphabet genau B ist:

$$B^* := \{w \in B^* \mid \text{alph}(w) = B\}.$$

Wie üblich wird ein *nichtdeterministischer endlicher Automat* (NFA) bzw. *Büchi-Automat* (BA) über A durch ein 5-Tupel $\mathcal{A} = (Q, A, I, F, \delta)$ angegeben, wobei Q die Menge der Zustände, $I \subseteq Q$ die Menge der Startzustände, $F \subseteq Q$ die Menge der Endzustände und $\delta \subseteq Q \times A \times Q$ die Überführungsrelation darstellt. Ein NFA akzeptiert ein Eingabewort $w \in A^*$, wenn der Automat nach lesen von w einen Endzustand erreicht hat. Ein BA akzeptiert ein Eingabewort $w \in A^\omega$, falls der Automat beim lesen von w unendlich oft einen Endzustand durchläuft. Für beide Automaten sei $L(\mathcal{A})$ die durch \mathcal{A} akzeptierte Sprache von Wörtern. Die deterministischen Gegenstücke zu NFA und BA sind zum einen die endlichen deterministischen Automaten (DFA) und zum anderen die deterministischen Büchi-Automaten (DBA). Wenn nicht näher spezifiziert, meinen wir mit Automat in diesem Kapitel beide (nichtdeterministische-) Automatenarten, also nichtdeterministische endliche Automaten bzw. Büchi-Automaten.

Für einen Automaten \mathcal{A} und einen Zustand $p \in Q$ definiert $\text{scc}(p, \mathcal{A})$ die starke Zusammenhangskomponente (SZK) von p in \mathcal{A} , d.h. die Menge der von p aus erreichbaren Zustände im Automaten. Außerdem sei $\text{alph_scc}(p, \mathcal{A})$ die Menge der Beschriftungen von allen Übergängen, die in dieser starken Zusammenhangskomponente vorkommen. Die Einschränkung des Automaten \mathcal{A} auf ein Teilalphabet $B \subseteq A$ sei definiert durch $\mathcal{A}|_B := (Q, A, I, F, \delta \cap (Q \times B \times Q))$, d.h. es fallen alle Übergänge weg, die nicht mit einem Zeichen aus B beschriftet sind.

2.2 Piecewise testable Sprachen und Separierbarkeit

Zunächst definieren wir, wann ein endliches Wort ein (verteiltes) Teilwort eines endlichen oder unendlichen Wortes ist. Für $u \in A^*, v \in A^\infty$ gelte $u \leqslant v$ („ u ist ein Teilwort von v “), wenn $u = a_1 \dots a_n$ mit $a_i \in A$ und $v \in A^* a_1 A^* \dots a_n A^*$ für $v \in A^*$ bzw. $v \in A^* a_1 A^* \dots a_n A^\omega$

für $v \in A^\omega$. Die Relation \leqslant ist eine Halbordnung auf A^* , wenn für v nur endliche Wörter zugelassen werden: Es ist klar, dass diese Relation reflexiv (jedes Wort ist Teilwort von sich selbst), antisymmetrisch (wenn $u \leqslant v$ und umgekehrt, dann $u = v$) und transitiv (wenn $u \leqslant v$ und $v \leqslant w$, dann ist $u \leqslant w$) ist. Wir setzen

$$\text{Sub}_n(u) := \{w \in A^* \mid w \leqslant u \wedge |w| \leqslant n\}$$

als die Menge aller Teilwörter von $u \in A^\infty$, deren Länge maximal n ist.

Haben zwei Wörter $u, v \in A^*$ die gleichen Teilwörter bis zur Länge n , dann sagen wir, dass sie \sim_n -äquivalent sind:

$$u \sim_n v \iff \text{Sub}_n(u) = \text{Sub}_n(v).$$

Zwei Wörter $\alpha, \beta \in A^\omega$ heißen \sim_n -äquivalent ($\alpha \sim_n \beta$), wenn sie die gleichen Teilwörter bis zur Länge n haben, d.h.

$$\text{Sub}_n(\alpha) = \text{Sub}_n(\beta).$$

und zusätzlich in beiden Wörtern dieselben Buchstaben unendlich oft vorkommen: $\inf(\alpha) = \inf(\beta)$.

Aus der Gleichheit von Mengen folgen die Eigenschaften Reflexivität, Symmetrie und Transitivität, so dass \sim_n tatsächlich eine Äquivalenzrelation ist. Eine reguläre Sprache $L \subseteq A^*$ bzw. $L \subseteq A^\omega$ heißt *piecewise testable* (PT), falls L eine endliche boolesche Kombination von Sprachen der Form $A^*a_1A^*\dots a_nA^*$ bzw. $A^*a_1A^*\dots a_nA^\omega$ ist, wobei $a_i \in A$. Ob ein Wort w zu einer PT-Sprache gehört, ist daher bestimmt durch die Menge seiner Teilwörter bis zu einer bestimmten Länge. Das bedeutet, dass eine reguläre Sprache L genau dann pieceweise testable ist, wenn es ein $n \in \mathbb{N}$ gibt, so dass L eine Vereinigung von \sim_n -Äquivalenzklassen ist.

Zwei reguläre Sprachen $L, K \in A^*$ bzw. $L, K \in A^\omega$ heißen *PT-separierbar*, wenn es eine piecewise testable Sprache $S \in A^*$ bzw. $S \in A^\omega$ (Separator) gibt, die sie trennt, d.h. falls gilt

$$L \subseteq S \text{ und } K \cap S = \emptyset.$$

2.3 Faktorisierungsmuster und (\vec{u}, \vec{B}) -Pfade

Für ein Wort $u \in A^*$, ein Teialphabet $B \subseteq A$ und Zustände $p, q \in Q$ vereinbaren wir folgende Notation, um bestimmte Arten von Pfaden in Automaten zu charakterisieren.

- $p \xrightarrow{u} q$ bezeichne einen Pfad von p nach q , der mit u beschriftet ist.
- $p \xrightarrow{B^*} q$ bezeichne einen Pfad von p nach q , wobei alle Übergänge mit Zeichen aus B beschriftet sind.
- $p \xrightarrow{B^\circledast} q$ bezeichne einen Pfad von p nach q , wobei alle Übergänge mit Zeichen aus B beschriftet sind und zusätzlich jedes Zeichen aus B mindestens einmal auftritt.

2.3 Faktorisierungsmuster und (\vec{u}, \vec{B}) -Pfade

Sei \mathcal{A} ein DFA über A . Für Wörter $u_0, \dots, u_p \in A^*$ und nichtleeren Teilalphabeten $B_1, \dots, B_p \subseteq A$, setzen wir $\vec{u} = (v_0, \dots, v_p)$ und $\vec{B} = (B_1, \dots, B_p)$. Schließlich fassen wir \vec{u} und \vec{B} zu einem Paar (\vec{u}, \vec{B}) zusammen und nennen es ein *Faktorisierungsmuster*. Ein (\vec{u}, \vec{B}) -Pfad ist ein erfolgreicher Pfad (der von einem Startzustand zu einem Endzustand führt), der die folgende Form hat:

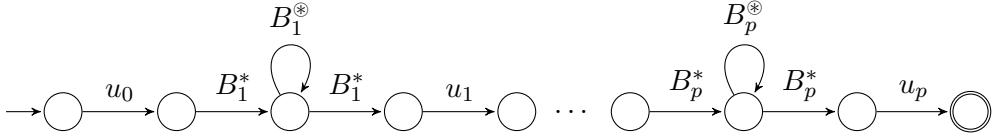


Abbildung 2.1: Ein (\vec{u}, \vec{B}) -Pfad

Bemerkung 2.1. Ein endlicher Automat \mathcal{A} lässt genau dann einen (\vec{u}, \vec{B}) -Pfad zu, wenn es geeignete $x_i, y_i, z_i \in A^*$ ($i \in \{1, \dots, p\}$, $p \in \mathbb{N}_0$) gibt, so dass $L(\mathcal{A})$ eine Sprache der Form

$$u_0(x_1y_1^*z_1)u_1 \dots (x_py_p^*z_p)u_p$$

enthält, wobei $\text{alph}(x_i) \cup \text{alph}(z_i) \subseteq \text{alph}(y_i) = B_i$, für alle $i \in \{1, \dots, p\}$.

