Formale Sprachen

(parallele und regulierte Ersetzung)

SoSe 2010 in Trier

Henning Fernau
Universität Trier
fernau@uni-trier.de

Formale Sprachen Gesamtübersicht

- Organisatorisches / Einführung
  Motivation / Erinnerung / Fragestellungen

- Diskussion verschiedener Sprachklassen:
  gesteuerte Ersetzungsverfahren
  parallele Ersetzungsverfahren

- Algebraische Ansätze:
  abstrakte Sprachfamilien
  formale Potenzreihen

Organisatorisches

Vorlesung: Mittwochs 12-14 Uhr, HZ 203,
Vorschlag: 12.25-13.55 (schafft bequemere Mittagspause)

Übungsbetrieb in Form von einer “Großen Übungsgruppe”
Montags 16-18 Uhr, HZ 204
BEGINN: in der zweiten Semesterwoche

Dozentensprechstunde DO 13-14 in meinem Büro H 410 (4. Stock)

Mitarbeitersprechstunde (Stefan Gulan) DO 13-14 H 413

Scheinkriterien (benotet) bzw. Modulprüfung nach einer mündlichen Prüfung.

Abschlusseigenschaften, Trios, AFLs: ein algebraischer Ausflug

—Vorstellung weiterer Abschlusseigenschaften

—Beziehungen zwischen Abschlusseigenschaften

—Trios, AFLs

Morphis

Eine Abbildung $h : V^* \to X^*$ mit Alphabeten V und X heißt *Homomorphismus* $\iff \forall x, y \in V^* : h(xy) = h(x)h(y)$.
$h : V^* \to X^*$ kann zu $h : 2^{V^*} \to 2^{X^*}$ erweitert werden durch
$h(L) := \{ h(v) \mid v \in L \}, L \subseteq V^*$.
Für eine Sprachklasse $\mathcal{A}$ sei
$H(\mathcal{A}) := \{ h(L) \mid L \in \mathcal{A}, L \subseteq V^*$ für ein Alphabet V, $h : V^* \to X^*$ ein Homomorphismus $\}$.

Es gelten die bekannten Eigenschaften von Homomorphismen zwischen Wort(!)monoiden:

(1) $h(\lambda) = \lambda$ (Damit ist h auch Monoid-Homomorphismus in der alten Definition; wesentlich geht hier die Tatsache ein, dass X^* frei erzeugtes Monoid ist. (hübsche Übung)

(2) h ist eindeutig bestimmt durch seine Bilder auf V
(genauer: es gilt für $x \in V^*, x = a_1 \dots a_n$ mit $a_i \in V : h(x) = h(a_1) \dots h(a_n)$)

Inverse Morphis

Sei $h : V^* \to X^*$ ein Homomorphismus, $M \subseteq X^*$, dann ist der *inverse Homomorphismus* definiert durch $h^{-1}(M) := \{ v \in V^* \mid h(v) \in M \}$.
Für eine Sprachklasse $\mathcal{A}$ sei $H^{-1}(\mathcal{A}) := \{ h^{-1}(M) \mid M \in \mathcal{A}, M \subseteq X^*$ für ein Alphabet $X, h : V^* \to X^*$ ein Homomorphismus $\}$.

Übung 1 *Der Homomorphismus $h : \{a, b\}^* \to \{a, b\}^*$ sei gegeben durch $a \mapsto aa$ und $b \mapsto ab$. Kennzeichnen Sie die folgenden Sprachen in der "üblichen" mengentheoretischen Schreibweise!*

- $L_{1,n} = h^{-1}(a^n)$
- $L_2 = h^{-1}(\{a\}^*)$
- $L_{3,n} = h^{-1}((\{a, b\}^n)^*)$

λ-freie Homomorphismen

Ein Homomorphismus h heißt *λ-frei* oder *nicht-löschend*, wenn $h(v) = \lambda \Rightarrow v = \lambda$ ($\iff \forall a \in V : h(a) \neq \lambda$).
Bezeichnungsweisen auf Sprachfamilienebene:
$H_\lambda(\mathcal{A})$ und $H_\lambda^{-1}(\mathcal{A})$.

Beachte: Es gibt λ-freie, nicht-injektive Homomorphismen: $h : \{a, b\} \to a^* : h(a) := h(b) := a$.

Nochmal Regularität

Anmerkung 1 *$\mathcal{L}_3$ ist die kleinste Sprachklasse $\mathcal{A}$, die die Familie FIN der endlichen Sprachen enthält und für die $\mathrm{H}(\mathcal{A}) = \mathrm{H}^{-1}(\mathcal{A}) = \mathcal{A}$ gilt.*
Denn nach LATTEUX *und* LEGUY**: $\forall \mathrm{R} \in \mathcal{L}_3 : \exists$ Homomorphismen $f, g, h : \mathrm{R} = f^{-1}(g(h^{-1}(\{a\})))$.*

Um die Grundtechnik der Aussage von Latteux und Leguy zu sehen, beschäftigen Sie sich bitte mit der folgenden Aufgabe:

Übung 2 *Aus der Grundvorlesung ist Ihnen bekannt, dass die regulären Sprachen genau die Sprachen sind, die von NEA erkannt werden. Zeigen Sie:*

- *Eine reguläre Sprache R ist sternabgeschlossen, d.h., $\mathrm{R}^* = \mathrm{R}$, genau dann, wenn es einen R erkennenden nichtdeterministischen endlichen Automaten gibt, dessen einziger Endzustand der Anfangszustand ist.*

- *Jede sternabgeschlossene Sprache lässt sich darstellen als $\mathrm{R} = f^{-1}(g(h^{-1}(\{\lambda\})))$ für geeignete Homomorphismen f, g, h.*

*siehe ICALP'83, LNCS 154, Seite 420ff. sowie TURAKAINEN, Bulletin of the EATCS, Band 20, 1983, Seite 162ff.

Schnitt mit regulären Sprachen

$\mathcal{A} \wedge \mathrm{REG} := \{ L \cap R \mid L \in \mathcal{A}, R \in \mathcal{L}_3 \}$ bezeichnet den Schnitt einer Sprachklasse mit den regulären Sprachen.

Auf Sprachfamilien sind die *Trio-Operationen* die folgenden Operationen:
– Homomorphismus H_λ
– inverser Homomorphismus H_λ^{-1} und
– Schnitt mit regulären Sprachen $\wedge\mathrm{REG}$.

Eine Sprachfamilie heißt *Trio* gdw. sie ist gegen die Trio-Operationen abgeschlossen.

Weitere Operationen Substitution

Eine Abbildung $\sigma : V^* \to 2^{X^*}$ heißt *Substitution*
$\iff \forall x, y \in V^* : (\sigma(xy) = \sigma(x)\sigma(y) \wedge \sigma(\lambda) = \{\lambda\})$.
Eine Substitution ist eindeutig durch ihre Bilder $\sigma(a), a \in V$ bestimmt:
$\sigma(x) = \sigma(a_1)\sigma(a_2)\ldots\sigma(a_n)$ für $x = a_1 \ldots a_n, a_i \in V$
Man kann eine Substitution also als Ersetzung von Zeichen aus dem urspünglichen Alphabet durch eine ganze Sprache auffassen.
Intuitiv können die Terminale aus V als Metavariablen für Sprachen angesehen werden.

Fortsetzung auf 2^{V^*}: $L \subseteq V^*, \sigma(L) := \bigcup_{x \in L} \sigma(x)$.
Fortsetzung auf Sprachfamilien $\mathcal{A}$: $\sigma(\mathcal{A}) := \{\, \sigma(L) \mid L \in \mathcal{A}, L \subseteq V^*, \sigma : V^* \to 2^{X^*}$ eine Substitution$\}$

Eine Substitution $\sigma : V^* \to 2^{X^*}$ heißt
– λ*-frei* $\iff \forall a \in V : \lambda \notin \sigma(a)$
– *endlich* $\iff \forall a \in V : \sigma(a)$ ist endlich
– *regulär* $\iff \forall a \in V : \sigma(a) \in \mathcal{L}_3$
– *kontextfrei* $\iff \forall a \in V : \sigma(a) \in \mathcal{L}_2$
– $\mathcal{A}$ *-Substitution* $\iff \forall a \in V : \sigma(a) \in \mathcal{A}$

Eine Sprachfamilie $\mathcal{A}$ heißt *abgeschlossen unter Substitution* genau dann, wenn $\mathcal{A}$ abgeschlossen ist unter $\mathcal{A}$ -Substitution.

