Über die Operation Fortsetzung bei formalen Sprachen

Robert Hartmann

24. September 2010

Inhaltsverzeichnis

1 Einleitung

In dieser Arbeit untersuchen wir die Operation Fortsetzung bei formalen Sprachen. Diese Operation wird in der Arbeit [St87] eingeführt.

Wir definieren den δ-Limes einer Wortmenge W^δ mit (s. [St87, Seite X])

Definition 1. $W^\delta = \{\beta : \beta \in X^\omega \textit{ und } \textit{pref}(\beta) \cap W \textit{ ist unendlich}\}$

Die folgende Eigenschaft (13) aus [St87] ist leich einzusehen: $(U \cup W)^\delta = U^\delta \cup W^\delta$

Definition 2. *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* $\textit{pref}(\beta) \backslash W$ *endlich ist.*

Beispiele für (σ, δ)-Teilmengen sind alle endlichen Sprachen und deren Komplemente. Weitere Beispiele sind Sprachen der Form $\textit{pref}(U)$ oder $W \cdot X^*$.
Eine Eigenschaft für diese Teilmengen ergibt sich wiefolgt :

Satz 1 (St87)**.** *Sei* U *eine* $(\sigma, \delta) -$ *Teilmenge von* X^**, dann gilt:*
$(U \cap W)^\delta = U^\delta \cap W^\delta, \qquad \textit{für alle } W \subseteq X^*$

Nun wird die Operation "Fortsetzung" wie in [St87] eingeführt, im nachfolgenden als $\triangleright$ bezeichnet. Die Fortsetzung eines Wortes w in eine Sprache $V \subseteq X^*$ sei definiert als:

Definition 3. $w \triangleright V := \textit{Min}_{\sqsubseteq}\{v : v \in V \wedge w \sqsubseteq v\} = \textit{Min}\,(w \cdot X^* \cap V)$

Diese Operation wird wie folgt auf Sprachen ausgedehnt, dabei bezeichnen wir die Fortsetzung einer Sprache W in eine Sprache $V \subseteq X^*$ mit:

Definition 4. $W \triangleright V := \bigcup_{w \in W} w \triangleright V$

Diese Operation hat nun folgende Eigenschaft bezüglich des δ-Limes: Während

$$(W \cap U)^\delta = W^\delta \cap U^\delta$$

nur für (σ, δ)-Teilemgen gilt, so gilt

$$(W \triangleright U)^\delta = W^\delta \cap U^\delta$$

für sämtliche Sprachen
Daher wird nun im Verlauf der Arbeit die Operation Fortsetzung untersucht.

2 Notation

In diesem Abschnitt führen wir die Notationen ein, die in der Arbeit verwendet werden. Mit $\mathbb{N} = \{0, 1, 2, 3, ..\}$ bezeichnen wir die Menge der natürlichen Zahlen. 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 $v \in X^\omega$ bezeichnen wir $w \cdot v$ als deren Konkatenation. Für eine Sprache W ist $W^* = \bigcup_{i \in \mathbb{N}} W^i$. Weiterhin bezeichnet $|w|$ 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 $\mathit{pref}(L) = \{v : v \sqsubseteq w \wedge w \in L\}$ alle Präfixe aller Wörter aus L.

Weiterhin bezeichnen wir mit $\mathit{Min}\ (L) = \{w : w \in L \wedge \forall v (v \sqsubset w \rightarrow v \notin L)\}$ alle Wörter der Sprache L, in denen kein Wort Präfix eines anderen Wortes ist. Abschließend bezeichnen wir die Operation Fortsetzung einer Sprache L in die Sprache W als $L \triangleright W = \bigcup_{u \in L} \mathit{Min}\ (u \cdot X^* \cap W)$

3 Eigenschaften

3.1 Allgemeines

In diesem Abschnitt betrachten wir Eigenschaften der Operation der Fortsetzung von Sprachen. Hierbei wird zunächst die algebraische Struktur $(2^{X^*}, \triangleright)$ untersucht. Anschliessend werden insbesondere die Stabilität bzw. Monotonie bezüglich mengentheoretischen Operationen betrachtet.

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

3.2.1 Kommutativität

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

Beispiel 1. *Es seien* $L = \{b\}$ *und* $W = \{bb\}$*. Dann haben wir* $L \triangleright W = \{bb\}$*, aber* $W \triangleright L = \emptyset$*.*

3.2.2 Assoziativität

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

Beispiel 2. *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* $L \triangleright (V \triangleright W) = L \triangleright \{aa, bb\} = \{aa, bb\}$*.*

3.2.3 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 ein linksneutrales Element E_l, so dass $E_l \triangleright L = L$ für alle $L \subseteq X^*$, gilt. Man erkennt leicht, dass X^* ein solches Element ist, denn es gilt $X^* \triangleright L = L$.

