Über die Operation Fortsetzung bei formalen Sprachen

Robert Hartmann

24. September 2010

Inhaltsverzeichnis

1 Einleitung

In dieser Arbeit untersuchen wir die Operation Fortsetzung für formalen Sprachen. Diese Operation wird in der Arbeit [St87] eingeführt. Dabei bezeichnen wir die Fortsetzung eines Wortes w in eine Sprache L, als das Minimum aller Wörter aus L, in denen w ein Präfix ist.
Man stelle sich einen Ableitungsbaum vor beginnend bei w. Man folgt nun allen Pfaden von w nach X^*. Trifft man auf einem Pfad auf ein Wort aus L, so wird dieses Wort dem Ergebnis hinzugefügt und diesem Pfad wird nicht mehr gefolgt.
Wir definieren die Fortsetzung einer Sprache L in eine Sprache W als Vereinigung der Fortsetzungen aller Wörter aus L in W.

Wie in der Arbeit [St87] behandelt, erleichtert die Operation Fortsetzung den Schnitt beim δ-Limes zweier Sprachen, mehr dazu im zweiten Abschnitt dieser Arbeit.

In dieser Arbeit wird deshalb die Operation Fortsetzung untersucht. Zunächst legen wir die verwendete Notation fest. Dann betrachten wir die algebraische Struktur $(2^{X^*}, \triangleright)$ und untersuchen diese auf typische Eigenschaften. Da diese Struktur nicht kommutativ ist, untersuchen wir dann die Stabilität der Operation Fortsetzung in Bezug auf die mengentheoretischen Operationen $\cap, \cup$ und der Konkatenation im Vorder- sowie Hinterglied. Im dann folgenden Abschnitt untersuchen wir die Operation Fortsetzung, wenn eine der beiden Operanden die spezielle Form $\mathit{pref}(L)$ oder $W \cdot X^*$ hat. Abschließend betrachten wir die Abgeschlossenheitseigenschaften der Operation Fortsetzung für die Klassen der CHOMSKY-Hierachie.

2 Notation

Mit $\mathbb{N} = \{0, 1, 2, 3, ..\}$ bezeichnen wir die Menge der natürlichen Zahlen. Es sei X ein Alphabet. Mit X^* bezeichnen wir die Menge aller endlichen Wörter über dem Alphabet X, einschließlich des leeren Wortes e. Wir bezeichnen weiterhin die Menge X^ω als Menge aller unendlichen Wörter über dem Alphabet X.

Für $w \in X^*$ und $v \in X^* \cup X^\omega$ bezeichne $w \cdot v$ die Konkatenation von w und v. Daraus ergibt sich in natürlicher Weise ein Produkt $L \cdot W$ von Mengen $L \subseteq X^*$ und $W \subseteq X^* \cup X^\omega$. Für eine Sprache W ist $W^* = \bigcup_{i \in \mathbb{N}} W^i$. Weiterhin bezeichne $|w|$ die Länge des Wortes $w \in X^*$. Wir definieren die Präfixrelation $\sqsubseteq$ wie üblich mit $w \sqsubseteq b \Leftrightarrow w \cdot b' = b,\ \textit{für ein}\ b' \in X^*$. Somit bildet $\textit{pref}(L) = \{v : v \sqsubseteq w \wedge w \in L\}$ alle Präfixe aller Wörter aus L.

Weiterhin bezeichnen wir mit $\textit{Min}\ (L) = \{w : w \in L \wedge \forall v(v \sqsubset w \to v \notin L)\}$ alle Wörter der Sprache L, in denen kein Wort Präfix eines anderen Wortes ist. Die Fortsetzung $\triangleright$ eines Wortes w in L definieren wir als $w \triangleright L = \textit{Min}\ (w \cdot X^* \cap L)$. Diese Definition weiten wir auf Sprachen aus und definieren die Fortsetzung einer Sprache L in die Sprache W als $L \triangleright W = \bigcup_{u \in L} u \triangleright W$

Wir definieren den δ-Limes einer Wortmenge W^δ wie in [St87] mit
$W^\delta = \{\beta : \beta \in X^\omega$ und $\textit{pref}(\beta) \cap W$ ist unendlich$\}$ Die Eigenschaft (13) aus [St87] ist leicht einzusehen: $(U \cup W)^\delta = U^\delta \cup W^\delta$. Eine Sprache nennen wir eine (σ, δ)-Teilmenge von X^* genau dann, wenn für alle $\beta \in X^\omega$ entweder $\textit{pref}(\beta) \cap W$ oder $\textit{pref}(\beta) \setminus W$ endlich ist. Beispiele für (σ, δ)-Teilmengen sind alle endlichen Sprachen und deren Komplemente. Weitere Beispiele sind Sprachen der Form $\textit{pref}(L)$ oder $W \cdot X^*$.
Für den Durchschnitt gilt nur $(U \cap W)^\delta = U^\delta \cap W^\delta$, falls einer der beiden Operanden eine (σ, δ)-Teilmenge ist. Die Operation der Fortsetzung $\triangleright$ wurde in [St87] eingefuehrt, um den Durchschnitt zweier δ-Limites zu beschreiben.