Ein Faktorisierungsmuster (\vec{u}, \vec{B}) heißt *echt*, wenn

- i) $\text{last}(u_i) \notin B_{i+1} \wedge \text{first}(u_i) \notin B_i$, für $i = 1, \dots, p-1$
- ii) $\text{last}(u_0) \notin B_1 \wedge \text{first}(u_p) \notin B_p$
- iii) $u_i = \epsilon \Rightarrow (B_i \subsetneq B_{i+1} \wedge B_{i+1} \subsetneq B_i)$, für $i = 1, \dots, p-1$.

Die Eigenschaften drücken aus, dass die Teilalphabete B_i so groß wie möglich und die Wörter u_i so kurz wie möglich gewählt werden.

Für ein gegebenes Faktorisierungsmuster (\vec{u}, \vec{B}) mit $\vec{u} = (u_0, \dots, u_p)$ und $\vec{B} = (B_1, \dots, B_p)$, definieren wir eine Sprache

$$\mathcal{L}(\vec{u}, \vec{B}, n) := u_0(B_1^*)^n u_1 \dots (B_p^*)^n u_p.$$

Eine Folge $(w_n)_n$ von Wörtern heißt (\vec{u}, \vec{B}) -adäquat, wenn

$$\forall n \geq 0 : w_n \in \mathcal{L}(\vec{u}, \vec{B}, n).$$

Eine Folge heißt adäquat, wenn sie (\vec{u}, \vec{B}) -adäquat zu einem geeigneten Faktorisierungsmuster (\vec{u}, \vec{B}) ist. Aus den Beobachtungen

- $u = u_1 \dots u_k$ mit $u_k \in B \Rightarrow u_1 \dots u_k (B^*)^n \subseteq u_1 \dots u_{k-1} (B^*)^n$,
- $u = u_1 \dots u_k$ mit $u_1 \in B \Rightarrow (B^*)^n u_1 \dots u_k \subseteq (B^*)^n u_2 \dots u_k$,
- $B_i \subseteq B_{i+1} \Rightarrow (B_i^*)^n (B_{i+1}^*)^n \subseteq (B_{i+1}^*)^n$ und
- $B_{i+1} \subseteq B_i \Rightarrow (B_i^*)^n (B_{i+1}^*)^n \subseteq (B_i^*)^n$

schließen wir, dass es zu jeder adäquaten Folge $(w_n)_n$ ein echtes Faktorisierungsmuster (\vec{u}, \vec{B}) gibt, so dass $(w_n)_n$ selbst (\vec{u}, \vec{B}) -adäquat ist.

2.4 ω -Faktorisierungsmuster und ω -(\vec{u}, \vec{B})-Pfade

Für Separierbarkeit über unendlichen Wörtern in Kapitel 4 sind leichte Anpassungen der Begriffe aus dem vorherigen Abschnitt notwendig, die hier erläutert werden.

Sei \mathcal{A} ein DBA über A . Für Wörter $u_0, \dots, u_p \in A^*$ und nichtleeren Teilalphabeten $B_1, \dots, B_p, B_{p+1} \subseteq A$, setzen wir $\vec{u} = (v_0, \dots, v_p)$ und $\vec{B} = (B_1, \dots, B_p, B_{p+1})$. Schließlich fassen wir \vec{u} und \vec{B} zu einem Paar (\vec{u}, \vec{B}) zusammen und nennen es ein ω -Faktorisierungsmuster. Aus Bequemlichkeitsgründen nennen wir dann (\vec{u}, \vec{C}) mit $\vec{C} = (B_1, \dots, B_p)$ das von (\vec{u}, \vec{B}) induzierte Faktorisierungsmuster. Ein ω -(\vec{u}, \vec{B})-Pfad ist ein erfolgreicher Pfad (der von einem Startzustand zu einem Endzustand führt, der über eine Schleife unendlich oft durchlaufen werden kann, um zu akzeptieren), der die folgende Form hat:

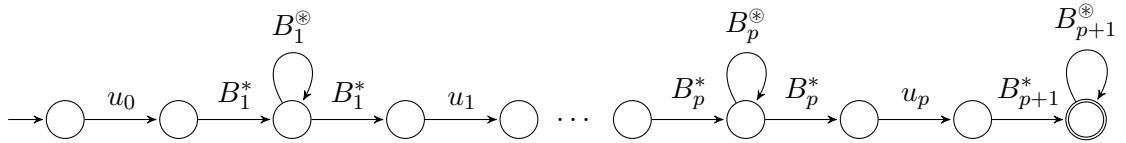


Abbildung 2.2: Ein ω -(\vec{u}, \vec{B})-Pfad

Dabei bezeichnen die Kanten Folgen von Übergängen, wie im vorherigen Abschnitt definiert.

Bemerkung 2.2. Ein Büchi-Automat \mathcal{A} lässt genau dann einen ω -(\vec{u}, \vec{B})-Pfad zu, wenn es geeignete $s, x_i, y_i, z_i \in A^*, \alpha \in A^\omega$ ($i \in \{1, \dots, p\}$, $p \in \mathbb{N}_0$) gibt, so dass $L(\mathcal{A})$ eine Sprache der Form

$$u_0(x_1y_1^*z_1)u_1 \dots (x_py_p^*z_p)u_ps\alpha$$

enthält, wobei $\text{alph}(x_i) \cup \text{alph}(z_i) \subseteq \text{alph}(y_i) = B_i$, für alle $i \in \{1, \dots, p\}$, $\text{alph}(s) \subseteq \text{alph}(\alpha)$ und $\alpha \in (B_{p+1}^*)^\omega$.

Ein ω -Faktorisierungsmuster (\vec{u}, \vec{B}) heißt *echt*, falls das dazugehörige induzierte Faktorisierungsmuster von (\vec{u}, \vec{B}) echt ist. D.h. die selbe Definition wie im vorherigen Abschnitt wird hier verwendet und die Präsenz von B_{p+1} spielt dabei keine Rolle.

Für ein gegebenes ω -Faktorisierungsmuster (\vec{u}, \vec{B}) mit $\vec{u} = (u_0, \dots, u_p)$ und $\vec{B} = (B_1, \dots, B_p, B_{p+1})$, definieren wir eine Sprache

$$\mathcal{L}(\vec{u}, \vec{B}, n) := u_0(B_1^*)^n u_1 \dots (B_p^*)^n u_p (B_{p+1}^*)^\omega.$$

Eine Folge $(w_n)_n$ von unendlichen Wörtern heißt ω -(\vec{u}, \vec{B})-adäquat, wenn

$$\forall n \geq 0 : w_n \in \mathcal{L}(\vec{u}, \vec{B}, n).$$

Eine Folge heißt kurz ω -adäquat, wenn sie ω -(\vec{u}, \vec{B})-adäquat zu einem geeigneten ω -Faktorisierungsmuster (\vec{u}, \vec{B}) ist. Analog zum vorherigen Abschnitt gilt: Zu jeder ω -adäquaten Folge $(w_n)_n$ gibt es ein echtes ω -Faktorisierungsmuster (\vec{u}, \vec{B}) , so dass $(w_n)_n$ selbst ω -(\vec{u}, \vec{B})-adäquat ist.

2.5 Faktorisierungswälder

Ein *Faktorisierungsbaum* von einem nichtleeren Wort x ist ein endlicher, geordneter, unbeschränkt verzweigter Baum $T(x)$, so dass gilt:

- i) alle Blätter von $T(x)$ sind jeweils mit einzelnen Buchstaben beschriftet,
- ii) alle inneren Knoten von $T(x)$ besitzen mindestens zwei Kinder,
- iii) hat ein Knoten, der mit y beschriftet ist, k Kinder, die mit y_1, \dots, y_k von links nach rechts beschriftet sind, dann ist $y = y_1 \dots y_k$.

Sei S eine endliche Halbgruppe und $\varphi : A^+ \rightarrow S$ ein Morphismus. Ein Faktorisierungsbaum heißt φ -Ramsey-artig, falls jeder innere Knoten entweder zwei Kinder hat oder $k \geq 3$ Kinder y_1, \dots, y_k , wobei dann $\varphi(y_1) = \dots = \varphi(y_k) = e$, für ein idempotentes Element e aus S .

Das Faktorisierungswaldtheorem (siehe [Sim90] und [Kuf08] für Beweise) besagt, dass es für jedes Wort w einen φ -Ramsey-artigen Faktorisierungsbaum $T(w)$ gibt, dessen Höhe $h(T(w))$ nach oben durch eine Konstante $3|S|$ beschränkt ist.

3 Separierbarkeit durch eine piecewise testable Sprache über endlichen Wörtern

Theorem 3.1. Gegeben seien zwei NFAs. Dann ist in Polynomialzeit entscheidbar, bzgl. der Anzahl der Zustände und der Größe des Alphabets, ob die Sprachen, die durch die NFAs akzeptiert werden, PT-separierbar sind.