Analog betrachten wir nun ein rechtsneutrales Element E_r, so dass $L \triangleright E_r = L$ für alle $L \subseteq X^*$, gilt. Auch hier erkennt man leicht, dass X^* ein solches Element ist, denn es gilt $L \triangleright X^* = L$

3.2.4 Nullelement

Die Suche nach dem Nullelement stellt sich als trivial dar. Wir suchen ein Nullelement N, so dass $L \triangleright N = N$ sowie $N \triangleright L = N$ für alle $L \subseteq X^*$ gilt. Man erkennt leicht, dass die leere Menge $\emptyset$ ein solches Nullelement darstellt, da $\emptyset \triangleright L = \emptyset$ und $L \triangleright \emptyset = \emptyset$ gelten.

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

3.3 BOOLEsche Operationen

In diesem Abschnitt befassen wir uns mit den Zusammenhängen der mengentheoretischen Operationen $\cap, \cup$ bezüglich der Monotonie und der Stabilität der Operation Fortsetzung $\triangleright$.

Aus der Definition folgt direkt die Eigenschaft:

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

Es folgt ebenfalls eine generelle Inklusionsbeziehung:

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

3.3.1 Monotonie im Hinterglied

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

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 3. $(L \cup V) \triangleright W = L \triangleright W \cup V \triangleright W$

Aus der Definition 4 und aus Eigenschaft 3 folgt:

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

Betrachten wir nun $L \cap V \subseteq L$ und $L \cap V \subseteq V$, dann haben wir mit Eigenschaft 4 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 3. *Es seien* $L = \{aa, bb\}, V = \{aa, b\}, W = \{aa, bb\}$.
Dann haben wir $(L \cap V) \triangleright W = \{aa\} \subset \{aa, bb\} = (L \triangleright W) \cap (V \triangleright W)$

3.3.2 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 3:

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. Sei $w \in Min\ (A \cup B)$. Dann ist w in A oder in B enthalten. OBdA 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 Gleichung 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 4. *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. 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 5. *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.

Eigenschaft 7. $L \subseteq W \not\rightarrow V \triangleright L \subseteq V \triangleright W$

Wie das folgende Beispiel zeigt:

Beispiel 6. *Sei* $L = \{aaba\}, W = \{aab, aaba\}$ *und* $V = \{a\}$*, also* $L \subseteq W$*.*
Dann haben wir $V \triangleright L = \{aaba\} \not\subseteq V \triangleright W = \{aab\}$

3.3.3 Rest

Aus Eigenschaft 1 wissen wir, dass $L \cap W \subseteq L \triangleright W$. Man kann nun die Fortsetzung von L in W aufsplitten, sodass sich folgende Beziehung ergibt:

Eigenschaft 8. $L \triangleright W = L \cap W \cup (L \backslash W) \triangleright W$

Wir bilden nun Inklusionsbeziehungen zwischen L und W und erhalten mit Eigenschaft 8, zwei Folgerungen. Zu beachten ist, dass in Folgerung 1 eine „genau dann wenn“ Beziehung gilt.

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

Folgerung 2. $L \supseteq W \rightarrow L \triangleright W = L \cap W = W$

3.4 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 7. *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 8. *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 9. *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 10. *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 vergleicht mit Wörtern aus W^2.
Demnach treten hier Eigenschaften nur auf wenn W eine ganz spezielle Gestalt hat, nämlich $W \subseteq W^2$ oder $W \supseteq W^2$

4 Eigenschaften bei Sprachen spezieller Gestalt

Im nun vorliegenden Abschnitt wird die Operation Fortsetzung im Hinblick auf Sprachen mit spezieller Gestalt untersucht. Dabei geht es hauptsächlich um Sprachen der Form $pref(L)$ oder $W \cdot X^*$.

Eigenschaft 9. $W \triangleright pref(V) = W \cap pref(V)$

Beweis. Es genügt die Eigenschaft für $W = \{w\}$ zu zeigen:
Aus der Definition wissen wir, dass $w \triangleright pref(V) = Min\ (\{w\} \cdot X^* \cap pref(V))$ gilt. ist $w' \in (\{w\} \cdot X^* \cap pref(V))$, so sieht man, dass $w \sqsubseteq w'$ gilt. Also haben wir wegen $w' \in pref(V)$ auch $w \in pref(V)$. Daraus folgt $Min\ (\{w\} \cdot X^* \cap pref(V)) = \{w\} \cap pref(V)$ □

Eigenschaft 10. $V \cdot X^* \triangleright W = V \cdot X^* \cap W$

Beweis. Die Inklusion $\supseteq$ folgt aus Eigenschaft 1. Zum Beweis der anderen Inklusion betrachten wir $w \in V \cdot X^* \triangleright W$. Es muss nun ein $v \in V \cdot X^*$ existieren, sodass $w \in \{v\} \triangleright W$ gilt. 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 11. $W \triangleright V \cdot X^* = (W \triangleright Min\ (V)) \cup (W \cap V \cdot X^*)$