3 Eigenschaften

In diesem Abschnitt betrachten wir Eigenschaften der Operation Fortsetzung für formale Sprachen. Hierbei wird zunächst die algebraische Struktur $(2^{X^*}, \triangleright)$ untersucht. Anschliessend untersuchen wir die Monotonie der Operation Fortsetzung bezüglich $\subseteq$ und die Stabilität bezüglich der mengentheoretischen Operationen $\cap$ und $\cup$.

3.1 Eigenschaften der algebraischen Struktur $(2^{X^*}, \triangleright)$

3.1.1 Kommutativität und Assoziativität

Die algebraische Struktur $A = (2^{X^*}, \triangleright)$ ist weder kommutativ, noch assoziativ. Um dies zu zeigen, wird ein Gegenbeispiel angegeben.

Beispiel 1. Es seien $L = \{aa, bb\}$, $V = \{aa, b\}$ und $W = \{aa, bb\}$.
Dann haben wir $(L \triangleright V) \triangleright W = \{aa\} \triangleright W = \{aa\}$, aber $L \triangleright (V \triangleright W) = L \triangleright \{aa, bb\} = \{aa, bb\}$ und $L \triangleright V = \{aa\}$, aber $V \triangleright L = \{aa, bb\}$.

3.1.2 Einselement

Da die algebraische Struktur A nicht kommutativ ist, werden bei der Untersuchung auf Einselemente sowohl linksneutrale, als auch rechtsneutrale Einselemente betrachtet. Wir betrachten zuerst linksneutrale Elemente. Man erkennt leicht, dass X^* ein solches Element ist, denn es gilt $X^* \triangleright L = L$. Analog ist X^* rechtsneutrales Element, denn es gilt $L \triangleright X^* = L$. Nehmen wir nun an, es gebe weitere links- oder rechtsneutrale Elemente E_l bzw. E_r, so haben wir $E_l = E_l \triangleright X^* = X^*$, da X^* rechtsneutrales Element ist, sowie $E_r = X^* \triangleright E_r = X^*$, da X^* auch linksneutrales Element ist.

3.1.3 Nullelement

Wir betrachten, da A nicht kommutativ ist, ... prosa. Wir betrachten zuerst linksabsorbierende Elemente. Man erkennt leicht, dass $\emptyset$ ein solches Element ist, denn es gilt $\emptyset \triangleright L = \emptyset$. Analog ist $\emptyset$ rechtsabsorbierendes Element, denn es gilt $L \triangleright \emptyset = \emptyset$. Nehmen wir an, es gebe weitere links- oder rechtsabsorbierende Elemente N_l bzw. N_r, so haben wir $N_l = N_l \triangleright \emptyset = \emptyset$, da $\emptyset$ rechtsabsorbierendes Element ist, sowie $N_r = \emptyset \triangleright N_r = \emptyset$, da $\emptyset$ auch linksabsorbierendes Element ist.

Damit kann die algebraische Struktur $A = (2^{X^*}, \triangleright)$ aufgrund fehlender Assoziativität weder eine Gruppe noch eine Halbgruppe sein, besitzt aber ein Null- und ein Einselement.

3.2 BOOLEsche Operationen

In diesem Abschnitt untersuchen wir die Operation Fortsetzung auf Stabilität bezüglich den mengentheoretischen Operationen $\cap$ und $\cup$ und auf Monotonie bezüglich $\subseteq$.

Aus der Definition folgt direkt die Eigenschaft:

Eigenschaft 1. *Es gilt genau dann* $\{w\} \triangleright W = \{w\}$, *wenn* $w \in W$.