Weitere Operationen Inverse Substitution

Es seien $\sigma : V^* \to 2^{X^*}$ eine Substitution und $M \subseteq X^*$. Wir definieren
$\sigma^{-1}(M) := \{ v \in V^* \mid \sigma(v) \cap M \neq \emptyset \}$ und
$\sigma^{-1}(\mathcal{A}) := \{ \sigma^{-1}(M) \mid M \in \mathcal{A} \}$.

[Links/Rechts-] Quotienten ([Links/Rechts-] Derivate)

Es seien $v \in X^*, L \subseteq X^*$. Definiere
$\delta_v(L) := v\backslash L := \{ w \mid vw \in L \}$ (*Linksderivat*) und
$\delta_v^r(L) := L/v := \{ w \mid wv \in L \}$ (*Rechtsderivat*).
Also gilt: $\{ v \}\delta_v(L) = L \cap \{ v \}X^* = \{ x \in L \mid v \sqsubseteq x \}$.
Es seien $L, M \subseteq X^*$. Definiere
$L\backslash M := \bigcup_{v \in L} \delta_v(M) = \{ w \mid \exists v \in L : vw \in M \}$ (*Linksquotient*; Achtung, es besteht bei dieser in der Literatur leider üblichen Schreibweise Verwechslungsgefahr mit der Mengendifferenz!) sowie
$M/L := \bigcup_{v \in L} \delta_v^r(M) = \{ w \mid \exists v \in L : wv \in M \}$ (*Rechtsquotient*).

Weitere Operationen Mischen (engl.: shuffle)

$v\&_{\{1,2\}^*}w := \{\, v_1w_1v_2w_2\ldots v_nw_n \,|\, n \geq 1; v = v_1\ldots v_n; w = w_1\ldots w_n; v_i, w_i \in U^* \,\}$ und

$L\&_{\{1,2\}^*}M := \bigcup_{v\in L, w\in M} v\&_{\{1,2\}^*}w.$

Spiegelwort und **Spiegelsprache** (auch: *Umkehrung*)

Für $w \in X^*$ definiere rekursiv

$$\overleftarrow{w} := w^R := \mathrm{mi}(w) := \begin{cases} \lambda & ; w = \lambda \\ \overleftarrow{v}\ a & ; w = av \end{cases}$$

$$\overleftarrow{L} := L^R := \mathrm{mi}(L) := \{\, \overleftarrow{w} \,|\, w \in L \,\}$$

Übung 3 *Sie haben soeben den Shuffle- (oder Misch-) Operator $\&_{\{1,2\}^*}$ kennengelernt.*
Zum Aufwärmen: Wie sieht die Sprache $a^3b^3\&_{\{1,2\}^*}a^2b^2$ aus?
Natürlich kann man Sprachen auch mit sich selbst mischen.
Zeigen Sie zunächst, dass $\&_{\{1,2\}^*}$ assoziativ ist!
Folgern Sie, dass 2^{X^*} für jedes Alphabet X zusammen mit $\&_{\{1,2\}^*}$ ein Monoid bildet!
Wie sieht insbesondere das neutrale Element aus?
Zeigen Sie weiter, dass dieses Monoid sogar kommutativ ist!
Jetzt können wir iteriertes Mischen definieren, und zwar seien $\&_{\{1,2\}^*}{}^0L := L$ und $\&_{\{1,2\}^*}{}^nL := L\&_{\{1,2\}^*}(\&_{\{1,2\}^*}{}^{n-1}L)$.
Überlegen Sie sich, für welche Sprachen L die Beziehung $\&_{\{1,2\}^*}{}^nL \subseteq \&_{\{1,2\}^*}{}^{n+1}L$ für alle $n \geq 0$ gilt!

Es sei nun $\mathbb{D}_1$ die Dyck-Sprache über dem Klammerpaar $(,)$ (also die Sprache der wohlgeformten Klammerausdrücke), und L sei $\{\, (^n)^n \mid n \in \mathbb{N}_0 \,\}$.
Zeigen Sie, dass für alle $n \geq 0$ gilt: $\&_{\{1,2\}^*}{}^nL \subseteq \mathbb{D}_1$! Überlegen Sie sich, ob für irgendein n sogar $\&_{\{1,2\}^*}{}^nL = \mathbb{D}_1$ gilt, und zeigen Sie ihre Behauptung!
Machen Sie sich abschließend Gedanken zu der Behauptung $\bigcup_{n\geq 0} \&_{\{1,2\}^*}{}^nL = \mathbb{D}_1$!

Transduktionen

Eine *Transduktion* über V^* und X^* ist eine Teilmenge τ von $V^* \times X^*$.
$L \subseteq V^* : \tau(L) := \{ \alpha \in X^* \mid \exists v \in L : (v, \alpha) \in \tau \}$,
$L' \subseteq X^* : \tau^{-1}(L') := \{ v \in V^* \mid \exists \alpha \in L' : (v, \alpha) \in \tau \}$.

Anmerkung 2 *Eine Transduktion über V^* und X^* ist nur ein anderer Name für eine binäre Relation zwischen V^* und X^*.*

Ein a-*Tranduktor* ist ein 6-Tupel $M = (Z, V, X, \delta, z_0, Z_F)$ mit
Z eine endliche Menge von Zuständen, das Zustandsalphabet
V das Eingabealphabet
X das Ausgabealphabet
$z_0 \in Z$ der Startzustand
$Z_F \subseteq Z$ die Menge der Endzustände
$\delta \subseteq Z \times V^* \times X^* \times Z$ endliche Menge von *Transaktionen*
Setze $\delta^0 := \{ (z, \lambda, \lambda, z) \mid z \in Z \}$ und
$\delta^{i+1} := \{ (z, vw, \alpha\beta, z'') \mid \exists z' : (z, v, \alpha, z') \in \delta^i \wedge (z', w, \beta, z'') \in \delta \}$ sowie
$\delta^* := \bigcup_{i=0}^{\infty} \delta^i$.

Dann ist die von M erzeugte Transduktion $\tau_M := \{ (v, \alpha) \mid \exists z \in Z_F : (z_0, v, \alpha, z) \in \delta^* \}$.

Eine Transduktion heißt *rational* genau dann, wenn es einen a-Tansduktor M mit $\tau = \tau_M$ gibt.

M heißt *nicht-löschend* genau dann, wenn $\delta^* \cap (Z \times V \times \{\lambda\} \times Z) = \emptyset$. Eine rationale Transduktion heißt *nicht-löschend* genau dann, wenn es einen nicht-löschenden a-Tansduktor M mit $\tau = \tau_M$ gibt.

Beispiele für rationale Transduktionen (1):

- *Homomorphismen:*
  Betrachte $\tau = \{ (v, w) \mid v \in V^*, w = h(v) \} \subseteq V^* \times X^*, h : V^* \to X^*$ Homomorphismus, und $M = (\{z\}, V, X, \delta, z, \{z\})$ mit $\delta = \{(z, a, h(a), z) \mid a \in V\}$.
  Offenbar ist τ nicht-löschend wenn h nicht-löschend ist.
  Ganz entsprechend sieht man ein, dass endliche Substitutionen rationale Transduktionen sind.
  Kleine Aufgabe: Warum sind auch reguläre Substitutionen rationale Transduktionen?

- *Inverse Homomorphismen:*
  Betrachte $\tau = \{ (v, w) \,|\, v \in V^*, w \in h^{-1}(v) \} \subseteq V^* \times X^*, h : X^* \to V^*$ Homomorphismus, und $M = (\{z\}, V, X, \delta, z, \{z\})$ mit $\delta = \{(z, h(a), a, z) \mid a \in X\}$.
  Offenbar ist τ nicht-löschend.

- Eine k-*beschränkte Löschung* ist eine Transduktion $\tau \subseteq (V \cup \{\$\})^* \times V^*$ (wobei $\$ \notin V$), die nur auf

  $$(\bigcup_{i=0}^{k} \{\$^i\} V)^+ .$$

  definiert ist und dort den durch $a \mapsto a$ für $a \in V$ und $\$ \mapsto \lambda$ gegebenen Morphismus $g_V : (V \cup \{\$\})^* \to V^*$ realisiert. τ wird durch den nicht-löschenden a-Transduktor $M = (\{z\}, V \cup \{\$\}, V, \delta, z, \{z\})$ mit $\delta = \{(z, \$^i a, a, z) \mid 0 \leq i \leq k, a \in V\}$ modelliert.