Zum Beweis von Eigenschaft 11 benutzen wir

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

Beweis Lemma 3. :
Es gilt die grundlegende Beziehung: $\forall w \in Min\ (W)\ \nexists w' \in W \cdot X^* : w' \sqsubset w$. Aus dieser Beziehung folgt unmittelbar $u \in L \triangleright Min\ (W) \rightarrow u \in L \triangleright W \cdot X^*$.
Zum Beweis der der anderen Inklusion genügt es, diese für den Fall $L = \{v\}$ zu zeigen.
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 11. Die Fortsetzung von W in $V \cdot X^*$ lässt sich, mit Hilfe von Eigenschaft 8, in zwei Teile aufspalten: $W \triangleright V \cdot X^* = (W \backslash V \cdot X^*) \triangleright V \cdot X^* \cup (W \cap V \cdot X^*)$.
Nun setzen wir $V' = W \backslash V \cdot X^*$ so, dass $V' \subseteq X^* \backslash V \cdot X^*$ gilt. Mit Lemma 3 erhalten wir $W \triangleright V \cdot X^* = W \backslash V \cdot X^* \triangleright Min\ (V \cdot X^*) \cup (W \cap V \cdot X^*) = W \triangleright Min\ (V) \cup (W \cap V \cdot X^*)$. □

5 Abgeschlossenheit in der CHOMSKY-Hierachie

5.1 Regularität

Satz 2. *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:
Schritt Eins: A_L und A_W lesen 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 Schritt 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 begibt sich 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).

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

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

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, ob das gefunden Präfix zum Wort w gehört. Damit akzeptiert der Automat A ein Eingabewort w genau dann, wenn $w \in L \triangleright W$ $\square$

5.2 Kontextfreiheit

5.2.1 deterministisch kontextfrei

Es existieren deterministisch kontextfreie Sprachen L, W, sodass $L \triangleright W$ nicht deterministisch kontextfrei ist.

Sei $L = \{a^n b^n c^i : i, n > 0\}$ 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 eine 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 $u \in \mathit{Min}\ (v \cdot X^* \cap W)$ *für ein* $v \in L$. Damit sieht man leicht, dass $L \triangleright W = \{a^n b^n c^n : n > 0\}$ und diese Sprache ist bekanntlich nicht kontextfrei, also auch nicht deterministisch kontextfrei

5.3 Entscheidbarkeit

In diesem Abschnitt betrachten wir die Erhaltung der Entscheidbarkeit bzw. der Aufzählbarkeit von Sprachen L, W in Bezug auf deren Fortsetzung $L \triangleright W$. Dabei betrachten wir drei verschieden Fälle.

5.3.1 L und W entscheidbar

Satz 3. *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 konstruieren wir zum Beweis von Satz 3 einen Algorithmus 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 $w' \in L \backslash W$ (# 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 $L \backslash W$ liegt.

$\square$

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 akzeptierbar, W entscheidbar

Satz 4. *Ist L eine aufzählbare Sprache und W eine entscheidbare Sprache, so ist $L \triangleright W$ aufzählbar.*

Beweis. 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 zählen nun ein Wort $u \in L$ auf (# 1). Gilt nun $u \sqsubseteq w$, so wird eine Arbeitskopie v mit $v := w$ erstellt (# 2). Im nächsten Schritt wird für alle Präfixe v mit $u \sqsubseteq v \sqsubset w$ sukzessive geprüft ob $v \in W$ (# 3).

Sollte dies der Fall sein, dann erfüllt das aufgezählte $u \in L$ nicht das Kriterium $u \in L \backslash W$. Die innere Schleife wird daher verlassen(# 4). Ein neues $u \in L$ wird aufgezählt und der Algorithmus läuft mit diesem u weiter.
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

```
if (w ∉ W) then
  T rejects
end if
while true do
  zähle ein u ∈ L auf {# 1}
  if  u ∈ pref({w})  then
    v := w {# 2}
    while v ≠ u do
      v := w' mit w' · x = v, x ∈ X {# 3}
      EXIT IF v ∈ W {# 4}
    end while
    if v = u then
      T accepts {# 5}
    end if
  end if
end while
```

5.3.3 L entscheidbar, W akzeptierbar

Satz 5. *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 $W = \{0^{n+1}\ 1^{n+1} : n \in \mathbb{N}\} \cup \{0^{n+1}\ 1 : n \in A\}$. Also ist W aufzählbar. Betrachten wir nun $L \triangleright W = 0 \triangleright W = \{0^{n+1}\ 1^{n+1} : n \in \mathbb{N} \wedge n \notin A\} \cup \{0^{n+1}\ 1 : n \in \mathbb{N} \wedge n \in A\}$. Angenommen $0 \triangleright W$ wäre aufzählbar, so müsste $0 \triangleright W \cap \{0^n\ 1^n : n \in \mathbb{N}\} = \{0^{n+1}\ 1^{n+1} : n \in \mathbb{N} \wedge n \notin A\}$ wiederum aufzählbar sein. Weil aber laut Vorraussetzung A nicht entscheidbar ist, kann $X^* \backslash A$ nicht aufzählbar sein. Widerspruch zur Annahme. $\square$

6 Quellen und Literatur

Literatur

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