Aus Eigenschaft 1 folgt sofort

Folgerung 1. $L \cap W \triangleright W = L \cap W$

und damit erhalten wir

Folgerung 2. $L \cap W \subseteq L \triangleright W \subseteq W$

Da die Operation Fortsetzung einer Sprache L in eine Sprache W über die Vereinigung $\cup$ aller Wörter aus L definiert wurde, folgt aus der Definition:

Eigenschaft 2. $(\bigcup_{i\in\mathbb{N}} L_i) \triangleright W = \bigcup_{i\in\mathbb{N}}(L_i \triangleright W)$

Beweis. Nach Definition gilt $(\bigcup_{i\in\mathbb{N}} L_i) \triangleright W = \bigcup_{w\in L_0\cup L_1\cup\ldots\cup L_n} w \triangleright W$
Wir betrachten nun die Vereinigung für jede Sprache, so dass $\bigcup_{w\in L_0\cup L_1\cup\ldots\cup L_n} w \triangleright W = (\bigcup_{w\in L_0} w \triangleright W) \cup (\bigcup_{w\in L_1} w \triangleright W) \cup \ldots \cup (\bigcup_{w\in L_n} w \triangleright W)$. Damit haben wir $(\bigcup_{i\in\mathbb{N}} L_i) \triangleright W = \bigcup_{i\in\mathbb{N}}(L_i \triangleright W)$ □

Aus Folgerung 2 wissen wir, dass $L \cap W \subseteq L \triangleright W$. Wir zerlegen nun die Fortsetzung von L in W so, dass sich folgende Beziehung ergibt:

Eigenschaft 3. $L \triangleright W = (L \cap W) \cup ((L \setminus W) \triangleright W)$

Beweis. Wir zerlegen zunächst L in $L \cap W$ und $L \setminus W$. Damit haben wir (1). Nach Anwenden der Eigenschaft 2 erhalten wir (2). Mit Eigenschaft 1 lässt sich der Ausdruck zu (3) vereinfachen.

$$L \triangleright W = ((L \cap W) \cup (L \setminus W)) \triangleright W \tag{1}$$
$$L \triangleright W = ((L \cap W) \triangleright W) \cup ((L \setminus W) \triangleright W) \tag{2}$$
$$L \triangleright W = (L \cap W) \cup ((L \setminus W) \triangleright W) \tag{3}$$

□

Wir bilden nun Inklusionsbeziehungen zwischen L und W und erhalten mit Eigenschaft 3, zwei Folgerungen. Zu beachten ist, dass in Folgerung 3 eine „genau dann wenn“ Beziehung gilt.

Folgerung 3. $L \subseteq W \leftrightarrow L \triangleright W = L \cap W = L$

Folgerung 4. $L \supseteq W \rightarrow L \triangleright W = W$

Für Folgerung 4 gilt nicht die Richtung $\leftarrow$, wie das folgende Beispiel zeigt:

Beispiel 2. Es seien $L = \{b\}, W = \{bb\}$. Dann haben wir $L \triangleright W = W$, aber $L \not\supseteq W$.

Da die Operation Fortsetzung nicht kommutativ ist, betrachten wir nun die Monotonie der Operation bezüglich Vorder- und Hinterglied getrennt.

3.2.1 Monotonie im Hinterglied

Wir betrachten zunächst die Stabilität und Monotonie der Operation Fortsetzung in Verbindung der BOOLEsche Operationen $\cap$ und $\cup$ im Hinterglied.

Aus Eigenschaft 2 folgt:

Eigenschaft 4. $L \subseteq W \to L \triangleright V \subseteq W \triangleright V$

Dabei muss diese Eigenschaft nicht notwendigerweise für die echte Inklusion gelten, wie das folgende Beispiel zeigt:

Beispiel 3. Es Es sei $L = \{a\}, W = \{a, ab\}$ und $V = \{b\}$. Dann haben wir $L \subset W$, aber $L \triangleright V \not\subset W \triangleright W$ wegen $\emptyset \not\subset \emptyset$.

Aus Eigenschaft 4 folgt die Inklusion:

Eigenschaft 5. $(L \cap V) \triangleright W \subseteq (L \triangleright W) \cap (V \triangleright W)$

Dabei muss nicht notwendig die Gleichheit gelten, wie das folgende Beispiel zeigt

Beispiel 4. Es seien $L = \{aa, bb\}, V = \{aa, b\}, W = \{aa, bb\}$.
Dann haben wir $(L \cap V) \triangleright W = \{aa\} \subset \{aa, bb\} = (L \triangleright W) \cap (V \triangleright W)$

3.2.2 Monotonie im Vorderglied

Anders als bei der vorherigen Betrachtung, leiten sich in diesem Abschnitt kaum Folgerungen direkt aus der Definition ab. So haben wir, im Gegensatz zu Eigenschaft 2:

Eigenschaft 6. $L \triangleright (V \cup W) \subseteq (L \triangleright V) \cup (L \triangleright W)$

Zum Beweis nutzen wir die folgende Beziehung.