<u>Übung</u> 4 *Manchmal findet sich in der Literatur eine leicht abgewandelte Definition von (rationalen) $\mathfrak{a}$-Transduktoren.*
Zeigen Sie: Eine Transduktion ist genau dann rational, wenn es einen $\mathfrak{a}$-Transduktor $M = (Z, V, X, \delta, z_0, Z_F)$ mit $\tau = \tau_M$ und

- *Transaktionen* $\delta \subseteq Z \times (V \cup \{\lambda\}) \times X^* \times Z$ *oder auch* $\delta \subseteq Z \times (V \cup \{\lambda\}) \times (X \cup \{\lambda\}) \times Z$ *oder auch*

- $|Z_F| = 1$

gibt.

Übung 5 *Eine Transduktion* τ *über* X^* *und* X^* *(d.h., Eingabe- und Ausgabe-Alphabet sind identisch) definiert eine Funktion* $f_\tau : 2^{X^*} \to 2^{X^*}$*. Zeigen Sie:*

- *Das Relationenprodukt* $\tau_1 \circ \tau_2$ *entspricht der Hintereinanderausführung (Komposition) von* $f_{\tau_1} \circ f_{\tau_2}$*, d.h.,* $f_{\tau_1 \circ \tau_2} = f_{\tau_1} \circ f_{\tau_2}$*.*

- *Die Hintereinanderausführung (in diesem obigen Sinne) zweier [nicht-löschender] rationaler Transduktionen ist eine [nicht-löschende] rationale Transduktion.*

Anmerkung: Manchmal werden in der Literatur “verallgemeinerte sequentielle Maschinen” (generalized sequential machines, kurz gsm) betrachtet. Diese sind spezielle a-Transduktoren, bei denen die endliche Transaktionenmenge allerdings durch $\delta \subseteq Z \times V \times X^* \times Z$ beschränkt ist. Mit Übung 4 sieht man, dass die wesentliche Einschränkung im Fehlen von λ-Übergängen besteht.
$\rightsquigarrow$ Sowohl gsm- als auch inverse gsm-Abbildungen sind rationale Transduktionen.

Mit den in der voranstehenden Übung 4 eingeführten Normalformen für rationale Transduktoren muss man vorsichtig sein im Falle nicht-löschender Transduktoren.
Nennen wir daher Transduktoren mit Übergangsrelation $\delta \subseteq Z \times (V \cup \{\lambda\}) \times X^+ \times Z$ *monoton*.
Es gilt:

Satz 3 *Eine rationale Transduktion τ ist genau dann nicht-löschend, wenn sie als Hintereinanderausführung $\tau = \tau_1 \circ \tau_2$ einer monotonen rationalen Transduktion τ_2 gefolgt von einer k-beschränkten Löschung τ_1 angegeben werden kann.*

Beweis: Wie soeben gesehen, sind monotone rationale Transduktionen sowie k-beschränkte Löschungen nicht-löschende rationale Transduktionen, und somit auch deren Komposition.

Ist umgekehrt τ eine nicht-löschende rationale Transduktion, so gibt es einen τ realisierenden nicht-löschenden a-Transduktor $M = (Z, V, X, \delta, z_0, Z_f)$. Setzen wir

$$k = \max\{|v|, |w| \mid \exists (z, v, w, z') \in \delta \cap Z \times V^* \times X^* \times Z\},$$

so ist

$$M_2 = (Z \cup \delta \times \{1, \ldots, k\}, V, X \cup \{\$\}, \delta_2, z_0, Z_f)$$

mit $\$ \notin X$ und $\delta_2 = \{(z, \lambda, \$, ((z, v, w, z'), 1)) \mid (z, v, w, z') \in \delta\} \cup \{(((z, v, w, z')i), a, b, z'') \mid (z, v, w, z') \in \delta \wedge 1 \leq i \leq k \wedge (i \leq |v| \rightarrow a = v_i) \wedge (i > |v| \rightarrow a = \lambda) \wedge (i \leq k - |w| \rightarrow b = \$) \wedge (i > k - |w| \rightarrow b = w_{i-k+|w|}) \wedge (i < k \rightarrow z'' = ((z, v, w, z'), i + 1)) \wedge (i = k \rightarrow z'' = z')\}$ monoton. τ_1 sei die naheliegende $k + 1$-beschränkte Löschung. □

Satz 4 (Nivat) *Eine Transduktion* $\tau \subseteq V^* \times X^*$ *ist genau dann rational, wenn es ein Alphabet* Y*, eine reguläre Sprache* $R \subseteq Y^*$ *und zwei Homomorphismen* $f : Y^* \to V^*$ *sowie* $g : Y^* \to X^*$ *gibt mit* $\tau = \{\, (f(y), g(y)) \mid y \in R\}$*.*

$$\begin{tikzcd}
 & R \subseteq Y^* \arrow[dl, "f"'] \arrow[dr, "g"] & \\
V^* \arrow[rr, "\tau"] & & X^*
\end{tikzcd}$$

Beweis:

„$\Rightarrow$“ Es sei $M = (Z, V, X, \delta, z_0, Z_F)$ ein rationaler Transduktor. Setze $Y := \delta$. Weiter sei

$$R := \{ (z_1, v_1, \alpha_1, z_1')(z_2, v_2, \alpha_2, z_2') \ldots (z_m, v_m, \alpha_m, z_m') \mid m \geq 0 \text{, wenn } z_0 \in Z_F \text{, sonst } m \geq 1; z_{i-1}' = z_i; (z_i, v_i, \alpha_i, z_i') \in Y = \delta; z_1 = z_0; z_m' \in Z_F \}.$$

R ist regulär, denn $R = L(G_R)$ mit folgender Typ-3-Grammatik:
$G_R = (N, T, S, P)$ mit $N := Z, T := Y, S := z_0$ und

$$P = \{ z \to (z, v, \alpha, z')z' \mid z, z' \in Z; (z, v, \alpha, z') \in Y = \delta \} \cup \{ z \to \lambda \mid z \in Z_F \}.$$

Definiere die Homomorphismen $f : Y^* \to V^*$ und $g : Y^* \to X^*$ durch
$f((z, v, \alpha, z')) := v$ und $g((z, v, \alpha, z')) := \alpha$.

Behauptung: Es gilt $\tau_M = \{ (f(y), g(y)) \mid y \in R \}$.

„$\subseteq$“ $(v, \alpha) \in \tau_M \iff \exists z_0, \ldots, z_m, \alpha_1, \ldots, \alpha_m, v_1, \ldots, v_m : z_0 \xrightarrow{v_1|\alpha_1} z_1 \xrightarrow{v_2|\alpha_2} z_2 \ldots \xrightarrow{v_m|\alpha_m} z_m, z_m \in Z_F, (z_{i-1}, v_i, \alpha_i, z_i) \in \delta, v = v_1 \ldots v_m, \alpha = \alpha_1 \ldots \alpha_m$.
Dann ist $y = (z_0, v_1, \alpha_1, z_1)(z_1, v_2, \alpha_2, z_2) \ldots (z_{m-1}, v_m, \alpha_m, z_m) \in R$, und es gilt $f(y) = v_1 \ldots v_m = v, g(y) = \alpha_1 \ldots \alpha_m = \alpha$.

„$\supseteq$“ $y \in R \Rightarrow y = (z_0, v_1, \alpha_1, z_1)(z_1, v_2, \alpha_2, z_2) \ldots (z_{m-1}, v_m, \alpha_m, z_m) \in Y^*$.

Dann ist $z_0 \xrightarrow{v_1|\alpha_1} z_1 \xrightarrow{v_2|\alpha_2} z_2 \ldots \xrightarrow{v_m|\alpha_m} z_m$ eine akzeptierende Berechnung von M mit Eingabe $v = v_1 \ldots v_m = f(y)$ und Ausgabe $\alpha = \alpha_1 \ldots \alpha_m = g(y)$. Also gilt $(f(y), g(y)) \in \tau_M$.

