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 bezeichnen die Menge X^* als Menge aller endlichen Wörter über dem Alphabet X.
Wir bezeichnen weiterhin die Menge X^ω als Menge aller unendlichen Wörter über dem Alphabet X.
Sei ferner die Präfixrelation $\sqsubseteq$ wie üblich definiert:

Definition 1.

$$w \sqsubseteq b \Leftrightarrow w \cdot b' = b, \text{ für ein } b' \in X^*$$

$$pref(L) = \{v : v \sqsubseteq w \wedge w \in L\}$$

Es wird nun der δ-Limes einer Wortmenge W^δ definiert (s. [St87, Seite X])

Definition 2.

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

Die folgende Eigenschaft (13) aus [St87] ist leich einzusehen:

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

Definition 3. *Eine Sprache nennen wir eine (σ, δ)-Teilmenge von X^* genau dann, wenn für alle $\beta \in X^\omega$ entweder $pref(\beta) \cap W$ oder $pref(\beta) \backslash W$ endlich ist.*

Beispiele für (σ, δ)-Teilmengen sind alle endlichen Sprachen und deren Komplemente. Weitere Beispiele sind Sprachen der Form $pref(U)$ 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:*

$$(U \cap W)^\delta = U^\delta \cap W^\delta, \qquad \text{für alle } W \subseteq X^*$$

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 4.

$$w \triangleright V := Min_{\sqsubseteq}\{v : v \in V \wedge w \sqsubseteq v\} = Min\,(w \cdot X^* \cap V)$$

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

Definition 5.

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

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 Eigenschaften

2.1 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.

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

2.2.1 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* $W \triangleright L = \emptyset$*.*

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

2.2.3 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 $L \triangleright X^* = L$

2.2.4 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.

2.3 Boolsche Operationen

Aus der Definition folgen direkt die Eigenschaften:

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

Eigenschaft 2. $(L \cup V) \triangleright W = L \triangleright W \cup V \triangleright W$

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

Aus der Eigenschaft 3 folgt durch Einschränkung der Mengen W bzw. L:

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

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

Aus Eigenschaft 2 folgt:

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

Damit erhalten wir die Inklusion:

Gleichung 2.1. $(L \cap V) \triangleright W \subseteq (L \triangleright W) \cap (V \triangleright W)$

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 $(L \cap V) \triangleright W = \{aa\} \subset \{aa, bb\} = (L \triangleright W) \cap (V \triangleright W)$

Gleichung 2.2. $L \triangleright (V \cup W) \subseteq (L \triangleright V) \cup (L \triangleright W)$

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 2.2. 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 $\cup$

Gleichung 2.3. $L \triangleright (V \cap W) \supseteq (L \triangleright V) \cap (L \triangleright W)$

Zum Beweis bemühen wir die Beziehung aus

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 2.3. 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 $L \triangleright (V \cap W) = \{bb, aaa\} \supset \{aaa\} = (L \triangleright V) \cap (L \triangleright W)$

2.4 Konkatenation

Bisher wurden hauptsächlich die Operationen $\cap$ sowie $\cup$ in Verbindung mit $\triangleright$ betrachtet. Für die Konkatenation ergeben sich keine Eigenschaften allgemeingültiger Natur.

So gibt es Sprachen, bei denen die Gleichheit in folgender Weise gegeben ist:

Beispiel 6. *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 können wir Sprachen angeben, sodass Teilmengenbeziehung in folgender Weise existieren:

Beispiel 7. *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* $L \triangleright (V \cdot W) \supset (L \triangleright V) \cdot (L \triangleright W)$.

Das letzte Beispiel in diesem Abschnitt zeigt deutlich, dass keine allgemeingültigen Eigenschaften für die Operation $\cdot$ bezüglich $\triangleright$ existieren:

Beispiel 8. *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* $L \triangleright (V \cdot W) \not\subset (L \triangleright V) \cdot (L \triangleright W)$

Betrachtet man nun die Konkatenation 'vorn', also die Beziehung $(L \cdot V) \triangleright V$ 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.

3 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 $\mathit{pref}(L)$ oder $W \cdot X^*$.