Lemma 1. *Min* $(A \cup B) \subseteq$ *Min* $A \cup$ *Min* B

Beweis von Lemma 1. Es sei $w \in Min\ (A \cup B)$. Dann ist w in A oder in B enthalten. OBdA Es sei $w \in A$. Es gilt außerdem für alle $v \sqsubset w$, dass $v \notin (A \cup B)$, also insbesondere $v \notin A$. Nach Definition von *Min* gilt also $w \in Min\ (A)$. Gleiches gilt analog, wenn man annimmt $w \in B$. □

Beweis von Eigenschaft 6. Es genügt, die Eigenschaft für $L = \{w\}$ zu zeigen.
Es gilt $w \triangleright (W \cup V) = Min\ (w \cdot X^* \cap (W \cup V)) = Min\ ((w \cdot X^* \cap W) \cup (w \cdot X^* \cap V))$ und mit Lemma 1 erhalten wir:
$w \triangleright (V \cup W) \subseteq Min\ (w \cdot X^* \cap V) \cup Min\ (w \cdot X^* \cap W) = (L \triangleright V) \cup (L \triangleright W)$ □

Dabei muss nicht notwendig die Gleichheit gelten, wie das folgende Beispiel zeigt.

Beispiel 5. Es seien $L = \{a, b\}, V = \{aaa\}, W = \{bb, aa\}$.
Dann haben wir $L \triangleright (V \cup W) = \{aa, bb\} \subset \{aa, bb, aaa\} = (L \triangleright V) \cup (L \triangleright W)$

Eine ähnliche Beziehung ergibt sich für die Operation $\cap$

Gleichung 3.1. $L \triangleright (V \cap W) \supseteq (L \triangleright V) \cap (L \triangleright W)$

Zum Beweis nutzen wir die Beziehung.

Lemma 2. *Min* $(A \cap B) \supseteq$ *Min* $A \cap$ *Min* B

Beweis von Lemma 2. Es sei $w \in (Min\ (A) \cap Min\ (B))$. Dann ist w sowohl in A, als auch in B enthalten. Außerdem gilt für alle v mit $v \sqsubset w$, dass $v \notin A$ und dass $v \notin B$. Nach Definition von *Min* gilt also $w \in Min\ (A \cap B)$. □

Beweis von Eigenschaft 3.1. Es genügt die Eigenschaft für $L = \{w\}$ zu zeigen.
Es gilt $w \triangleright (V \cap W) = Min\ (w \cdot X^* \cap (V \cap W)) = Min\ ((w \cdot X^* \cap V) \cap (w \cdot X^* \cap W))$ und mit Lemma 2 erhalten wir
$w \triangleright (V \cap W) \supseteq Min\ (w \cdot X^* \cap V) \cap Min\ (w \cdot X^* \cap W) = (L \triangleright V) \cap (L \triangleright W)$ □

Dabei muss nicht notwendig die Gleichheit gelten, wie das folgende Beispiel zeigt

Beispiel 6. Es seien $L = \{a, b\}\,, V = \{aaa, b, bb\}\,, W = \{bb, aaa\}$
Dann haben wir $L \triangleright (V \cap W) = \{bb, aaa\} \supset \{aaa\} = (L \triangleright V) \cap (L \triangleright W)$

Es gilt keine Monotonie im Vorderglied, $L \subseteq W \not\to V \triangleright L \subseteq V \triangleright W$, wie das folgende Beispiel zeigt:

Beispiel 7. Es sei $L = \{aaba\}, W = \{aab, aaba\}$ und $V = \{a\}$, also $L \subseteq W$.
Dann haben wir $V \triangleright L = \{aaba\} \not\subseteq V \triangleright W = \{aab\}$

3.3 Konkatenation

Bisher wurden hauptsächlich die boolschen Operationen $\cap$ sowie $\cup$ in Verbindung mit der Operation Fortsetzung $\triangleright$ betrachtet. Nun betrachten wir ob spezielle Eigenschaften der Operation Fortsetzung mit der Konkatenation existieren.

Es existieren Sprachen L, V und W derart, dass $L \triangleright (V \cdot W) = (L \triangleright V) \cdot (L \triangleright W)$ gilt.

Beispiel 8. Es seien $L = \{e\} \quad , V = \{a\} \quad , W = \{b\}$, so erhalten wir
$L \triangleright (V \cdot W) = \{ab\} = \{ab\} = (L \triangleright V) \cdot (L \triangleright W)$, also $L \triangleright (V \cdot W) = (L \triangleright V) \cdot (L \triangleright W)$

Weiterhin existieren Sprachen derart, dass $L \triangleright (V \cdot W) \supset (L \triangleright V) \cdot (L \triangleright W)$ gilt.

Beispiel 9. Es seien $L = \{a\} \quad , V = \{aa\} \quad , W = \{b, a\}$
Dann erhalten wir $L \triangleright (V \cdot W) = \{aab, aaa\} \supset \{aaa\} = (L \triangleright V) \cdot (L \triangleright W)$, also $L \triangleright (V \cdot W) \supset (L \triangleright V) \cdot (L \triangleright W)$.