„$\Leftarrow$" Ist $R \subseteq Y^*$ regulär, so gibt es eine Typ-3-Grammatik $G = (N, Y, S, P)$, $P \subseteq N \times NY^* \cup N \times Y^*$ mit $R = L(G)$. Es sei z_f ein neues Symbol ($\notin N$). Wir betrachten den a-Transduktor $M = (Z, V, X, \delta, z_0, Z_F)$ mit $Z = N \cup \{ z_f \}, z_0 := S, Z_F := \{ z_f \}$ und
$\delta := \{ (A, f(y), g(y), B) \mid A \to By \in P \} \cup \{ (A, f(y), g(y), z_f) \mid A \to y \in P \}$.

Behauptung: Es gilt $\{ (f(y), g(y)) \mid y \in R \} = \tau_M$.

„$\subseteq$" Beweis kann über
$A \Rightarrow_G^i By \iff (A, f(y), g(y), B) \in \delta^i$ und $A \Rightarrow_G^i y \iff (A, f(y), g(y), z_f) \in \delta^i$
geführt werden. Dieses wiederum geht induktiv über i.
Dann gilt $(S, f(y), g(y), z_f) \in \delta^* \iff (f(y), g(y)) \in \tau_M$.

„$\supseteq$" $(v, \alpha) \in \tau_M \iff (S, v, \alpha, z_f) \in \delta^* \iff S = A_0 \xrightarrow{v_1|\alpha_1} A_1 \xrightarrow{v_2|\alpha_2} A_2 \ldots \xrightarrow{v_m|\alpha_m} A_m = z_f, v = v_1 \ldots v_m, \alpha = \alpha_1 \ldots \alpha_m$. Nach Definition von τ_M (durch R, f, g) existieren $y_1, \ldots, y_m \in Y$ mit $v_i = f(y_i), \alpha_i = g(y_i), A_{i-1} \to A_i y_i (1 \leq i \leq m)$ und $A_{m-1} \to y_m$. Aus $A_0 = S$ folgt $y = y_1 \ldots y_m \in R$ und überdies gilt $f(y) = v, g(y) = \alpha$ (eigentlich Induktion über Ableitungslänge). $\square$

Der Beweis des Satzes von Nivat liefert sofort:

Korollar 5 *Eine Transduktion* $\tau \subseteq V^* \times X^*$ *ist genau dann nicht-löschend rational, wenn es ein Alphabet* Y*, eine reguläre Sprache* $R \subseteq Y^*$ *und zwei Homomorphismen* $f : Y^* \to V^*$ *sowie* $g : Y^* \to X^*$ *gibt, wobei* g *nicht-löschend ist, mit* $\tau = \{\, (f(y), g(y)) \mid y \in R\}$*.*

Beispiele für rationale Transduktionen (2):
Um die Rationalität der im folgenden betrachteten Transduktionen zu zeigen, benutzen wir jeweils den Satz von Nivat. In manchen Beispielen wäre die (direkte) Angabe eines rationalen Transduktors allerdings einfacher (s.o.).

- *Identität auf* X^**:*
  Wir betrachten also $\tau = \{ (x, x) \mid x \in X^* \} \subseteq X^* \times X^*$. Es sei $X = \{ x_1, \ldots, x_n \}, V := X$.
  Setze $Y := V \times X = X \times X = \{ (x_1, x_1), \ldots, (x_1, x_n), (x_2, x_1), \ldots, (x_2, x_n), \ldots, (x_n, x_n) \}$.
  Weiter setzen wir $Y' := \{ (x_1, x_1), \ldots, (x_n, x_n) \} \subseteq Y$. $R := Y'^*$ ist reguläre Sprache über Y.
  $(X \times X)^*$ kann in offensichtlicher Weise in $X^* \times X^*$ eingebettet werden, f, g sind dann Projektionen.
  $f : Y^* \to X^* : (v, x) \mapsto v$, $g : Y^* \to X^* : (v, x) \mapsto x$.

- *Rationale Konstanten:*
  Es sei $\tau = \{ (v, w) \mid v \in V^*, w \in K \} \subseteq V^* \times X^*$, $K \subseteq X^*$ reguläre Sprache. Setze $Y := V \cup X$, o.B.d.A. $V \cap X = \emptyset$. $R := \{ vw \mid v \in V^*, w \in K \}$ ist reguläre Sprache über Y. Es werde
  $f : Y^* \to V^*$ definiert durch $f(a) := \begin{cases} a & ; a \in V \\ \lambda & ; a \in X \end{cases}$ und
  $g : Y^* \to X^*$ definiert durch $g(a) := \begin{cases} a & ; a \in X \\ \lambda & ; a \in V \end{cases}$.

- *Homomorphismen:*
  Betrachte $\tau = \{ (v, w) \mid v \in V^*, w = h(v) \} \subseteq V^* \times X^*, h : V^* \to X^*$ Homomorphismus.
  Setze $Y := V$. $R := Y^* = V^*$ ist reguläre Sprache über Y.
  Es sei $f : V^* \to V^*$ Identität und $g : V^* \to X^*$ mit $g = h$.

- *Inverse Homomorphismen:*
  Betrachte $\tau = \{ (v, w) \mid v \in V^*, w \in h^{-1}(v) \} \subseteq V^* \times X^*, h : X^* \to V^*$ Homomorphismus.
  Setze $Y := X$. $R := Y^* = X^*$ ist reguläre Sprache über Y.
  Es sei $f : X^* \to V^*$ mit $f = h$ und $g : X^* \to X^*$ Identität.

<u>Übung</u> 6 *Wir betrachten nochmals die sternabgeschlossenen regulären Sprachen, siehe Übung 2. Als zusätzliche Sprachoperationen betrachten wir in dieser Übung:*

- *Durchschnitt mit sternabgeschlossenen regulären Sprachen;*

- *Markierung* $\mu(u) = u\#$ *für alle* $u \in X^*$*, wobei* # *ein neuer Buchstabe ist, d.h.* $\# \notin X$.

Zeigen Sie: 1. Die Operation "Durchschnitt mit sternabgeschlossenen regulären Sprachen" lässt sich darstellen als $f^{-1} \circ g \circ h^{-1}$*, wobei* f, g, h *geeignet gewählte, von der sternabgeschlossenen reguären Sprache* R *abhängige Homomorphismen sind.*
2. Zeigen Sie unter Benutzung des Satzes von Nivat, dass sich jede rationale Transduktion darstellen lässt als Hintereinanderausführung von (1) einer Markierung, (2) einem inversen Homomorphismus, (3) einem Durchschnitt mit sternabgeschlossenen regulären Sprachen und (4) einem Homomorphismus. (Umgekehrt sind solche Hintereinanderausführungen rationale Transduktionen.)
3. Folgern Sie aus dem bisher Gezeigten, dass eine Transduktion τ *genau dann rational ist, wenn sie sich als* $\tau = h_4 \circ h_3^{-1} \circ h_2 \circ h_1^{-1} \circ \mu$ *schreiben läßt, wobei die* h_i *Homomorphismen, und* μ *eine Markierungsfunktion ist.*

Wir kennen viele Trios

Der Satz von Nivat erleichtert wesentlich den folgenden Nachweis des Abschlusses gegen die Trio-Operationen von den Sprachfamilien $\mathcal{L}_i, i \in \{0, 1, 2, 3\}$. Außerdem lassen sich so leicht abstrakte Eigenschaften formulieren und beweisen.

Satz 6 *Die Sprachfamilien $\mathcal{L}_i, i \in \{0, 1, 2, 3\}$, sind abgeschlossen gegenüber rationalen Transduktionen, die (bis auf $\mathcal{L}_1$) auch löschend sein dürfen.*

Beweis: Nachzurechnen: Abgeschlossenheit von $\mathcal{L}_i$ gegenüber rationalen Transduktionen.
Es sei also τ eine rationale Transduktion, gegeben durch a-Transduktor $M = (Z, V_T, V_T', \delta, z_0, Z_F)$.
Ist τ i.allg. löschend, so können wir $\delta = \delta^0 \cup \delta^1$ voraussetzen. Weiter sei $L \in \mathcal{L}_i$ durch eine Typ-i-Grammatik $G = (V_N, V_T, S, P)$ beschrieben mit Produktionen, die

$$P \cap ((V_N \cup V_T)^+ \times ((V_N \cup V_T)^* V_T (V_N \cup V_T)^*)) = P \cap (V_N \times (V_T \cup \{\lambda\}))$$

erfüllen. Wir konstruieren Typ-i-Grammatik $G' = (V_N', V_T', S', P')$ mit $L(G') = \tau(L(G))$.