Aus den folgenden zwei Propositionen folgt direkt Theorem 3.1.

Proposition 3.2. Seien \mathcal{A}_1 und \mathcal{A}_2 zwei NFAs. Dann sind $L(\mathcal{A}_1)$ und $L(\mathcal{A}_2)$ genau dann nicht PT-separierbar, wenn ein Faktorisierungsmuster (\vec{u}, \vec{B}) existiert, so dass \mathcal{A}_1 und \mathcal{A}_2 einen gemeinsamen (\vec{u}, \vec{B}) -Pfad zulassen.

Proposition 3.3. Gegeben seien zwei NFAs \mathcal{A}_1 und \mathcal{A}_2 , dann kann in Polynomialzeit entschieden werden, bzgl. der Anzahl der Zustände und der Größe des Alphabets, ob ein Faktorisierungsmuster (\vec{u}, \vec{B}) existiert, so dass beide NFAs einen gemeinsamen (\vec{u}, \vec{B}) -Pfad zulassen.

3.1 Ein Algorithmus in Polynomialzeit

Beweis. (Proposition 3.3) Zunächst zeigen wir, dass das folgende Problem in PTIME (bzgl. $|A|, |\mathcal{A}_1|, |\mathcal{A}_2|$) entscheidbar ist.

Gegeben: Zustände $p_1, q_1, r_1 \in Q_1$ und $p_2, q_2, r_2 \in Q_2$.

Frage: Existiert ein nichtleeres Teialphabet $B \subseteq A$, so dass beide Automaten ein gemeinsames Muster der folgenden Form enthalten?

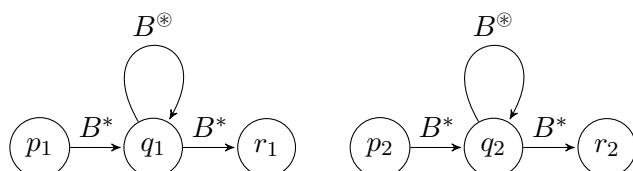


Abbildung 3.1: Ein gemeinsames Muster in \mathcal{A}_1 und \mathcal{A}_2

Dazu wird eine Folge von Alphabeten C_i abnehmender Größe berechnet, die B von oben her approximieren sollen. Angenommen, es existiert ein solch gefordertes Teialphabet B , dann muss B in $C_1 := \text{alph_scc}(q_1, \mathcal{A}_1) \cap \text{alph_scc}(q_2, \mathcal{A}_2)$ enthalten sein. Indem z.B. der Algorithmus von Tarjan, der starke Zusammenhangskomponenten in Linearzeit berechnet,

verwendet wird, kann C_1 ebenfalls in Linearzeit berechnet werden. Anschließend schränken wir beide Automaten auf das Alphabet C_1 ein und iterieren den Prozess, und erhalten so eine Folge $(C_i)_i$ nach der Vorschrift:

$$C_{i+1} := \text{alph_scc}(q_1, \mathcal{A}_1|_{C_i}) \cap \text{alph_scc}(q_2, \mathcal{A}_2|_{C_i}).$$

Nach einer endlichen Anzahl von $n \leq |A|$ Schritten ist die Folge stabil, d.h. es gilt $C_n = C_{n+1}$. Falls $C_n = \emptyset$, dann gibt es kein nichtleeres Teilalphabet $B \subseteq A$, so dass es eine B^* -Schleife um q_1 und q_2 gibt.

Falls $C_n \neq \emptyset$, dann ist $B = C_n$ das maximale nichtleere Alphabet B , so dass es B^* -Schleifen um q_1 in \mathcal{A}_1 und q_2 in \mathcal{A}_2 gibt. Es verbleibt zu zeigen, dass es Pfade $p_1 \xrightarrow{B^*} q_1 \xrightarrow{B^*} r_1$ und $p_2 \xrightarrow{B^*} q_2 \xrightarrow{B^*} r_2$ gibt. Dies kann jedoch z.B. mittels Tiefensuche in durchgeführt werden.

Zusammengefasst erhalten wir für die Laufzeit: Die Anzahl der Iterationen n bis $C_n = C_{n+1}$ erreicht wird, ist beschränkt durch $|A|$ und jede einzelne Berechnung von C_i kann in Linearzeit bzgl. der Größe von \mathcal{A}_1 und \mathcal{A}_2 durchgeführt werden. Das impliziert, dass in Polynomialzeit (bzgl. $|A|$ und der Größe der NFAs) entschieden werden kann, ob man ein gemeinsames Muster wie in Abbildung 3.1 in \mathcal{A}_1 und in \mathcal{A}_2 vorfinden kann.

Nun bauen wir aus \mathcal{A}_1 und \mathcal{A}_2 zwei modifizierte NFAs $\tilde{\mathcal{A}}_1$ und $\tilde{\mathcal{A}}_2$ wie folgt. Zuerst initialisieren wir $\tilde{\mathcal{A}}_i$ als eine Kopie von \mathcal{A}_i . Für jedes 4-Tupel $\tau = (p_1, r_1, p_2, r_2) \in Q_1^2 \times Q_2^2$ für das es ein Alphabet $B \subseteq A$, Zustände $q_1 \in Q_1, q_2 \in Q_2$ und Pfade $p_i \xrightarrow{B^*} q_i \xrightarrow{B^*} q_i \xrightarrow{B^*} r_i$ (wie in Abbildung 3.1) gibt, fügen wir in beide Automaten einen neuen Buchstaben a_τ zum Alphabet hinzu. Ferner fügen wir Abkürzungen in Form von Übergängen $p_1 \xrightarrow{a_\tau} r_1$ und $p_2 \xrightarrow{a_\tau} r_2$ in beide Automaten hinzu. Die Zahl von solchen Tupeln $(p_1, q_1, r_1, p_2, q_2, r_2)$ beträgt höchstens $|Q_1|^3 \cdot |Q_2|^3$ und mit dem oben gezeigten Teil erhalten wir, dass die Berechnung dieser neuen Übergänge in Polynomialzeit durchgeführt werden kann. Weiter folgt, dass die Berechnung der modifizierten Automaten $\tilde{\mathcal{A}}_1$ und $\tilde{\mathcal{A}}_2$ in PTIME möglich ist.

Nach Konstruktion dieser Automaten gibt es ein Faktorisierungsmuster (\vec{u}, \vec{B}) , so dass $\mathcal{A}_1, \mathcal{A}_2$ genau dann einen gemeinsamen (\vec{u}, \vec{B}) -Pfad zulassen, wenn $L(\tilde{\mathcal{A}}_1) \cap L(\tilde{\mathcal{A}}_2) \neq \emptyset$. Dies kann tatsächlich in Polynomialzeit entschieden werden, weil $\tilde{\mathcal{A}}_1$ und $\tilde{\mathcal{A}}_2$ in PTIME gebaut wurden und dasselbe für das Bilden des Kreuzproduktautomaten nach der üblichen Konstruktion gilt, der den Schnitt zweier regulärer Sprachen erkennt. \square

3.2 Charakterisierung der PT-Separierbarkeit

Als nächstes soll Proposition 3.2 bewiesen werden. Wir beginnen mit der Rückrichtung. Die Hinrichtung wird in Lemma 3.9 bewiesen.

Behauptung 3.4. Seien $x, x', y, y', z, z' \in A^*$ endliche Wörter, so dass

- i) $\text{alph}(x) \cup \text{alph}(z) \subseteq \text{alph}(y)$,
- ii) $\text{alph}(x') \cup \text{alph}(z') \subseteq \text{alph}(y') = \text{alph}(y)$.

Dann ist $xy^n z \sim_n x'y'^n z'$, für alle $n \in \mathbb{N}$.

Beweis. Die Inklusionen $\text{alph}(y)^{\leq n} = \text{Sub}_n(y^n) \subseteq \text{Sub}_n(xy^n z) \subseteq \text{alph}(y)^{\leq n}$ implizieren $\text{Sub}_n(xy^n z) = \text{alph}(y)^{\leq n}$, analog $\text{Sub}_n(x'y'^n z') = \text{alph}(y')^{\leq n} = \text{alph}(y)^{\leq n}$ und man erhält die Behauptung $xy^n z \sim_n x'y'^n z'$. \square

Lemma 3.5. Seien $\mathcal{A}_1, \mathcal{A}_2$ NFAs, die einen gemeinsamen (\vec{u}, \vec{B}) -Pfad zulassen. Dann sind die Sprachen $L = L(\mathcal{A}_1)$ und $K = L(\mathcal{A}_2)$ nicht PT-separierbar.