Eigenschaft 5. $W \triangleright \mathit{pref}(V) = W \cap \mathit{pref}(V)$

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

Eigenschaft 6. $V \cdot X^* \triangleright W = V \cdot X^* \cap W$

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 7. $W \triangleright V \cdot X^* = (W \triangleright \mathit{Min}\ (V)) \cup (W \cap V \cdot X^*)$

Zum Beweis von Eigenschaft 7 benutzen wir

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

Beweis Lemma 3. :
Es gilt die grundlegende Beziehung: $\forall w \in \mathit{Min}\ (W)\ \nexists w' \in W \cdot X^* : w' \sqsubset w$. Aus dieser Beziehung folgt unmittelbar $u \in L \triangleright \mathit{Min}\ (W) \rightarrow u \in L \triangleright W \cdot X^*$.
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 $w \in \mathit{Min}\ (W \cdot X^*) = \mathit{Min}\ (W)$ □

Beweis von Eigenschaft 7. Die Fortsetzung von W in $\cdot X^*$ lässt sich 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 $W \triangleright V \cdot X^* = W \backslash V \cdot X^* \triangleright \mathit{Min}\ (V \cdot X^*) \cup (W \cap V \cdot X^*) = W \triangleright \mathit{Min}\ (V) \cup (W \cap V \cdot X^*)$. □

4 Abgeschlossenheit in der CHOMSKY-Hierachie

4.1 Regularität

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

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:
A_L und A_W lesen das Wort w parallel. Falls A_L ein Präfix v von w akzeptiert, wählt A nicht-deterministisch aus ob Schritt zwei aktiviert wird oder nicht.
Wird Schritt zwei aktiviert, so liest A_W das Wort w weiter, während A_L in einem Ruhezustand z'_f verweilt. Akzeptiert A_W nun ein Präfix v', $v \sqsubseteq v' \sqsubset w$, so lehnt A das Eingabewort w ab. Findet der Automat A_W kein solches Präfix v' und akzeptiert das Eingabewort w, so akzeptiert auch A.

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

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

Beweis. 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 $w \in L \triangleright W$ □

4.2 Kontextfreiheit

4.2.1 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 und G_W gibt, sodass $L(G_L) = L$ und $L(G_W) = W$ gilt.
Betrachtet man sich nun ein Wort $u \in L \triangleright W$ so muss u laut Definition folgende Struktur besitzen $u \in Min\ (v \cdot X^* \cap W)\ \textit{für ein}\ v \in L$. 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

4.3 Entscheidbarkeit

4.3.1 L und W entscheidbar

Satz 3. *Sind L und W entscheidbare Sprachen, so ist auch $L \triangleright W$ entscheidbar.*

Es gibt die Turing Maschinen T_L mit $L(T_L) = L$ und T_W mit $L(T_W) = W$.
Vom Eingabewort w wird in jedem Schleifendurchlauf ein Buchstabe hinten abgeschnitten und damit werden schrittweise alle Präfixe v' von w untersucht. Gilt nun $v' \in W$ so lehnen wir ab, weil ein kürzeres Wort aus W gefunden wurde. Gilt $v' \in L$ so akzeptieren wir.

Algorithm 1 entscheide $L \triangleright W$, Input w

```
if (w ∉ W) then
  T rejects
else
  if (w ∈ L) then
    T accepts
  end if
end if
w' = w
repeat
  w' ← cut(w')
  if (w' ∈ W) then
    T rejects
  end if
  if (w' ∈ L) then
    T accepts
  end if
until (w' == e)
T rejects
```

4.3.2 L akzeptierbar, W entscheidbar

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

Sei w die Eingabe. Wir zählen $v \in L$ auf und prüfen für alle u mit $v \sqsubseteq u \sqsubset w$, ob ein $u \in W$ liegt. Sollte dies der Fall sein, darf man für das aktuelle v nicht akzeptieren, daher wird ein neues $v \in L$ aufgezählt und der Algorithmus läuft weiter. Gilt für kein solches u die Beziehung $u \in W$, so akzeptieren wir.

Algorithm 2 akzeptiere $L \triangleright W$, Input w

```
if (w ∉ W) then
  T rejects
end if
while true do
  zähle ein u ∈ L auf
  if  u ∈ pref({w})  then
    v := w
    while v ≠ u do
      v := w' mit w' · x = v, x ∈ X
      exit WHEN v ∈ W
    end while
    if v = u then
      T accepts
    end if
  end if
end while
```

4.3.3 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. $\square$

5 Schlusswort

6 Quellen und Literatur

Literatur

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