- $i = 0$: Wir setzen $V'_N = (Z \times V_N \times Z) \cup \{\, \Gamma_{\alpha,z,z'} \,|\, z, z' \in Z \wedge \alpha \in V_T \cup \{\lambda\}\} \cup \{\, S', \#, \$ \,\}$. Ein Zeichen $(z, A, z') \in Z \times V_N \times Z$ bezeugt zweierlei: (1) wir hatten eine Simulation der ursprünglichen Typ-0-Grammatik beginnend mit dem Nonterminal A vor, (2) wir suchten den zu simulierenden Transduktor von Zustand z in den Zustand z' zu befördern mit Hilfe der Eingabe, die als Terminalwort schließlich von A abgeleitet wird.

$$
\begin{array}{rl}
P' := & \{\, S' \to \#(z_0, S, z_F) \,|\, z_F \in Z_F \,\} \\
\cup & \{\, \#A' \to A'\#, A'\# \to \#A', \# \to \$ \,|\, A' \in V'_N \,\} \\
\cup & \{\, \#(z_1, A_1, z_2)(z_2, A_2, z_3) \cdots (z_m, A_m, z_{m+1}) \to \#(y_1, B_1, y_2)(y_2, B_2, y_3) \cdots (y_n, B_n, y_{n+1}) \,| \\
 & A_1 \cdots A_m \to B_1 \cdots B_n \in P \wedge A_1, \ldots, A_m, B_1, \ldots, B_n \in V_N \wedge \\
 & z_1, \ldots, z_{m+1}, y_1, \ldots, y_{n+1} \in Z \wedge z_1 = y_1 \wedge z_{m+1} = y_{n+1} \,\} \\
\cup & \{\, \$(z, A, z') \to \Gamma_{\lambda,z,z'}\$ \,|\, A \to \lambda \in P \wedge z, z' \in Z \} \\
\cup & \{\, \Gamma_{\alpha,z,z'} \to \gamma\Gamma_{\alpha,z'',z'} \,|\, (z, \lambda, \gamma, z'') \in \delta \wedge \alpha \in V_T \cup \{\lambda\}\} \\
\cup & \{\, \Gamma_{\alpha,z,z'} \to \gamma \,|\, (z, \alpha, \gamma, z') \in \delta \wedge \alpha \in V_T \cup \{\lambda\}\} \\
\cup & \{\, \$(z, A, z') \to \Gamma_{a,z,z''}\Gamma_{\lambda,z'',z'}\$ \,|\, A \to a \in P \wedge z, z', z'' \in Z \} \\
\cup & \{\, \Gamma_{\lambda,z,z'}\$ \to \gamma \,|\, (z, \lambda, \gamma, z') \in \delta \wedge z' \in Z_F \,\}
\end{array}
$$

Der Beweis der Richtigkeit der Konstruktion müsste wieder per Induktion über die Ableitungslänge geführt werden. Wir geben hier nur eine intuitive Begründung.

"$\tau(L(G)) \subseteq L(G')$": O.E. können wir annehmen, dass eine Ableitung in der Grammatik G in zwei Phasen verläuft: In der ersten wird ein Nonterminalstring aufgebaut, der in der zweiten mit Hilfe von Produktionen der Form $A \to a$ (mit $A \in V_N$, $a \in (V_T \cup \{\lambda\})$) in einen Terminalstring verwandelt wird.
Während der Simulation der ersten Phase durch G' taucht in der Satzform # auf, um diese Phase zu kennzeichnen.
Um den Transduktor kümmert man sich nur insoweit, als dass Zustandsübergänge geraten werden und auf richtige Anschlüsse achtgegeben wird.
Während der Simulation der zweiten Phase, die durch das Zeichen $\$$ gekennzeichnet ist, wird die Richtigkeit der geratenen Übergänge mit Hilfe der Γ-Produktionen geprüft und ggf. der dank τ erforderliche Ausgabestring erzeugt.

"$L(G') \subseteq \tau(L(G))$": Wie schon erwähnt, kennzeichnen # und $\$$ zwei Phasen in der Ableitung von G' unmittelbar in der Satzform. Die erste, #-, Phase kann leicht ersichtlich durch G simuliert werden, wobei die hineingeratenen Zustandsübergänge in der zweiten, $\$$-, Phase durch den Transduktor simuliert werden.

- $i = 1$: Wir gehen von einer monotonen Grammatik G aus. Die Konstruktion ist der für $i = 0$ sehr ähnlich, muß allerdings auf nichtlöschende Produktionen und monotone Transduktoren hin abgeändert werden.

$$
\begin{array}{rl}
P' := & \{\, S' \to \lambda \,|\, S \to \lambda \in P \wedge z_0 \in Z_F \,\} \\
\cup & \{\, S' \to (z_0, S, z_F) \,|\, z_F \in Z_F \,\} \\
\cup & \{\, (z_1, A_1, z_2)(z_2, A_2, z_3) \cdots (z_m, A_m, z_{m+1}) \to (y_1, B_1, y_2)(y_2, B_2, y_3) \cdots (y_n, B_n, y_{n+1}) \,| \\
 & A_1 \cdots A_m \to B_1 \cdots B_n \in P \wedge A_1, \ldots, A_m, B_1, \ldots, B_n \in V_N \wedge \\
 & z_1, \ldots, z_{m+1}, y_1, \ldots, y_{n+1} \in Z \wedge z_1 = y_1 \wedge z_{m+1} = y_{n+1} \,\} \\
\cup & \{\, \Gamma_{\alpha,z,z'} \to \gamma \Gamma_{\alpha,z'',z'} \,|\, (z, \lambda, \gamma, z'') \in \delta \wedge \alpha \in V_T^* \,\} (*) \\
\cup & \{\, \Gamma_{\alpha,z,z'} \to \gamma \,|\, (z, \alpha, \gamma, z') \in \delta \wedge \alpha \in V_T^* \,\} (*) \\
\cup & \{\, (z, A, z') \to \Gamma_{a,z,z'} \,|\, A \to a \in P \wedge z, z' \in Z \,\} \\
\cup & \{\, (z, A, z') \to \Gamma_{a,z,z''} \Gamma_{\lambda,z'',z'} \,|\, A \to a \in P \wedge z, z', z'' \in Z \,\}
\end{array}
$$

Bei $(*)$ dürfen wir nur solche α's aufführen, welche auch in der Definition von der Überführungsrelation δ als "Eingaben" auftauchen, damit die Produktionenmenge endlich bleibt.

Nach Satz 3 müssen wir in unserem Fall noch die Abgeschlossenheit der kontextsensitiven Sprachen unter k-beschränkter Löschung nachweisen. Die Grundidee hierbei ist, eine Satzform $A_1 \ldots A_n$ (mit $(n \bmod 2k) = i$) in $2k$-Gruppen aufgespalten zu schreiben und zu behandeln, d.h., wir haben jetzt Satzformen der Art

$$
\begin{array}{c|c|c|c}
A_1 & A_{2k+1} & \ldots & A_{n-i+1} \\
A_2 & A_{2k+2} & \ldots & A_{n-i+2} \\
\vdots & \vdots & & \vdots \\
\vdots & \vdots & & A_n \\
A_{2k} & A_{4k} & \ldots & .
\end{array}
$$

Die vertikalen Säulen von $2k$ Zeichen werden als jeweils ein neues “Mega-Zeichen” interpretiert und die Zeichen der letzten Säule noch in dem “Mega-Zeichen” der vorigen Säule kodiert. Eine k-Löschung auf dem ursprünglichen Alphabet wird so eine nicht-löschenden Transduktion auf dem “Mega-Alphabet”. *

- $i = 2$: Die bekannte Tripelkonstruktion wird durch einen Zwischenschritt erweitert.

$$
\begin{array}{rl}
P' := & \{\, S' \to (z_0, S, z_F) \mid z_F \in Z_F \,\} \\
\cup & \{\, (y_1, A, y_{n+1}) \to (y_1, B_1, y_2)(y_2, B_2, y_3) \cdots (y_n, B_n, y_{n+1}) \mid \\
& A \to B_1 \cdots B_n \in P \wedge A, B_1, \ldots, B_n \in V_N \wedge y_1, \ldots, y_{n+1} \in Z \,\} \\
\cup & \{\, (z, A, z') \to \Gamma_{\lambda,z,z'} \mid A \to \lambda \in P \wedge z, z' \in Z \,\} \\
\cup & \{\, \Gamma_{\alpha,z,z'} \to \gamma \Gamma_{\alpha,z'',z'} \mid (z, \lambda, \gamma, z'') \in \delta \wedge \alpha \in V_T \cup \{\lambda\} \,\} \\
\cup & \{\, \Gamma_{\alpha,z,z'} \to \gamma \mid (z, \alpha, \gamma, z') \in \delta \wedge \alpha \in V_T \cup \{\lambda\} \,\} \\
\cup & \{\, (z, A, z') \to \Gamma_{a,z,z'} \mid A \to a \in P \wedge z, z' \in Z \,\} \\
\cup & \{\, (z, A, z') \to \Gamma_{a,z,z''} \Gamma_{\lambda,z'',z'} \mid A \to a \in P \wedge z, z', z'' \in Z \,\}
\end{array}
$$

*Die Idee dieser Konstruktion heißt auch “Bandkompression”, da sie üblicherweise bei Turing-Bändern vorgenommen wird.