Beweis. Wir nehmen an, S ist eine piecewise testable Sprache mit $L \subseteq S$. Nach Bemerkung 2.1 gibt es passende $x_i, y_i, z_i \in A^*$, so dass L eine Sprache der Form

$$\tilde{L} := u_0(x_1 y_1^* z_1) u_1 \dots (x_p y_p^* z_p) u_p$$

enthält, wobei $\text{alph}(x_i) \cup \text{alph}(z_i) \subseteq \text{alph}(y_i) = B_i$ für alle i . Entsprechend enthält K eine Sprache der Form

$$\tilde{K} := u_0(x'_1 y'_1^* z'_1) u_1 \dots (x'_p y'_p^* z'_p) u_p,$$

mit $\text{alph}(x'_i) \cup \text{alph}(z'_i) \subseteq \text{alph}(y'_i) = B_i$ für alle i . Das Ziel ist zu zeigen, dass es zu jedem $n \in \mathbb{N}$ ein $w_1 \in \tilde{L}$ sowie ein $w_2 \in \tilde{K}$ gibt, so dass die Wörter w_1 und w_2 zueinander \sim_n -äquivalent sind. Durch Anwendung von Behauptung 3.4 erhält man, dass für alle $1 \leq i \leq p$ und $n \in \mathbb{N}$ gilt

$$x_i y_i^n z_i \sim_n x'_i y'_i'^n z'_i.$$

Da die Äquivalenz \sim_n zudem eine Kongruenz ist, erhalten wir für alle $n \in \mathbb{N}$

$$u_0(x_1 y_1^n z_1) u_1 \dots (x_p y_p^n z_p) u_p \sim_n u_0(x'_1 y'_1'^n z'_1) u_1 \dots (x'_p y'_p'^n z'_p) u_p.$$

Die linke Seite dieser Äquivalenz stellt unser gewünschtes Wort $w_1 \in \tilde{L}$ und die rechte Seite unser gewünschtes Wort $w_2 \in \tilde{K}$ dar, die beide zueinander \sim_n -äquivalent sind. Folglich liegen w_1 und w_2 in derselben Äquivalenzklasse. Die piecewise testable Sprache S bildet eine Vereinigung aus \sim_n -Äquivalenzklassen und daher sind S und K nicht disjunkt. Das wiederum bedeutet, dass L und K nicht separierbar sind. \square

Wir fahren mit der Hinrichtung von Proposition 3.2 fort.

Lemma 3.6. Jede Folge $(w_n)_n$ von endlichen Wörtern $w_n \in A^*$ besitzt eine adäquate Teilfolge.

Beweis. Die zentrale Idee hierbei ist das Faktorisierungswaldtheorem mit dem Morphismus $\text{alph} : A^+ \rightarrow 2^A$ anzuwenden. Ausgehend von einer beliebigen Wortfolge $(w_n)_n$ betrachten wir die Folge $(T(w_n))_n$ von alph -Ramsey-artigen Faktorisierungsbäumen, gegeben durch das Faktorisierungswaldtheorem. D.h. die Höhe jedes Faktorisierungsbäums $T(w_n)$ ist nach oben durch eine endliche Konstante beschränkt. An dieser Stelle dürfen wir annehmen, dass die Folge der Höhen der Bäume $T(w_n)$ eine Konstante $H \geq 0$ ist. Gegebenenfalls nehmen wir aus der Folge eine passende Teilfolge heraus, um diese Bedingung sicherzustellen. Die Aussage des Lemmas wird nun mit Induktion nach H gezeigt.

Wenn $H = 0$, dann muss jedes Wort w_n aus einem einzigen Buchstaben bestehen. Folglich kann aus $(w_n)_n$ eine konstante Teilfolge herausgezogen werden, die selbst adäquat ist.

Für die Stelligkeit der Wurzel von $T(w_n)$ schreiben wir $\text{arity}(w_n)$ und nennen es die *Stelligkeit* von w_n . Wenn $H > 0$, können im Induktionsschritt zwei Fälle unterschieden werden.

1. $(w_n)_n$ enthält eine Teilfolge mit beschränkter Stelligkeit. Aus diesem Grund können wir aus w_n eine Teilfolge mit konstanter Stelligkeit $\hat{K} \geq 0$ gewinnen. Jedes w_n kann daher in \hat{K} Faktoren zerlegt werden

$$w_n = w_{n,1} \dots w_{n,\hat{K}},$$

wobei das i -te Kind der Wurzel im Baum $T(w_n)$ mit $w_{n,i}$ beschriftet ist. Der Unterbaum von jedem $w_{n,i}$, der selbst auch alph-Ramsey-artig ist, kann höchstens die Höhe $H - 1$ haben. Nach Induktionsvoraussetzung können wir aus $(w_{n,i})_n$ eine adäquate Teilfolge gewinnen. Das machen wir iterativ für $i = 1, \dots, \hat{K}$ und gewinnen so aus $(w_n)_n$ eine Teilfolge $(w_{\sigma(n)})_n$, so dass alle $(w_{\sigma(n),i})_n$ adäquat sind. Das impliziert, dass die Teilfolge $(w_{\sigma(n)})_n$ von $(w_n)_n$ selbst adäquat ist. Dabei wurde ausgenutzt, dass endliche Produkte von adäquaten Folgen selbst adäquat sind.

2. Die Stelligkeit von w_n strebt gegen unendlich für $n \rightarrow \infty$. Wir dürfen annehmen, dass für jedes $n \in \mathbb{N}$, $\text{arity}(w_n) \geq \max\{n, 3\}$ (ggf. ziehen wir eine passende Teilfolge heraus). Da die Stelligkeit von allen Wörtern in dieser Folge mindestens drei ist, werden alle Kinder der Wurzel auf das gleiche idempotente Element in 2^A abgebildet. Das wiederum bedeutet, dass jedes Wort der Teilfolge von der Form

$$w_{\sigma(n)} = w_{n,1} \dots w_{n,K_n}$$

ist, mit $K_n \geq n$ und es gibt ein Teilalphabet $B \subseteq A$ mit $w_{n,i} \in B^\circledast$ für alle i . Daraus schließen wir, $w_{\sigma(n)} \in (B^\circledast)^{K_n} \subseteq (B^\circledast)^n$. Daher ist $(w_{\sigma(n)})_n$ adäquat.

□

An dieser Stelle wird für Lemma 3.8 das technische Lemma 8.2.5 aus [Alm94] benötigt.

Lemma 3.7. Seien X, Y endliche Mengen und (P, \leq) eine Halbordnung.

Seien $f : X \rightarrow Y$, $g : Y \rightarrow X$, $p : X \rightarrow P$, $q : Y \rightarrow P$ Abbildungen, die die folgenden Eigenschaften erfüllen:

- i) $\forall x \in X : p(x) \leq q(f(x))$,
- ii) $\forall y \in Y : q(y) \leq p(g(y))$,
- iii) $\forall x_1, x_2 \in X : f(x_1) = f(x_2) \wedge p(x_1) = q(f(x_1)) \Rightarrow x_1 = x_2$,
- iv) $\forall y_1, y_2 \in Y : g(y_1) = g(y_2) \wedge q(y_1) = p(g(y_1)) \Rightarrow y_1 = y_2$.

Dann sind f und g zueinander invers und $p = q \circ f$ und $q = p \circ g$.

Beweis. Es genügt zu zeigen, dass $g(f(x)) = x$ für alle $x \in X$, d.h. $g \circ f = \text{id}_X$. Denn daraus würde folgen

$$p(x) \stackrel{i)}{\leq} q(f(x)) \stackrel{ii)}{\leq} p(g(f(x))) = p(x),$$

d.h. $p = q \circ f$. Völlig analog dazu kann $f \circ g = \text{id}_Y$ sowie $q = p \circ g$ gezeigt werden.

Angenommen, es gäbe ein $x_0 \in X$ derart, so dass für $y_0 = f(x_0)$ und $x_1 = g(y_0)$ gilt $x_1 \neq x_0$.

Sei außerdem $y_1 = f(x_1)$. Zwei Fälle können dabei unterschieden werden:

Falls $y_1 = y_0$, dann ist $f(x_1) = f(x_0)$ und die Ungleichungskette

$$p(x_1) \stackrel{i)}{\leq} q(f(x_1)) = q(y_1) = q(y_0) \stackrel{ii)}{\leq} p(g(y_0)) = p(x_1)$$

führt nach Anwendung von Eigenschaft iii) auf den Widerspruch $x_0 = x_1$.