Es existieren Sprachen derart, dass $L \triangleright (V \cdot W) \subset (L \triangleright V) \cdot (L \triangleright W)$ gilt.

Beispiel 10. Es seien $L = \{ab, e\} \quad , V = \{aba, bab\} \quad , W = \{e, aba, bab\}$, so erhalten wir $L \triangleright (V \cdot W) = L \triangleright \{aba, abaaba, ababab, bab, bababa, babbab\} = \{aba, bab\}$ und $(L \triangleright V) \cdot (L \triangleright W) = \{aba, bab\} \cdot \{e, aba\} = \{aba, abaaba, bab, bababa\}$. Also haben wir $L \triangleright (V \cdot W) \subset (L \triangleright V) \cdot (L \triangleright W)$

Zuletzt existieren Sprachen derart, dass sowohl $L \triangleright (V \cdot W) \not\subset (L \triangleright V) \cdot (L \triangleright W)$ als auch $L \triangleright (V \cdot W) \not\supset (L \triangleright V) \cdot (L \triangleright W)$ gelten. Es gilt also eine mengentheoretische Unvergleichbarkeit.

Beispiel 11. Es seien $L = \{aab, a\} \quad V = \{aa\} \quad W = \{b\}$
Dann erhalten wir $L \triangleright (V \cdot W) = \{aab\} \neq \{aa\} = (L \triangleright V) \cdot (L \triangleright W)$, das bedeutet also, dass beide Seiten mengentheoretisch unvergleichbar sind, also $L \triangleright (V \cdot W) \not\supset (L \triangleright V) \cdot (L \triangleright W)$, sowie $L \triangleright (V \cdot W) \not\subset (L \triangleright V) \cdot (L \triangleright W)$

Da wir für alle Inklusionsbeziehungen Sprachen gefunden haben, die diese erfüllen, so kann es keine Eigenschaft bezüglich der Inklusion geben, welche für alle Sprachen L, V, W gelten würde.

Betrachtet man nun die Konkatenation 'vorn', also die Beziehung $(L \cdot V) \triangleright W$ zu $(L \triangleright W) \cdot (V \triangleright W)$, so sieht man leicht, dass es sich auf der linken Seite der Gleichung um Wörter aus W handelt die man mit Wörtern aus W^2 vergleicht.
Demnach treten hier spezielle Inklusionsbeziehungen nur auf, wenn W die Gestalt $W \subseteq W^2$ oder $W \supseteq W^2$ hat.

4 Eigenschaften bei Sprachen spezieller Gestalt

In diesem Abschnitt wird die Operation Fortsetzung untersucht, wenn eine der beiden Operanden die spezielle Gestalt $pref(L)$ oder $W \cdot X^*$ hat.

Eigenschaft 7. $W \triangleright pref(V) = W \cap pref(V)$

Beweis. Die Inklusion $\supseteq$ folgt aus Eigenschaft 2.
Zum Beweis der anderen Inklusion betrachten wir $w \in W \triangleright pref(V)$. Es muss also ein $u \in W$ existieren, sodass $w \in \{u\} \triangleright pref(V)$. Daraus folgt, dass u ein Präfix von w sein muss, was wiederum bedeutet, dass $u \in pref(V)$. Damit haben wir $w \in Min\ (\{u\} \cdot X^* \cap pref(V)) \to w \in \{u\} \cap pref(V)$. □

Für die Umkehrung von Eigenschaft 7 $pref(V) \triangleright W$ gibt es keine Vereinfachung, wie das folgende Beispiel zeigt.

Beispiel 12. Es seien $V = \{aab, ab\}$ und $W = \{a, abb, aaab\}$, so dass $pref(V) = \{e, a, aa, ab, aab\}$. Dann haben wir $pref(V) \triangleright W = \{a, aab, aaab\}$, aber $pref(V) \cap W = \{a\}$ und $Min\ (pref(V) \setminus \{e\}) \triangleright W = \{a, abb\}$ Damit fehlt das Element $aaab$. Ein Betrachten von $Min\ (pref(V))$ hat keinen Sinn, da trivialerweise $Min\ (pref(V)) = \{e\}$, wenn $V \neq \emptyset$.