- $i = 3$: Hier finden wir fast die Produktkonstruktion aus den Lehrbüchern. O.E. können wir $P \subseteq V_N \times ((V_T \cup \{\lambda\})V_N \cup \{\lambda\})$ voraussetzen. Es sei $V'_N = (Z \times (V_N \cup \{L\})) \cup \{S'\}$.

$$\begin{aligned}
P' := \;& \{ S' \rightarrow (z_0, S) \} \\
\cup \;& \{ (z, A) \rightarrow v(z', B) \mid A \rightarrow wB \in P \wedge w \in V_T \cup \{\lambda\} \wedge (z, w, v, z') \in \delta \} \\
\cup \;& \{ (z, A) \rightarrow v(z', L) \mid A \rightarrow w \in P \wedge w \in V_T \cup \{\lambda\} \wedge (z, w, v, z') \in \delta \} \\
\cup \;& \{ (z, A) \rightarrow v(z', A) \mid A \in V_N \cup \{L\} \wedge (z, \lambda, v, z') \in \delta \} \\
\cup \;& \{ (z, L) \rightarrow v \mid \exists z' \in Z_F (z, \lambda, v, z') \in \delta \}
\end{aligned}$$

$\square$

Abgeschlossenheit unter Homomorphismen

Satz 7 *(a)* $\mathcal{L}_0, \mathcal{L}_2, \mathcal{L}_3$ *sind* H*-stabil.*

(b) $\mathcal{L}_1$ *ist* H_λ*-stabil (abgeschlossen unter nicht-löschenden Morphismen).*

(c) $\mathrm{H}(\mathcal{L}_1) = \mathcal{L}_0$*.*

Beweis:

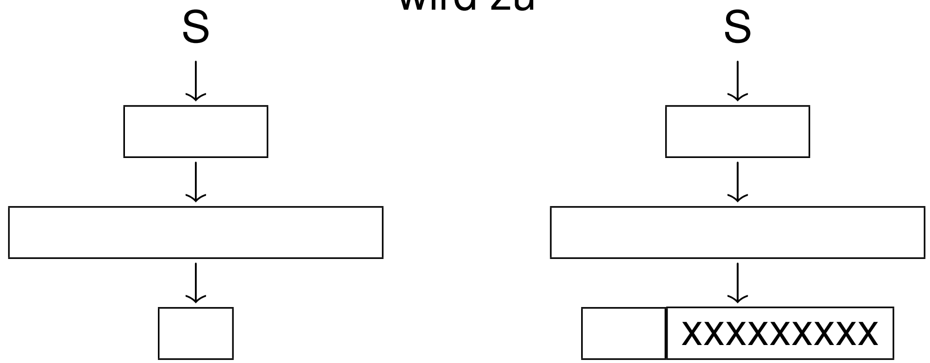
(a),(b) Dies folgt unmittelbar aus dem voranstehenden Satz und dem Satz von Nivat.

(c) Klar ist: $\mathrm{H}(\mathcal{L}_1) \subseteq \mathrm{H}(\mathcal{L}_0) = \mathcal{L}_0$.

Es bleibt also zu zeigen: $\mathcal{L}_0 \subseteq \mathrm{H}(\mathcal{L}_1)$, d.h., zu jeder Typ-0-Grammatik G gibt es eine Typ-1-Grammatik G' und einen Homomorphismus h mit $\mathrm{L}(\mathrm{G}) = \mathrm{h}(\mathrm{L}(\mathrm{G}'))$.

Hierbei stellt sich das folgende Problem: Typ-0-Grammatiken können löschend sein, bei Typ-1-Grammatiken bleiben die abgeleiteten Satzformen in jedem Schritt gleichlang oder werden länger.

Idee: wird zu



Ableitung in Typ-0-Grammatik

Ableitung in Typ-1-Grammatik

Die „x“-Symbole sind spezielle „don’t-cares“, die vom Homomorphismus dann auf λ abgebildet und dadurch eliminiert werden.

Es sei $G = (V_N, V_T, S, P)$ eine Typ-0-Grammatik in Normalform. Sei $\$$ ein neues Symbol, $\$ \notin V_N \cup V_T$. Setze $G' := (V_N, V_T \cup \{\$\}, S, P')$ mit

$$
\begin{aligned}
P' := \; & \{ A \to a \in V_N \times V_T \mid A \to a \in P \} \\
\cup \; & \{ A_1 \dots A_n \to B_1 \dots B_m \in V_N^+ \times V_N^* \mid A_1 \dots A_n \to B_1 \dots B_m \in P, n \leq m \} \\
\cup \; & \{ A_1 \dots A_n \to B_1 \dots B_m \$^{m-n} \in V_N^+ \times V_N^* \{ \$ \}^{m-n} \mid A_1 \dots A_n \to B_1 \dots B_m \in P, n > m \} \\
\cup \; & \{ \$A \to A\$ \mid A \in V_N \}
\end{aligned}
$$

Offensichtlich gilt:

(1) G' ist monoton, und damit liegt $L(G')$ in $\mathcal{L}_1$.

(2) $\forall w \in V_T^* : (S \Rightarrow_G^* w \iff \exists i \geq 0 : S \Rightarrow_{G'}^* w\$^i)$.
Das ist sichergestellt, weil in der Grammatik G' ein Wort so abgeleitet werden kann, dass erst als letzte Schritte die Terminal-Regeln angewandt werden, d.h. zuvor besteht die abgeleitete Satzform lediglich aus Nonterminalen. Dann ist es aber durch die Regeln $\$A \to A\$$ möglich, die „\$“-Symbole nach rechts zu schieben.
Das kann während des Ableitungsprozesses mehrmals nötig werden, weil die „\$“ eine Folge von Nonterminalen trennen können, für die ggf. eine linke Regelseite existiert!

Definiere Homomorphismus $h : (V_T \cup \{ \$ \}^*) \to V_T^* : \begin{cases} a \mapsto a, a \in V_T \\ \$ \mapsto \lambda \end{cases}$

Dann gilt $h(L(G')) = L(G)$. $\square$

Trios & Co.

Eine Sprachfamilie $\mathcal{A}$ heißt *Trio* (auch *Kegel*, *engl.: cone*) gdw.

(1) $\exists \mathrm{L} \in \mathcal{A} : \mathrm{L} \neq \emptyset$, d.h., es gibt eine nichttriviale Sprache in $\mathcal{A}$.

(2) $\mathcal{A}$ ist stabil gegen die Trio-Operationen, also H_λ, H^{-1} und $\wedge\mathrm{REG}$.

Ein Trio $\mathcal{A}$ heißt *voll* gdw. es auch noch gegen H (bel.!) abgeschlossen ist.
Das von einer Sprachfamilie $\mathcal{A}$ erzeugte Trio (Bezeichnung: *TRIO($\mathcal{A}$)*) ist das kleinste Trio, das $\mathcal{A}$ enthält, d.h. die kleinste Familie $\mathcal{B}$ mit:
– $\exists \mathrm{L} \neq \emptyset : \mathrm{L} \in \mathcal{B}$
– $\mathcal{A} \subseteq \mathcal{B}$ und
– $\mathcal{B} = \mathrm{H}_\lambda(\mathcal{B}) = \mathrm{H}_\lambda^{-1}(\mathcal{B}) = \mathcal{B} \wedge\mathrm{REG}$.