Daher betrachten wir den Fall $y_1 \neq y_0$: Definiere induktiv zwei injektive (unendliche) Folgen $(x_i)_{i \geq 1}$ und $(y_i)_{i \geq 0}$ durch $y_i = f(x_i)$ und $x_{i+1} = g(y_i)$. Das widerspricht jedoch der Annahme, dass X und Y endliche Mengen sind. \square

Lemma 3.8. Seien (\vec{u}, \vec{B}) und (\vec{t}, \vec{C}) echte Faktorisierungsmuster. Seien $(v_n)_n$ und $(w_n)_n$ zwei Folgen von endlichen Wörtern mit

- $(v_n)_n$ ist (\vec{u}, \vec{B}) -adäquat,
- $(w_n)_n$ ist (\vec{t}, \vec{C}) -adäquat,
- $\forall n \in \mathbb{N} : v_n \sim_n w_n$.

Dann stimmen beide Faktorisierungsmuster überein: $(\vec{u}, \vec{B}) = (\vec{t}, \vec{C})$.

Beweis. Wir definieren allgemein für ein Faktorisierungsmuster (\vec{u}, \vec{B}) die Zahl

$$\|(\vec{u}, \vec{B})\| := \left(\sum_{i=0}^p |u_i| \right) + p.$$

Diese Zahl ist eine untere Schranke für die Länge jedes Wortes einer Folge, die (\vec{u}, \vec{B}) -adäquat ist. Benutzt wurde dabei, dass alle B_i nicht leer sind, was den Summanden p erklärt. Setze nun

$$k := \max\{\|(\vec{u}, \vec{B})\|, \|(\vec{t}, \vec{C})\|\}.$$

Von der Folge $(v_n)_n$ betrachten wir jetzt das Wort $v_1 = u_0 b_1 u_1 \dots b_p u_p$, wobei $b_i \in B_i^\circledast$, für alle i . In diesem Wort wollen wir alle Faktoren b_i mit k pumpen und definieren

$$v_1^{(k)} := u_0 b_1^k u_1 \dots b_p^k u_p.$$

Laut Definition ist $(v_n)_n$ eine (\vec{u}, \vec{B}) -adäquate Folge, wenn

$$\forall n \in \mathbb{N} : v_n \in u_0 (B_1^\circledast)^n u_1 \dots (B_p^\circledast)^n u_p.$$

Ein Faktor b_i^k von $v_1^{(k)}$ hat die Länge $k \cdot |b_i|$ und daher gilt die Teilwortbeziehung $b_i^k \leq x$ für beliebige Wörter $x \in (B_i^\circledast)^\ell$ mit $\ell \geq k \cdot |b_i|$. Daraus erhalten wir für jedes $\ell \geq k \cdot \max\{|b_1|, \dots, |b_p|\}$, dass $v_1^{(k)} \leq v_\ell$. Zusätzlich gilt für jedes $\ell' \geq \max\{\ell, |v_1^{(k)}|\}$, dass $v_1^{(k)} \in \text{Sub}_{\ell'}(v_{\ell'})$. Das sieht man so: Aus $\ell' \geq \ell$ folgt, dass $v_1^{(k)}$ ein Teilwort von $v_{\ell'}$ ist und wegen $\ell' \geq |v_1^{(k)}|$ hat $v_1^{(k)}$ höchstens die Länge ℓ' . Anwendung der Voraussetzung $v_n \sim_n w_n$ ($\forall n \in \mathbb{N}$) ergibt, dass auch $v_1^{(k)} \leq w_{\ell'}$ gilt.

Auf die gleiche Weise verfährt man mit der Folge $(w_n)_n$. Für das Wort $w_1 = t_0 c_1 t_1 \dots c_q t_q$

gibt es ein $m \geq k \cdot \max\{|c_1|, \dots, |c_q|\}$, so dass für alle $m' \geq \max\{m, |w_1^{(k)}|\}$, $w_1^{(k)} \leq w_{m'}$ und $w_1^{(k)} \leq v_{m'}$ gilt.

Wenn wir nun $M := \max\{\ell', m'\}$ setzen, erhalten wir die Teilwortbeziehungen

$$v_1^{(k)} \leq v_M, w_M \text{ und } w_1^{(k)} \leq v_M, w_M.$$

Betrachte einen Faktor b_i^k von $v_1^{(k)}$, der auch ein Teilwort von w_M ist. Aus den Aussagen $k \geq \|(\vec{t}, \vec{C})\|$ und $|b_i| > 0$ folgt mit dem Schubfachprinzip, dass es ein $C_j \subseteq A$ gibt mit $\text{alph}(b_i) \subseteq C_j$.

Die Idee ist es nun zwischen der Menge von indizierten Alphabeten in \vec{B} und der Menge von jenen in \vec{C} eine Bijektion herzustellen, die uns dabei hilft die Gleichheit der Faktorisierungsmuster zu zeigen.

Setze $\mathbf{B} := \{(B_1, 1), \dots, (B_p, p)\}$ und $\mathbf{C} := \{(C_1, 1), \dots, (C_q, q)\}$. Definiere eine Abbildung $f : \mathbf{B} \rightarrow \mathbf{C}$, wobei (B_i, i) auf dasjenige (C_j, j) abgebildet wird, so dass ein $c'_j \in (C_j^*)^M$ der erste Faktor von w_M ist, um b_i vollständig zu lesen, während $v_1^{(k)}$ als Teilwort von w_M gelesen wird. Die Abbildung $g : \mathbf{C} \rightarrow \mathbf{B}$ ist analog zu f definiert. f und g erhalten die Ordnung der Indizes und erhalten punktweise das Alphabet. Als nächstes soll gezeigt werden, dass \mathbf{B} und \mathbf{C} mit Hilfe von f und g bijektiv aufeinander abgebildet werden können. Dazu zeigen wir, dass $f = g^{-1}$, indem Lemma 3.7 angewandt wird.

Die Abbildungen f und g erfüllen die Bedingungen des Lemmas, indem wir setzen $X = \mathbf{B}$, $Y = \mathbf{C}$ und P als die Menge von Alphabeten mit der Mengeninklusion als Halbordnung. Ferner seien p und q die Projektionen auf die erste Komponente.

Die Voraussetzungen i) und ii) des Lemmas sind erfüllt, weil f und g punktweise das Alphabet erhalten. Angenommen, dass $(C_j, j) = f(B_{i_1}, i_1) = f(B_{i_2}, i_2)$ und $B_{i_1} = C_j$. Das bedeutet, dass ein Faktor b_{i_1} und ein Faktor b_{i_2} von $v_1^{(k)}$ gelesen werden innerhalb des gleichen Faktors c'_j von w_M . Deshalb ist $\text{alph}(b_{i_1} u_{i_1} \dots b_{i_2}) \subseteq \text{alph}(c'_j) = q(f(B_{i_1}, i_1)) = B_{i_1} = \text{alph}(b_{i_1})$. Aber die Annahme war, dass (\vec{u}, \vec{B}) ein echtes Faktorisierungsmuster ist, also muss i_1 gleich i_2 sein. Dies zeigt, dass Bedingung iii) des Lemmas gilt. Bedingung iv) kann ähnlich gezeigt werden.

Wir erhalten als Implikation, dass f und g bijektive Abbildungen zwischen \mathbf{B} und \mathbf{C} definieren, und daher ist $p = q$ und $B_i = C_i$ für jedes i . Da wir es mit echten Faktorisierungsmustern zu tun haben, impliziert $v_1^{(k)} \leq w_M$ jetzt, dass $t_i \leq u_i$ für jedes i . Genauso impliziert $w_1^{(k)} \leq v_M$, dass $u_i \leq t_i$ für jedes i . Folglich haben wir dank der Antisymmetrie von \leq (für endliche Wörter) die Gleichheit $u_i = t_i$ für jedes i und das ergibt die gewünschte Gleichheit der Faktorisierungsmuster. \square

Die Hinrichtung von Proposition 3.2 wird jetzt bewiesen.