Eigenschaft 8. $V \cdot X^* \triangleright W = V \cdot X^* \cap W$

Beweis. Die Inklusion $\supseteq$ folgt aus Eigenschaft 2. Zum Beweis der anderen Inklusion betrachten wir $w \in V \cdot X^* \triangleright W$. Es existiert also ein $v \in V \cdot X^*$, sodass $w \in \{v\} \triangleright W$ ist. Daraus folgt, dass v ein Präfix von w sein muss, was wiederum bedeutet, dass $w \in V \cdot X^*$. Somit gilt die Inklusion $V \cdot X^* \triangleright W \subseteq W \cap V \cdot X^*$. □

Eigenschaft 9. $W \triangleright V \cdot X^* = (W \triangleright Min\ (V)) \cup (W \cap V \cdot X^*)$

Zum Beweis von Eigenschaft 9 benutzen wir

Lemma 3. *Es sei* $L \subseteq X^* \setminus W \cdot X^*$*, so gilt* $L \triangleright W \cdot X^* = L \triangleright Min\ (W)$

Beweis Lemma 3. :
Die Inklusion $L \triangleright Min\ (W) \subseteq L \triangleright W \cdot X^*$ ist, wie man leicht sieht, trivial.
Zum Beweis der der anderen Inklusion genügt es, diese für den Fall $L = \{v\}$ zu zeigen.
Es Es sei nun $w \in v \triangleright W \cdot X^*$, wobei nach Vorraussetzung $v \notin W \cdot X^*$ gelte. Dann gilt für kein u, $v \sqsubseteq u \sqsubset w$ oder $u \sqsubseteq v$ die Beziehung $u \in W \cdot X^*$.
Also haben wir $w \in Min\ (W \cdot X^*) = Min\ (W)$ □

Beweis von Eigenschaft 9. Die Fortsetzung von W in $V \cdot X^*$ lässt sich, mit Hilfe von Eigenschaft 3, in zwei Teile aufspalten (1). Da $W \setminus V \cdot X^* \subseteq X^* \setminus V \cdot X^*$, wenden wir Lemma 3 an und erhalten (2), was sich zu (3) vereinfachen lässt.

$$W \triangleright V \cdot X^* = ((W \setminus V \cdot X^*) \triangleright V \cdot X^*) \cup (W \cap V \cdot X^*) \tag{1}$$
$$W \triangleright V \cdot X^* = ((W \setminus V \cdot X^*) \triangleright Min\ (V \cdot X^*) \cup (W \cap V \cdot X^*) \tag{2}$$
$$W \triangleright V \cdot X^* = (W \triangleright Min\ (V)) \cup (W \cap V \cdot X^*) \tag{3}$$

□

5 Abgeschlossenheitseigenschaften

In diesem Abschnitt betrachten wir die Abgeschlossenheitseigenschaften der Klassen der CHOMSKY-Hierachie bezüglich der Operation Fortsetzung. Dabei betrachten wir in den folgenden Abschnitten die Klassen der refulären, kontextfreien, aufzählbaren und entscheidbaren Sprachen.

5.1 Regularität

Satz 1. *Sind L und W reguläre Sprachen, so ist auch $L \triangleright W$ regulär.*

Beweis. Der nicht-deterministische Automat $A_L = (X, Z, z_0, \delta_L, Z_f)$ akzeptiere L, der deterministische Automat $A_W = (X, S, s_0, f, S_f)$ akzeptiere W. Wir konstruieren einen nicht-determnninistischen Automaten A, der $L \triangleright W$ akzeptiert

Arbeitsweise:
In Phase eins lesen A_L und A_W das Wort w parallel (1). Falls A_L ein Präfix v von w akzeptiert, wählt A nicht-deterministisch aus ob Schritt zwei aktiviert wird oder nicht (Nicht-Determinismus von A_L).
Wird Phase zwei aktiviert, so liest A_W das Wort w weiter, während A_L in einem Ruhezustand z'_f verweilt (2). Akzeptiert A_W nun ein Präfix v' mit $v \sqsubseteq v' \sqsubset w \quad v' \in W$, so lehnt A das Eingabewort w ab. Dazu geht A in den Zustand (z'_f, s_x) aus dem keine Transition herausführt (3). Findet der Automat A_W kein solches Präfix v' und akzeptiert das Eingabewort w (und kein Präfix von w), so akzeptiert auch A und endet im Zustand (z'_f, s_x) (4).