Beispiele und Folgerungen

Beispiel: TRIO$(\mathrm{FIN}) = \mathcal{L}_3$ (vgl. Anmerkung 1)

Korollar 8 (aus dem Satz von Nivat) *Eine Sprachfamilie ist genau dann ein volles Trio, wenn sie gegen beliebige rationale Transduktionen abgeschlossen ist.*

Eine Sprachfamilie ist genau dann ein Trio, wenn sie gegen beliebige nicht-löschende rationale Transduktionen abgeschlossen ist.

Übung 7 *Ist $\mathcal{L}$ ein volles Trio und $\mathrm{L} \in \mathcal{L}$, $\mathrm{R} \in \mathcal{L}_3$ beliebig, so sind $\mathrm{L} \cup \mathrm{R}$, $\mathrm{L} \cdot \mathrm{R}$, $\mathrm{R} \cdot \mathrm{L}$ und $\mathrm{L} \&_{\{1,2\}^*} \mathrm{R}$ in $\mathcal{L}$.*

Satz 9 *Jedes volle Trio ist unter Quotientenbildung mit einer regulären Menge abgeschlossen.*

Konstruktion: Es sei $\mathcal{L}$ ein volles Trio. Ferner seien $L \in \mathcal{L}$ und $R \in \mathcal{L}_3$ vorgegeben mit $L, R \subseteq V^*$. Für jedes $a \in V$ sei a' ein neues Symbol, und $V' := \{ a' \mid a \in V \}$. Wir definieren zwei Homomorphismen $f, h : (V \cup V')^* \to V^*$ durch $f(a) = f(a') = h(a') = a$ und $h(a) = \lambda$ für $a \in V$. Nun ist

$$L/R = \{ w \mid \exists v \in R : wv \in L \} = h(f^{-1}(L) \cap ((V')^* \circ R)) \in \mathcal{L}.$$

Entsprechend sieht man den Abschluss gegen Linksquotient. $\square$

Satz 10 *Jedes Trio ist unter Derivaten abgeschlossen.*

Konstruktion: (wie eben ...) Es sei $\mathcal{L}$ ein volles Trio. Ferner seien $L \in \mathcal{L}$ vorgegeben mit $L \subseteq V^*$ sowie $v \in V^*$. Für jedes $a \in V$ sei a' ein neues Symbol, und $V' := \{ a' \mid a \in V \}$. Wir definieren zwei nicht-löschende Homomorphismen $f, h : (V \cup V')^* \to V^*$ durch $f(a) = h(a) = a$ und $f(a') = av$ sowie $h(a') = a$ für $a \in V$. Nun ist

$$L/v = \{ w \mid wv \in L \} = h(f^{-1}(L) \cap (V^* \circ V')) \in \mathcal{L}.$$

Entsprechend sieht man den Abschluss gegen Linksderivat. $\square$

Satz 11 *Jedes volle Trio ist abgeschlossen gegenüber regulärer Substitution.*

Konstruktionsidee: Jedem Symbol a entspricht eine reguläre Sprache $\sigma(a)$. Ein a-Transduktor kann daher bei Einlesen von dem Symbol a den $\sigma(a)$ entsprechenden endlichen Automaten (mit Hilfe von "λ-Eingaben") simulieren und so ein beliebiges Wort aus $\sigma(a)$ als Ausgabe erzeugen. $\square$

Abstrakte Sprachfamilien

Eine Sprachfamilie $\mathcal{A}$ heißt *abstrakte Sprachfamilie*, kurz *AFL* (*engl.: abstract familiy of languages*) genau dann, wenn gilt:

(1) $\exists \mathrm{L} \in \mathcal{A} : \mathrm{L} \neq \emptyset$, d.h., es gibt eine nichttriviale Sprache in $\mathcal{A}$.

(2) $\mathcal{A}$ ist stabil gegen $\vee, \circ, ^+, \mathrm{H}_\lambda, \mathrm{H}^{-1}$ und $\wedge\mathrm{REG}$.

Eine AFL heißt *voll*, kurz *FAFL* (*engl.: full abstract familiy of languages*) gdw. sie ist ein volles Trio.

Die von einer Sprachfamilie $\mathcal{A}$ erzeugte AFL (Bezeichnung: $\mathrm{AFL}(\mathcal{A})$) ist die kleinste AFL, die $\mathcal{A}$ enthält.

Beispiele: $\mathcal{L}_3, \mathcal{L}_2$ und $\mathcal{L}_0$ sind FAFLs, $\mathcal{L}_1$ ist AFL.

Satz 12 *$\mathcal{L}_3$ ist die kleinste FAFL, d.h., jede FAFL enthält $\mathcal{L}_3$.*

Beweis: Es sei V_T ein „geeignetes" Grundalphabet der FAFL $\mathcal{A}$. Wir wollen zeigen, dass $R \subseteq V_T^*$, $R \in \mathcal{L}_3$ in $\mathcal{A}$ liegt.
Es sei $\emptyset \neq L \subseteq X^* \in \mathcal{A}$, $\varphi : X^* \to V_T^*$ ein Homomorphismus.
Es seien $h : V_T^* \to V_T^*$ und $g : X^* \to X^*$ die Homomorphismen, die alle Wörter auf das leere Wort abbilden.
Mit $L \in \mathcal{A}$ und der Abgeschlossenheit unter den Trio-Operationen gilt
$h^{-1}(\varphi(g(L))) = V_T^* \in \mathcal{A}$ (denn $g(L) = \varphi(g(L)) = \{e\}$) $\rightsquigarrow R = V_T^* \cap R \in \mathcal{A}$. $\square$

Da wir praktisch nur Trio-Operationen verwenden, haben wir hiermit auch Bemerkung 1 bewiesen.

Unabhängigkeit der AFL-Operationen

Satz 13 *Jedes $\mathcal{A}$, das stabil ist gegen $\vee, {}^*, \mathrm{H}, \mathrm{H}^{-1}$ und $\wedge\mathrm{REG}$, ist auch $\circ$-stabil.*

Beweis: Es seien $L_1, L_2 \in \mathcal{A}, L_1, L_2 \subseteq V_T^*; \$, \# \notin V_T$.
Wir definieren $\varphi : (V_T \cup \{\$, \#\})^* \to V_T^* : \begin{cases} \varphi(a) = a, a \in V_T \\ \varphi(\$) = \varphi(\#) = \lambda \end{cases}$
Dann gilt: $L_1\$ = \varphi^{-1}(L_1) \cap V_T^*\$ \in \mathcal{A}$ und $L_2\# = \varphi^{-1}(L_2) \cap V_T^*\# \in \mathcal{A} L_1\$L_2\# = (L_1\$ \cup L_2\#) \cap V_T^*\$V_T^*\# \in \mathcal{A} \Rightarrow L_1L_2 = \varphi(L_1\$L_2\#) \in \mathcal{A}$. $\square$

Mithin brauchen wir den Konkatenationsabschluss bei FAFL's nie zu beweisen.

Übung 8 *Gilt obiger Satz auch analog für nicht-volle AFL's ?*

Satz 14 *Wenn $\mathcal{L}_3 \subseteq \mathcal{A}$ und $\mathcal{A}$ substitutionsstabil und $\wedge\mathrm{REG}$-stabil, dann ist $\mathcal{A}$ eine FAFL.*

Beweis:

(1) klar: z.B. $V_T^* \in \mathcal{L}_3$, deshalb $\exists L \in \mathcal{A} : L \neq \emptyset$.

(2)
- Abgeschlossenheit unter Homomorphismen:
  $\mathcal{L}_3 \subseteq \mathcal{A}$ und $\mathcal{A}$ substitutionstabil $\Rightarrow \mathcal{A}$ ist H-stabil.
- Abgeschlossenheit unter Vereinigung:
  Seien $L_1, L_2 \in \mathcal{A}\,; L_1, L_2 \subseteq V_T^*$. Betrachte Substitution $\sigma : \{\, a, b \,\}^* \longrightarrow 2^{V_T^*}$ mit $\sigma(a) = L_1$ und $\sigma(b) = L_2$. Aus $\{\, a, b \,\} \in \mathcal{A}$ folgt $\sigma(\{\, a, b \,\}) = L_1 \cup L_2 \in \mathcal{A}$.
- Abgeschlossenheit unter Kleene-Stern:
  Sei $L \in \mathcal{A}, L \subseteq V_T^*$. Dann betrachte Substitution $\sigma : \{\, a \,\}^* \longrightarrow 2^{V_T^*}$ mit $\sigma(a) = L$. Aus $a^* \in \mathcal{A}$ folgt $L^* = \sigma(a^*) \in \mathcal{A}$.