Lemma 3.9. *Gegeben seien zwei DFAs \mathcal{A}_1 und \mathcal{A}_2 . Wenn die von den Automaten erkannten Sprachen $L = L(\mathcal{A}_1)$ und $K = L(\mathcal{A}_2)$ nicht PT-separierbar sind, dann lassen beide DFAs einen gemeinsamen (\vec{u}, \vec{B}) -Pfad zu.*

Beweis. Nach Annahme, gibt es für jedes $n \in \mathbb{N}$ ein $v_n \in L$ und ein $w_n \in K$ mit der Eigenschaft

$$v_n \sim_n w_n.$$

Dies definiert eine unendliche Folge von Wortpaaren $(v_n, w_n)_n$. Wir wollen nun in beiden Folgen v_n und w_n adäquate Teilfolgen finden.

Nach Lemma 3.6, kann man aus $(v_n, w_n)_n$ eine Teilfolge extrahieren, deren erste Komponente eine adäquate Folge bildet. Aus dieser Teilfolge von Paaren, erneut unter Verwendung von Lemma 3.6, extrahieren wir eine Teilfolge, deren zweite Komponente auch adäquat ist. Die erste Komponente bleibt dabei adäquat (ausgenutzt wird dabei, dass die Eigenschaft einer Folge adäquat zu sein, erhalten bleibt, wenn eine beliebige Teilfolge daraus ausgewählt wird, weil $(B^\otimes)^n \subseteq (B^\otimes)^m$ für $n \geq m$). Daher kann man annehmen, dass $(v_n)_n$ adäquat zu $(w_n)_n$ ist.

Lemma 3.8 zeigt, dass man das gleiche Faktorisierungsmuster (\vec{u}, \vec{B}) wählen kann, so dass $(v_n)_n$ und $(w_n)_n$ (\vec{u}, \vec{B}) -adäquat ist. Schließlich erhalten wir mit der nächsten Behauptung 3.10, dass \mathcal{A}_1 und \mathcal{A}_2 einen (\vec{u}, \vec{B}) -Pfad zulassen. \square

Behauptung 3.10. Wenn $L(\mathcal{A})$ eine (\vec{u}, \vec{B}) -adäquate Folge enthält, dann lässt \mathcal{A} einen (\vec{u}, \vec{B}) -Pfad zu.

Beweis. $L(\mathcal{A})$ enthält eine (\vec{u}, \vec{B}) -adäquate Folge $(v_n)_n$, d.h.

$$\forall n \geq 0 : v_n \in u_0(B_1^\otimes)^n u_1 \dots (B_p^\otimes)^n u_p \cap L(\mathcal{A}).$$

Sei v_n ein ausreichend großer Term in dieser Folge, z.B. mit $n > |Q(\mathcal{A})|$. Jetzt muss der Pfad, der benutzt wird, um v_n zu lesen, alle B_i^\otimes -Schleifen traversieren und nach der Form von v_n ist dies ein (\vec{u}, \vec{B}) -Pfad. \square

4 Separierbarkeit durch eine piecewise testable Sprache über unendlichen Wörtern

Theorem 4.1. Gegeben seien zwei Büchi-Automaten. Dann ist in Polynomialzeit entscheidbar, bzgl. der Anzahl der Zustände und der Größe des Alphabets, ob die Sprachen, die durch die Büchi-Automaten akzeptiert werden, PT-separierbar sind.

Aus den folgenden zwei Propositionen folgt Theorem 4.1.

Proposition 4.2. Seien \mathcal{A}_1 und \mathcal{A}_2 zwei Büchi-Automaten. Dann sind $L(\mathcal{A}_1)$ und $L(\mathcal{A}_2)$ genau dann nicht PT-separierbar, wenn ein ω -Faktorisierungsmuster (\vec{u}, \vec{B}) existiert, so dass \mathcal{A}_1 und \mathcal{A}_2 einen gemeinsamen ω -(\vec{u}, \vec{B})-Pfad zulassen.

Proposition 4.3. Gegeben seien zwei Büchi-Automaten \mathcal{A}_1 und \mathcal{A}_2 . Dann kann in Polynomialzeit entschieden werden, bzgl. der Anzahl der Zustände und der Größe des Alphabets, ob ein ω -Faktorisierungsmuster (\vec{u}, \vec{B}) existiert, so dass beide Büchi-Automaten einen gemeinsamen ω -(\vec{u}, \vec{B})-Pfad zulassen.

4.1 Ein Algorithmus in Polynomialzeit

Beweis. (Proposition 4.3) Zunächst zeigen wir, dass das folgende Problem in PTIME (bzgl. $|A|, |\mathcal{A}_1|, |\mathcal{A}_2|$) entscheidbar ist.

Gegeben: Zustände $p_1, q_1, r_1 \in Q_1$ in \mathcal{A}_1 und $p_2, q_2, r_2 \in Q_2$ in \mathcal{A}_2 .

Frage: Existiert ein nichtleeres Teialphabet $B \subseteq A$, so dass beide Automaten ein gemeinsames Muster der folgenden Form enthalten?

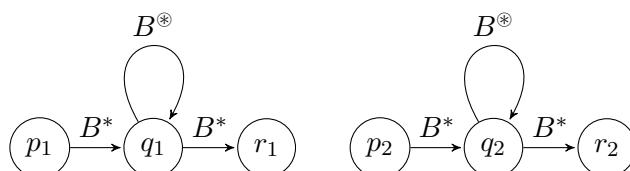


Abbildung 4.1: Ein gemeinsames Muster in \mathcal{A}_1 und \mathcal{A}_2

Dazu wird eine Folge von Alphabeten C_i abnehmender Größe berechnet, die das maximale Teialphabet B , mit dem die Schleifen beschriftet sind, von oben approximieren sollen. Angenommen, es existiert ein solch gefordertes B , dann muss B in $C_1 := \text{alph_scc}(q_1, \mathcal{A}_1) \cap \text{alph_scc}(q_2, \mathcal{A}_2)$

enthalten sein. Indem z.B. der Algorithmus von Tarjan, der starke Zusammenhangskomponenten in Linearzeit berechnet, verwendet wird, kann C_1 ebenfalls in Linearzeit berechnet werden. Anschließend schränken wir beide Automaten auf das Alphabet C_1 ein und iterieren den Prozess, und erhalten so eine Folge $(C_i)_i$ nach der Vorschrift:

$$C_{i+1} := \text{alph_scc}(q_1, \mathcal{A}_1|_{C_i}) \cap \text{alph_scc}(q_2, \mathcal{A}_2|_{C_i}).$$

Nach einer endlichen Anzahl von $n \leq |A|$ Schritten ist die Folge stabil, d.h. es gilt $C_n = C_{n+1}$. Falls $C_n = \emptyset$, dann gibt es kein nichtleeres Teilalphabet $B \subseteq A$, so dass es eine B^* -Schleife um q_1 und q_2 gibt.

Falls $C_n \neq \emptyset$, dann ist $B = C_n$ das maximale nichtleere Alphabet B , so dass es B^* -Schleifen um q_1 in \mathcal{A}_1 und q_2 in \mathcal{A}_2 gibt. Es verbleibt zu zeigen, dass es Pfade $p_1 \xrightarrow{B^*} q_1 \xrightarrow{B^*} r_1$ und $p_2 \xrightarrow{B^*} q_2 \xrightarrow{B^*} r_2$ gibt. Dies kann jedoch z.B. mittels Tiefensuche in linearer Zeit durchgeführt werden.

Zusammengefasst erhalten wir für die Laufzeit: Die Anzahl der Iterationen n bis $C_n = C_{n+1}$ erreicht wird, ist beschränkt durch $|A|$ und jede einzelne Berechnung von C_i kann in Linearzeit bzgl. der Größe von \mathcal{A}_1 und \mathcal{A}_2 durchgeführt werden. Das impliziert, dass in Polynomialzeit (bzgl. $|A|$ und der Größe der Büchi-Automaten) entschieden werden kann, ob man ein gemeinsames Muster wie in Abbildung 4.1 in \mathcal{A}_1 und in \mathcal{A}_2 vorfinden kann.