$A = (X, (Z \cup \{z'_f\}) \times (S \cup \{s_x\}), (z_0, s_0), \delta, \{(z'_f, s_x)\}), s_x \notin S$ mit

$$
\begin{aligned}
\delta &= \{((z_i, s_i), x, (z_j, s_j)) : (z_i, x, z_j) \in \delta_L \wedge f(s_i, x) = s_j\} && (1)\\
&\cup \{((z_i, s_i), x, (z'_f, s_j)) : (z_i, x, z') \in \delta_L \wedge z' \in Z_f \wedge f(s_i, x) = s_j\} && (2)\\
&\cup \{((z'_f, s_i), x, (z'_f, s_j)) : f(s_i, x) = s_j \wedge s_j \notin S_f\} && (3)\\
&\cup \{((z'_f, s_i), x, (z'_f, s_x)) : f(s_i, x) = s_j \wedge s_j \in S_f\} && (4)
\end{aligned}
$$

Nach Konstruktion ist klar, dass der Automat A nur Wörter aus $L \triangleright W$ akzeptiert. Er akzeptiert auch alle Wörter aus $L \triangleright W$, weil der Automat A_L, welcher die Präfixe akzeptiert, nicht-deterministisch entscheidet, wann Phase zwei beginnt. In Phase zwei läuft dann nur A_W deterministisch auf der Eingabe w weiter und akzeptiert, wenn das nicht-deterministisch ausgewählte Präfix in w fortgesetzt werden kann. Damit akzeptiert der Automat A ein Eingabewort w genau dann, wenn $w \in L \triangleright W$ $\square$

5.2 Kontextfreiheit

Es existieren deterministisch kontextfreie, lineare Sprachen L, W derart, dass $L \triangleright W$ nicht einmal kontextfrei ist.

Beispiel 13. Es seien $L = \{a^n b^n c^i : i, n > 0\}$ und $W = \{a^i b^n c^n : i, n > 0\}$. Sowohl L als auch W sind deterministische, kontextfreie und auch lineare Sprachen, da es je einen deterministischen Kellerautomaten gibt, der L sowie W akzeptiert und es lineare Grammatiken

$G_L = (\{S, A, C\}, \{a, b, c\}, S, \{(S, Cc), (C, Cc), (C, aAb), (A, aAb), (A, e)\})$ und
$G_W = (\{S, A, B\}, \{a, b, c\}, S, \{(S, aA), (A, aA), (A, bBc), (B, bBc), (B, e)\})$ gibt, sodass
$L(G_L) = L$ und $L(G_W) = W$ gilt.

Betrachten wir nun ein Wort $u \in L \triangleright W$ so muss u laut Definition folgende Struktur besitzen $u \in Min\ (v \cdot X^* \cap W)$ für ein $v \in L$. Damit sieht man leicht, dass $L \triangleright W = \{a^n b^n c^n : n > 0\}$ gilt. Diese Sprache ist bekanntlich nicht einmal kontextfrei.

5.3 Entscheidbarkeit und Aufzählbarkeit

In diesem Abschnitt betrachten wir, ob für entscheidbare bzw. aufzählbare Sprachen L und W auch deren Fortsetzung $L \triangleright W$ wiederum entscheidbar bzw. aufzaehlbar ist. Dabei betrachten wir drei verschieden Fälle.

Wir betrachten zunächst $L \triangleright W$, wenn L und W entscheidbar sind. Dabei werden wir feststellen, dass auch die Fortsetzung $L \triangleright W$ entscheidbar ist. Ein Algorithmus wird dazu angegeben.
Anschließend werden wir die Fortsetzung betrachten, wenn L nur aufzählbar und W entscheidbar ist. In diesem Fall ist die Fortsetzung von L in W ebenso aufzählbar. Dazu geben wir einen Algorithmus an.
Im letzten Fall betrachten wir $L \triangleright W$, wenn W nur aufzählbar ist. In diesem Fall ist die Fortsetzung beider Sprachen nicht einmal aufzählbar. Dies gilt sogar , wenn L eine endliche Sprache ist.

5.3.1 L und W entscheidbar

Satz 2. *Sind L und W entscheidbare Sprachen, so ist auch $L \triangleright W$ entscheidbar.*

Beweis. Wir wissen, dass zu jeder entscheidbaren Sprache L ein Algorithmus angegeben werden kann, welcher L entscheidet. Daher geben wir zum Beweis von Satz 2 einen Algorithmus an, welcher $L \triangleright W$ entscheidet.

Arbeitsweise des Algorithmus:
Zunächst wird geprüft ob $w \in W$, da dies eine zwingende Voraussetzung laut Definition der Operation Fortzsetung ist (# 1). Jetzt wissen wir, dass $w \in W$. Gilt zusätzlich $w \in L$, so ist klar, dass $w \in L \triangleright W$ nach Eigenschaft 2 (# 2).

Jetzt setzen wir w' mit $w' := w$ als Arbeitskopie. Im folgenden Schleifendurchlauf schneiden wir mit $cut()$ den letzten Buchstaben von w' ab (# 3). Gilt jetzt $w' \in W$ so müssen wir mit NEIN entscheiden, weil die Eingabe jetzt nicht mehr dem *Min* Kriterium genügt. Anschließend wird geprüft ob $w' \in L \setminus W$ (# 5) gilt. Wenn ja, dann haben wir ein das erste Präfix von w gefunden und können für die Eingabe w mit JA entscheiden.