- Abgeschlossenheit unter rationalen Transduktionen: Es sei $L \in \mathcal{A}$, $L \subseteq V_T^*$, $\# \notin V_T$. Es sei ferner $\tau \subseteq (V_T \cup \{\#\})^* \times X^*$ eine rationale Transduktion, definiert durch den Transduktor $M = (Z, V_T \cup \{\#\}, X, \delta, z_0, Z_F)$.
  Nach Übung 4 können wir $\delta \subseteq Z \times (V_T \cup \{\lambda\}) \times X^* \times Z$ voraussetzen.
  Definiere eine reguläre Substitution $\sigma : V_T^* \to 2^{(Z \times (V_T \cup \{\#\}) \times X^* \times Z)^*}$ durch $a \mapsto \{ (z_1, \#, w, z_2) \mid (z_1, \lambda, w, z_2) \in \delta \}^* \{ (z_1, a, w, z_2) \mid (z_1, a, w, z_2) \in \delta \} \{ (z_1, \#, w, z_2) \mid (z_1, \lambda, w, z_2) \in \delta \}^*$.
  Der Homomorphismus $g : (V_T \cup \{\#\})^* \to V_T^*$ sei gegeben durch $a \mapsto a$ für $a \in V_T$ und $\# \mapsto \lambda$.
  Die Sprache $R = \{ (z_{i_0}, a_1, w_1, z_{i_1})(z_{i_1}, a_2, w_2, z_{i_2}) \cdots (z_{i_{m-1}}, a_m, w_m, z_m) \mid a_1, \ldots, a_m \in V_T \cup \{\#\} \wedge w_1, \ldots, w_m \in X^* \wedge \forall 1 \leq j \leq m (z_{i_{j-1}}, g(a_j), w_j, z_{i_j}) \in \delta \wedge z_{i_0} = z_0 \wedge z_{i_m} \in Z_F \}$ ist regulär.*
  Es sei der Homomorphismus $h : (Z \times (V_T \cup \{\lambda\}) \times X^* \times Z)^* \to X^*$ gegeben durch $(z, a, w, z') \mapsto w$.
  Nach Voraussetzung liegt $h(\sigma(L) \cap R) = \tau(L)$ in $\mathcal{A}$. $\square$

*siehe entsprechende Konstruktion im Satz von Nivat

Noch mehr Arbeit...

<u>**Übung**</u> **9** *Zeigen Sie, dass die Sprachfamilie* $\mathcal{A} =$ *TRIO*$(\{\mathrm{L}\})$ *vereinigungsstabil ist für jede Sprache* L.
Hinweis: Benutzen Sie Korollar 8.

<u>**Übung**</u> **10** *Zeigen Sie, dass jede schnittabgeschlossene AFL substitutionsabgeschlossen ist.*

<u>**Übung**</u> **11** *Ist jede FAFL gegen Mischen (Shuffle) abgeschlossen?*

Satz 15 *Jede Typ-0-Sprache ist das homomorphe Bild eines Schnitts zweier deterministisch kontextfreier Sprachen:* $\mathcal{L}_0 = \mathrm{H}(\mathrm{DCF} \wedge \mathrm{DCF})$.

Beweis: Es sei $L \in \mathcal{L}_0, L = L(\mathcal{M}), \mathcal{M}$ eine Turing-Maschine mit Bandalphabet Γ und Zustandsmenge $Q; L \subseteq V_T^*, V_T \subseteq \Gamma$, so daß $v \in L \iff$ es existiert eine akzeptierende Berechnung von $\mathcal{M}$ auf V.

Sei $K_0 \vdash^1_{\mathcal{M}} K_1 \vdash^1_{\mathcal{M}} \ldots \vdash^1_{\mathcal{M}} K_m$ Konfigurationsfolge mit
(1) $K_i \in \Gamma^* Q \Gamma^*$
(2) $K_0 = q_0 v, q_0$ Startzustand
(3) $K_m \in \Gamma^* Q_F \Gamma^*, Q_F$ Endzustände
o.B.d.A. m ungerade.

$L_0 := \{ K\$ \overleftarrow{K'} \mid K, K' \in \Gamma^* Q \Gamma^*, K \vdash^1_{\mathcal{M}} K' \}, L_0' := \{ \overleftarrow{K} \$K' \mid K, K' \in \Gamma^* Q \Gamma^*, K \vdash^1_{\mathcal{M}} K' \} \in \mathrm{DCF}$
$K = \alpha a q b \beta, \alpha, \beta \in \Gamma^*; a, b \in \Gamma; q \in Q$; aus $a = \lambda$ folgt $\alpha = \lambda$, und aus $b = \lambda$ folgt $\beta = \lambda$.
K' hat dann die Form $\alpha q' a b' \beta$ bei Linksbewegung $(q, b) \to (q', b', L)$;
entsprechend bei Rechtsbewegung.
Bei $a = \lambda$ und analog bei $b = \lambda$ muss explizit ein Blanksymbol eingeführt werden.

Dann gilt $v \in L \iff K_0 \vdash^1_{\mathcal{M}} K_1 \vdash^1_{\mathcal{M}} \ldots \vdash^1_{\mathcal{M}} K_m$ terminierende Konfigurationsfolge.

Erste Idee: Kodiere die Berechnung von $\mathcal{M}$ auf v in das Wort

$$w := \overbrace{K_0\$ \overleftarrow{K_1}}^{L_1} \underbrace{\$}_{L_2} \overbrace{K_2\$ \overleftarrow{K_3}}^{L_1} \underbrace{\$}_{L_2} \cdots \overbrace{K_{m-1}\$ \overleftarrow{K_m}}^{L_1}.$$

Dann liegt w im Schnitt der beiden DCF-Sprachen $L_1 := (L_0\$)^*L_0$ und $L_2 := \{ q_0 \}V_T^*\$(L_0'\$)^*\Gamma^*Q_F\Gamma^*$. Umgekehrt repräsentiert Jedes $q_0v\$x \in L_1 \cap L_2$ für ein $x \in Y^*$ mit $Y := \Gamma \cup Q \cup \{ \$ \}$ eine akzeptierende Berechnung von M auf v.

Zweite Idee: Schreibe in $K_0 = q_0v$ das v mit einem anderen Alphabet $\overline{V_T} := \{ \bar{a} \mid a \in V_T \}$. Bilde $w' := q_0\bar{v}\$ \overleftarrow{K_1} \$K_2 \ldots$ bzw. L_1' und L_2'.

Betrachte $h : (Y \cup \overline{V_T})^* \to V_T^* : \begin{cases} h(a) := \lambda \text{ ; } a \in Y \\ h(\overline{a}) := a \text{ ; } \overline{a} \in \overline{V_T} \end{cases}$

Dann gilt $h(w') = v$. Offenbar ist $h(L_1' \cap L_2') = L$.

Alles zusammen ergibt $\mathcal{L}_0 \subseteq H(\mathrm{DCF} \wedge \mathrm{DCF}) \subseteq H(\mathcal{L}_0 \wedge \mathcal{L}_0) \subseteq H(\mathcal{L}_0) \subseteq \mathcal{L}_0$. $\square$

Anmerkung 16 *In obiger Konstruktion gilt* $v \in L \iff L_1 \cap \underbrace{L_2 \cap \overbrace{\{q_0 v\$\}Y^*}^{\in REG}}_{\in DCF} \neq \emptyset$*.*

Also ist das Schnittleerheitsproblem für zwei (deterministische) kontextfreie Grammatiken ebenso wie das Wortproblem für Typ-0-Sprachen unentscheidbar.

Anmerkung 17 *Jede nichtdeterministische Realzeit-Sprache ist das Bild eines nicht-löschenden Homomorphismus vom Schnitt dreier kontextfreier Sprachen:*
$NTIME(n) = H_\lambda(CF \wedge CF \wedge CF)$*.*
($NTIME(n)$ *sind die in Realzeit erkennbaren Sprachen.)*