Nun bauen wir aus \mathcal{A}_1 und \mathcal{A}_2 zwei modifizierte Büchi-Automaten $\tilde{\mathcal{A}}_1$ und $\tilde{\mathcal{A}}_2$ wie folgt. Zuerst initialisieren wir $\tilde{\mathcal{A}}_i$ als eine Kopie von \mathcal{A}_i . Für jedes 4-Tupel $\tau = (p_1, r_1, p_2, r_2) \in Q_1^2 \times Q_2^2$ für das es ein Alphabet $B \subseteq A$, Zustände $q_1 \in Q_1, q_2 \in Q_2$ und Pfade $p_i \xrightarrow{B^*} q_i \xrightarrow{B^*} q_i \xrightarrow{B^*} r_i$ (wie in Abbildung 3.1) gibt, fügen wir in beide Automaten einen neuen Buchstaben a_τ zum Alphabet hinzu. Ferner fügen wir Abkürzungen in Form von Übergängen $p_1 \xrightarrow{a_\tau} r_1$ und $p_2 \xrightarrow{a_\tau} r_2$ in beide Automaten hinzu. Die Zahl von solchen Tupeln $(p_1, q_1, r_1, p_2, q_2, r_2)$ beträgt höchstens $|Q_1|^3 \cdot |Q_2|^3$ und mit dem oben gezeigten Teil erhalten wir, dass die Berechnung dieser neuen Übergänge in Polynomialzeit durchgeführt werden kann. Weiter folgt, dass die Berechnung der modifizierten Automaten $\tilde{\mathcal{A}}_1$ und $\tilde{\mathcal{A}}_2$ in PTIME möglich ist.

Nach Konstruktion dieser Automaten gibt es ein ω -Faktorisierungsmuster (\vec{u}, \vec{B}) , so dass $\mathcal{A}_1, \mathcal{A}_2$ genau dann einen gemeinsamen ω -(\vec{u}, \vec{B})-Pfad zulassen, wenn $L(\tilde{\mathcal{A}}_1) \cap L(\tilde{\mathcal{A}}_2) \neq \emptyset$. Dies kann tatsächlich in Polynomialzeit entschieden werden, weil $\tilde{\mathcal{A}}_1$ und $\tilde{\mathcal{A}}_2$ in PTIME gebaut wurden. \square

4.2 Charakterisierung der PT-Separierbarkeit

Als nächstes soll Proposition 4.2 bewiesen werden. Wir beginnen mit der Rückrichtung. Die Hinrichtung wird in Lemma 4.7 bewiesen.

Lemma 4.4. *Seien $\mathcal{A}_1, \mathcal{A}_2$ Büchi-Automaten, die einen gemeinsamen ω -(\vec{u}, \vec{B})-Pfad zulassen. Dann sind die Sprachen $L = L(\mathcal{A}_1)$ und $K = L(\mathcal{A}_2)$ nicht PT-separierbar.*

Beweis. Wir nehmen an, $S \subseteq A^\omega$ ist eine piecewise testable Sprache mit $L \subseteq S$. Nach Bemerkung 2.2 gibt es passende $s, x_i, y_i, z_i \in A^*, \alpha \in A^\omega$, so dass L eine Sprache der Form

$$\tilde{L} := u_0(x_1 y_1^* z_1) u_1 \dots (x_p y_p^* z_p) u_p s \alpha$$

enthält, wobei $\text{alph}(x_i) \cup \text{alph}(z_i) \subseteq \text{alph}(y_i) = B_i$, für alle $i \in \{1, \dots, p\}$, $\text{alph}(s) \subseteq \text{alph}(\alpha)$ und $\alpha \in (B_{p+1}^*)^\omega$. Entsprechend enthält K mit geeigneten $s', x'_i, y'_i, z'_i \in A^*$, $\beta \in A^\omega$ eine Sprache der Form

$$\widetilde{K} := u_0(x'_1 y'^*_1 z'_1) u_1 \dots (x'_p y'^*_p z'_p) u_p s' \beta,$$

mit $\text{alph}(x'_i) \cup \text{alph}(z'_i) \subseteq \text{alph}(y'_i) = B_i$ für alle i , $\text{alph}(s') \subseteq \text{alph}(\beta)$ und $\beta \in (B_{p+1}^*)^\omega$. Das Ziel ist zu zeigen, dass es zu jedem $n \in \mathbb{N}$ ein $w_1 \in \widetilde{L}$ sowie ein $w_2 \in \widetilde{K}$ gibt, so dass die Wörter w_1 und w_2 zueinander \sim_n -äquivalent sind. Durch Anwendung von Behauptung 3.4 erhält man, dass für alle $1 \leq i \leq p$ und $n \in \mathbb{N}$ gilt

$$x_i y_i^n z_i \sim_n x'_i y'^n_i z'_i.$$

Da die Äquivalenz \sim_n zudem eine Kongruenz ($\alpha_1 \sim_n \beta_1 \wedge \alpha_2 \sim_n \beta_2 \Rightarrow \alpha_1 \alpha_2 \sim_n \beta_1 \beta_2$) ist, erhalten wir für alle $n \in \mathbb{N}$

$$u_0(x_1 y_1^n z_1) u_1 \dots (x_p y_p^n z_p) u_p s \alpha \sim_n u_0(x'_1 y'^n_1 z'_1) u_1 \dots (x'_p y'^n_p z'_p) u_p s' \beta.$$

Die linke Seite dieser Äquivalenz stellt unser gewünschtes Wort $w_1 \in \widetilde{L}$ und die rechte Seite unser gewünschtes Wort $w_2 \in \widetilde{K}$ dar, die beide zueinander \sim_n -äquivalent sind. Folglich liegen w_1 und w_2 in derselben Äquivalenzklasse. Die piecewise testable Sprache S bildet eine Vereinigung aus \sim_n -Äquivalenzklassen und daher sind S und K nicht disjunkt. Das wiederum bedeutet, dass L und K nicht separierbar sind. \square

Wir fahren mit der Hinrichtung von Proposition 4.2 fort.

Lemma 4.5. *Jede Folge $(\beta_n)_n$ von unendlichen Wörtern $\beta_n \in A^\omega$ besitzt eine ω -adäquate Teilfolge.*

Beweis. Die Folge $(\beta_n)_n$ zerlegen wir in zwei Folgen $(u_n)_n$ und $(\alpha_n)_n$ mit

$$\beta_n = u_n \alpha_n,$$

so dass für alle $n \in \mathbb{N}$ jeweils gilt, $u_n \in A^*$, $\alpha_n \in A^\omega$ und zusätzlich $\inf(\alpha_n) = \text{alph}(\alpha_n)$, d.h. jedes in α_n vorkommende Zeichen muss unendlich oft darin vorkommen. Indem Lemma 3.6 angewandt wird, kann aus $u_n \alpha_n$ eine Teilfolge $u_{\sigma(n)} \alpha_{\sigma(n)}$ gewonnen werden, so dass $(u_{\sigma(n)})_n$ adäquat ist. Dank diesem Argument dürfen wir ohne Einschränkung bereits annehmen, dass $(u_n)_n$ adäquat ist.

Als nächstes soll gezeigt werden, dass es eine Teilfolge $(\alpha_{\sigma(n)})_n$ und ein nichtleeres Teialphabet $B_{p+1} \subseteq A$ gibt, so dass $\alpha_{\sigma(n)} \in (B_{p+1}^*)^\omega$ für alle $n \in \mathbb{N}$.

Wir betrachten die Alphabete $\text{alph}(\alpha_n)$ als Teilmengen von A . Die Anzahl aller Teilmengen von A beträgt $2^{|A|}$ und ist daher endlich. Da es sich bei der Folge α_n insbesondere um eine unendliche Folge handelt, muss es ein nichtleeres Teialphabet $B_{p+1} \subseteq A$ geben, so dass B_{p+1} das Alphabet von unendlich vielen α_n ist. Daraus schließt man, dass es eine Teilfolge $(\alpha_{\sigma(n)})_n$ gibt, so dass $\text{alph}(\alpha_{\sigma(n)}) = B_{p+1}$ für alle $n \in \mathbb{N}$ ist. Jedes Zeichen aus B_{p+1} kommt aber unendlich oft in $\alpha_{\sigma(n)}$ vor. Das erlaubt es uns, $\alpha_{\sigma(n)}$ in der Form $b_1 b_2 b_3 \dots$ mit $b_i \in B_{p+1}$ zu schreiben. Z.B. können wir annehmen, dass b_1 der kleinste Präfix von $\alpha_{\sigma(n)}$ ist mit $b_1 \in B_{p+1}$. Sei $\alpha_{\sigma(n)} = b_1 \gamma$ für ein passendes $\gamma \in A^\omega$. Analog ist b_2 der kleinste Präfix von γ , mit $b_2 \in B_{p+1}$ usw. Das impliziert, dass $\alpha_{\sigma(n)} \in (B_{p+1}^*)^\omega$ für alle $n \in \mathbb{N}$. Zusammengefasst folgt daraus die Aussage des Lemmas:

- $(u_n)_n$ ist adäquat, d.h. $\forall n \in \mathbb{N} : u_n \in u_0(B_1^*)^n u_1 \dots (B_p^*)^n u_p$ und
- $\forall n \in \mathbb{N} : \alpha_n \in (B_{p+1}^*)^\omega$

impliziert

$$\forall n \in \mathbb{N} : \beta_n = u_n \alpha_n \in u_0(B_1^*)^n u_1 \dots (B_p^*)^n u_p (B_{p+1}^*)^\omega,$$

d.h. $(\beta_n)_n$ bzw. eine Teilfolge ist ω -adäquat. \square

Lemma 4.6. Seien (\vec{u}, \vec{B}) und (\vec{t}, \vec{C}) echte ω -Faktorisierungsmuster. Seien $(v_n)_n$ und $(w_n)_n$ zwei Folgen von unendlichen Wörtern mit

- $(v_n)_n$ ist ω -(\vec{u}, \vec{B})-adäquat,
- $(w_n)_n$ ist ω -(\vec{t}, \vec{C})-adäquat,
- $\forall n \in \mathbb{N} : v_n \sim_n w_n$.