Ansonsten wird die Schleife solange weiter durchlaufen, bis w' nur noch aus dem leeren Wort besteht, anschliessend wird abgelehnt, da kein Präfix von v von w gefunden wurde, welches in $L \setminus W$ liegt.

Algorithm 1 entscheide $L \triangleright W$

```
Input w
if (w ∉ W) then
   T rejects {# 1}
else
   if (w ∈ L) then
      T accepts {# 2}
   end if
end if
w' = w {# 3}
repeat
   w' ← cut(w') {# 4}
   if (w' ∈ W) then
      T rejects
   end if{# 5}
   if (w' ∈ L) then
      T accepts
   end if
until (w' == e)
T rejects
```

□

5.3.2 L aufzählbar, W entscheidbar

Satz 3. *Ist L eine aufzählbare Sprache und W eine entscheidbare Sprache, so ist $L \triangleright W$ aufzählbar.*

Beweis. Es Es sei $U = \{u_i : 0 \leq i \leq n \wedge u_i \in L\}$ eine Aufzählung von L.
Wir konstruieren einen Algorithmus welcher die Eingabe w akzeptiert genau dann, wenn $w \in L \triangleright W$. Ist $w \notin W$ so kann nach Definition auch nicht $w \in L \triangleright W$ gelten.

Wir wählen das Wort u_i aus der Aufzählung U von L aus (# 1). Gilt nun $u_i \sqsubseteq w$, so wird eine Arbeitskopie v mit $v := w$ erstellt (# 2). Im nächsten Schritt wird für alle Präfixe v mit $u_i \sqsubseteq v \sqsubset w$ sukzessive geprüft ob $v \in W$ (# 3).

Sollte dies der Fall sein, dann erfüllt das aufgezählte $u_i \in L$ nicht das Kriterium $u_i \in L \setminus W$. Die innere Schleife wird daher verlassen und die FOR Schleife wählt das nächste u_{i+1} aus(# 4).
Gilt aber für alle Präfixe $v \notin W$, so erreicht die innere Schleife ihre Abbruchbedingung und wir akzeptieren die Eingabe w (# 5).

Algorithm 2 akzeptiere $L \triangleright W$

```
Input w
Sei
if (w ∉ W) then
  T rejects
end if
{# 1}
for i = 0 to ∞ do
  if u_i ∈ pref({w}) then
    v := w {# 2}
    while v ≠ u_i do
      v ← cut(v) {# 3}
      EXIT IF v ∈ W {# 4}
    end while
    if v = u_i then
      T accepts {# 5}
    end if
  end if
end for
```

□

5.3.3 L entscheidbar, W aufzählbar

Satz 4. *Es sei L eine entscheidbare Sprache und W eine aufzählbare Sprache, so ist $L \triangleright W$ nicht notwendig aufzählbar.*

Beweis. Wir wählen ein $A \subseteq \mathbb{N}$ welches aufzählbar, aber nicht entscheidbar ist und setzen W wie in (1) und L wie in (2). W ist also aufzählbar und L sogar regulär und endlich. Angenommen (3) wäre aufzählbar, so wäre (4) auch aufzählbar. Das würde bedeuten, dass $\mathbb{N} \setminus A$ aufzählbar und damit A entscheidbar wäre. Dies ergibt aber einen Widerspruch zur Annahme, da wir A so gewählt haben, dass es nicht entscheidbar ist.

$$W = \{0^{n+1}\,1^{n+1} : n \in \mathbb{N}\} \cup \{0^{n+1}\,1 : n \in A\} \tag{1}$$

$$L = \{0\} \tag{2}$$

$$L \triangleright W = \{0^{n+1}\,1^{n+1} : n \in \mathbb{N} \land n \notin A\} \cup \{0^{n+1}\,1 : n \in \mathbb{N} \land n \in A\} \tag{3}$$

$$(L \triangleright W) \cap \{0^n\,1^n : n \in \mathbb{N}\} = \{0^{n+1}\,1^{n+1} : n \in \mathbb{N} \land n \notin A\} \tag{4}$$

□

6 Quellenverzeichnis

Literatur

[St87] L. Staiger, Sequential Mappings of ω-Languages, 1987, pp. 148–170.

[St97] L. Staiger, ω-Languages. in: Handbook of Formal Languages, (G. Rozenberg and A. Salomaa Eds.), Springer-Verlag, Berlin 1997, Vol. 3, pp. 339–387.

[Wagner94] K. Wagner, Einführung in die theoretische Informatik, Springer-Verlag, Berlin 1994.