Dann stimmen beide ω -Faktorisierungsmuster überein: $(\vec{u}, \vec{B}) = (\vec{t}, \vec{C})$.

Beweis. Indem Lemma 3.8 für die entsprechenden induzierten Faktorisierungsmuster von (\vec{u}, \vec{B}) und (\vec{t}, \vec{C}) angewandt wird, folgt sofort: $u_i = t_i$ für $i = 0, \dots, p$ und $B_i = C_i$ für $i = 1, \dots, p$. Bleibt zu zeigen, dass $B_{p+1} = C_{p+1}$.

Beide Folgen enthalten nur unendliche Wörter aus A^ω und aus der Bedingung $v_n \sim_n w_n$, für alle n folgt, dass $\inf(v_n) = \inf(w_n)$, d.h. in v_n und in w_n kommen die selben Buchstaben unendlich oft vor. Das kann daher nur sein, wenn $B_{p+1} = C_{p+1}$. \square

Jetzt kann die Hinrichtung von Proposition 4.2 bewiesen werden.

Lemma 4.7. Gegeben seien zwei Büchi-Automaten \mathcal{A}_1 und \mathcal{A}_2 . Wenn die von den Automaten erkannten Sprachen $L = L(\mathcal{A}_1)$ und $K = L(\mathcal{A}_2)$ nicht PT-separierbar sind, dann lassen beide Büchi-Automaten einen gemeinsamen ω -(\vec{u}, \vec{B})-Pfad zu.

Beweis. Nach Annahme, gibt es für jedes $n \in \mathbb{N}$ ein $v_n \in L$ und ein $w_n \in K$ mit der Eigenschaft

$$v_n \sim_n w_n.$$

Dies definiert eine unendliche Folge von Wortpaaren $(v_n, w_n)_n$. Wir wollen nun in beiden Folgen $(v_n)_n$ und $(w_n)_n$ ω -adäquate Teilstreichen finden.

Nach Lemma 4.5, kann man aus $(v_n, w_n)_n$ eine Teilstreiche extrahieren, deren erste Komponente eine ω -adäquate Folge bildet. Aus dieser Teilstreiche von Paaren (mit Lemma 4.5) extrahieren wir eine Teilstreiche, deren zweite Komponente auch ω -adäquat ist. Die erste Komponente bleibt dabei ω -adäquat (ausgenutzt wird dabei, dass die Eigenschaft einer Folge ω -adäquat zu sein, erhalten bleibt, wenn eine beliebige Teilstreiche daraus ausgewählt wird, weil $(B^*)^n \subseteq (B^*)^m$ für $n \geq m$). Daher kann man annehmen, dass $(v_n)_n$ und $(w_n)_n$ ω -adäquat ist. Lemma 4.6 zeigt, dass man das gleiche ω -Faktorisierungsmuster (\vec{u}, \vec{B}) wählen kann, so dass $(v_n)_n$ und $(w_n)_n$ ω -(\vec{u}, \vec{B})-adäquat ist. Schließlich erhalten wir mit der nächsten Behauptung 4.8, dass \mathcal{A}_1 und \mathcal{A}_2 einen ω -(\vec{u}, \vec{B})-Pfad zulassen. \square

Behauptung 4.8. Wenn $L(\mathcal{A})$ eine ω -(\vec{u}, \vec{B})-adäquate Folge enthält, dann lässt \mathcal{A} einen ω -(\vec{u}, \vec{B})-Pfad zu.

Beweis. $L(\mathcal{A})$ enthalte eine ω -(\vec{u}, \vec{B})-adäquate Folge $(v_n)_n$, d.h.

$$\forall n \in \mathbb{N} : v_n \in u_0(B_1^*)^n u_1 \dots (B_p^*)^n u_p (B_{p+1}^*)^\omega \cap L(\mathcal{A}).$$

Sei v_n ein ausreichend großer Term in dieser Folge, z.B. mit $n > |Q|$. Jetzt muss der Pfad, der verwendet wird, um v_n zu lesen, alle B_i^* -Schleifen traversieren und nach der Form von v_n ist dies ein ω -(\vec{u}, \vec{B})-Pfad. \square

5 Zusammenfassung

Gezeigt wurde in Kapitel 4, dass das Separierbarkeitsproblem für reguläre Sprachen durch eine piecewise testable Sprache über unendlichen Wörtern in PTIME liegt. Meines Wissens stellt dies ein neues Resultat dar, das in dieser Form noch nicht veröffentlicht wurde. Insgesamt konnten dieselben Techniken wie im ursprünglichen Beweis von Rooijen und Zeitoun angewandt werden. Lediglich kleine Änderungen in den verwendeten Begriffen und Anpassungen in den Beweisen waren notwendig, um die neue Situation anzugehen. Die Laufzeit vom Entscheidungsalgorithmus ist beim Übergang von NFAs zu Büchi-Automaten polynomiell geblieben, d.h. es gab keine Verschlechterungen bzgl. der Komplexität.

Für die Charakterisierbarkeit floss das Faktorisierungswaldtheorem für endliche Wörter im Zusammenhang mit ω -adäquaten Teilfolgen bei Lemma 4.5 an entscheidender Stelle ein, indem Lemma 3.6 für Folgen von endlichen Wörtern angewandt wurde. Ich habe dazu für Lemma 4.5 gezeigt, dass man für den ω -Teil für alle Worte einer Folge von unendlichen Wörtern ein Teilalphabet findet, so dass der ω -Teil aus unendlich vielen Faktoren besteht, deren Alphabete alle gleich diesem Teilalphabet sind.

Die Bedingung aus Lemma 3.8, wann zwei endliche Folgen ein gemeinsames Faktorisierungsmuster teilen, konnte auf den unendlichen Fall in Lemma 4.6 übertragen werden. Dabei hat das technische Lemma 3.7 von Almeida geholfen, die Gleichheit der Faktorisierungsmuster über eine bijektive Abbildung herzuleiten. Außerdem habe ich in Lemma 4.6 gezeigt, dass aus der \sim_n -Äquivalenz-Bedingung die Gleichheit der Teilalphabete in den ω -Faktorisierungsmustern, die für den ω -Teil zuständig sind, gefolgert werden kann.

In zukünftigen Arbeiten könnte untersucht werden, ob es möglich ist, das Separierbarkeitsproblem simultan über endliche und unendliche Wörter entscheiden zu können.

Literaturverzeichnis

- [Alm94] J. Almeida. *Finite semigroups and universal algebra*. World Scientific, 1994.
- [Alm96] J. Almeida. Some Algorithmic Problems for Pseudovarieties. Technischer Bericht, Publ. Math. Debrecen, 1996.
- [AZGJ97] J. Almeida, M. Zeitoun, P. Gomes, T. Jussieu. The pseudovariety J is hyperdecidable. Technischer Bericht, Theoretical Informatics and Applications 31, 1997.
- [DK11] V. Diekert, M. Kufleitner. Fragments of first-order logic over infinite words. *Theory of Computing Systems*, 48(3):486–516, 2011.
- [Kuf08] M. Kufleitner. The height of factorization forests. In *Mathematical Foundations of Computer Science 2008*, S. 443–454. Springer, 2008.
- [RZ13] L. van Rooijen, M. Zeitoun. The separation problem for regular languages by piecewise testable languages. *CoRR*, abs/1303.2143, 2013. URL <http://arxiv.org/abs/1303.2143>.
- [Sim90] I. Simon. Factorization forests of finite height. *Theoretical Computer Science*, 72(1):65–94, 1990.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

Ort, Datum, Unterschrift