Vorlesung Baumautomaten (Mitschrift)

Benedikt Elßmann (3720358)
be57xocu@studserv.uni-leipzig.de

Universität Leipzig

13. Juni 2019

Inhaltsverzeichnis

0 Einleitung

Automaten lesen Wörter $w = a_1 \dots a_n$ und geben "accept" aus oder nicht.
Dafür gibt es Erweiterungen, wie etwa:

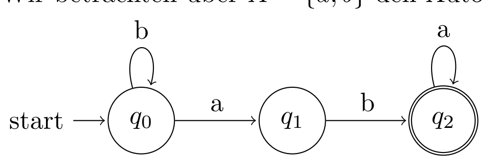
- gewichtete Automaten, das heißt der Output ist ein Semiringelement
- Automaten mit Gedächtnis (Stack)
- Automaten über anderen Strukturen
  - ω-Wörter $w = a_1 \dots a_n$
  - Graphen
  - Bäume
  - Kombinationen dieser

Typische Fragestellungen:

- Ausdrucksstärke
- Darstellung als rationale Ausdrücke (Kleene)
- Darstellung als Grammatik
- Darstellung als Logik

1 Bäume und Baumautomaten

Wir betrachten über $A = \{a, b\}$ den Automaten $\mathcal{A}$:

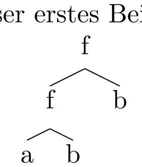


mit $L(\mathcal{A}) = b^*aba^*$.
Betrachtung des Wortes $w = baba \in L(\mathcal{A})$:

Der eindeutige erfolgreiche Lauf für w lässt sich darstellen als:

$q_0baba \to bq_0aba \to baq_1ba \to babq_2a \to babaq_2 \in F$ (Finalzustand)

Baumautomaten funktionieren analog. Unser erstes Beispiel wird



Akzeptiert mit dem Lauf:

$$f(f(a,b),b) \rightarrow f(f(q_a,q_b),q_a) \rightarrow f(q_a(q_a,q_b),q_a) \rightarrow q_f(q_a(q_a,q_b),q_a)$$

mit $q_f \in F$

1.1 Definition Rangalphabet

Ein paar (Σ, rk), wobei Σ eine endliche Menge von Symbolen und $rk : \Sigma \to \mathbb{N}$ eine Abbildung ist, heißt Rangalphabet.
Für $f \in \Sigma$ heißt $rk(f)$ der Rang (oder die Stelligkeit) von f.

Intuitiv: $rk(f)$ ist die Anzhal der Kinder von f in einem Baum.
Insbesondere ist die Anzahl der Kinder für jedes Symbol fest.

Gilt $rk(f) = n$, schreiben wir auch $f^{(n)}$ statt f. wir schreiben:

- 0-stellige Symbole (Konstanten) $a, b, \ldots$
- unär, binär, $\ldots f, g, \ldots$

Wir setzen $\Sigma^{(n)} = \{f \in \Sigma | rk(f) = n\}$

In $f(f(a,b),b)$ ist also $rk(f) = 2, rk(b) = 0$

1.2 Definition Term, Tree

Sei (Σ, rk) ein Rangalphabet. Die Menge T_Σ der Bäume üeber Σ ist induktiv definiert durch:

- $\Sigma^0 \subseteq T_\Sigma$
- $f^{(n)} \in \Sigma$. $t_1, \ldots, t_n \in T_\Sigma$, dann ist $f(t_1, \ldots, t_n) \in T_\Sigma$

Intuitiv sind $t_1, \ldots, t_n$ die Kinder von f.
Z.B. ist $f(f(a,b),b)$ der Term $f(f(a,b),b)$.

1.3 Definition Höhe

Sei (Σ, rk) ein Rangalphabet. Die Höhe ht ist gegeben durch:

- für $a^{(0)} \in \Sigma : ht(a) = 1$.
- für $f(t_1, \dots, t_n) \in T_\Sigma : ht(f) = 1 + max\{ht(t_i) | i \in \{i, \dots, n\}\}$

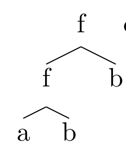
Ziel: Zugriff auf einen Knoten innterhalb eines Baumes und deren Label.
Dafür ordnen wir den Knoten Positionen zu. Das geht induktiv wie folgt:

1.4 Definition Position

Sei (Σ, rk) ein Rangalphabet. Die Positionenmenge ist definiert durch:

- für $a^{(0)} \in T_\Sigma$ ist $Pos(a) = \{\varepsilon\}$
- für $f(t_1, \dots, t_n) \in T_\Sigma$ ist $Pos(f(t_1, \dots, t_n)) = \{\varepsilon\} \cup 1 \cdot Pos(t_1) \cup \dots \cup n \cdot Pos(t_n)$

Beispiel:
Betrachtung von $f(f(a,b),b)$ bzw.



der Term $f(f(a,b),b)$

$Pos(f) = \{\varepsilon, 1, 2, 1.1, 1.2\}$

1.5 Definition der Label an den Positionen

Für einen Term der Form $t = f(t_1, \dots, t_n)$ ist das Symbol $t(p)$ in t an p-ter Position induktiv definert durch:

- $t(\varepsilon) = f$
- $t(ip) = t_i(p), i \in \{1, \dots, n\}$

Beispiel: Betrachtung von $f(f(a,b),b)$
Dann ist
$t(\varepsilon) = f$
$t(1) = t(1 \cdot \varepsilon) = t_1(\varepsilon) = f$
$t(2) = t(2 \cdot \varepsilon) = t_2(\varepsilon) = b$
$t(1.1) = t_1(1) = a$
$t(1.2) = t_2(1) = b$

1.6 Definition Sub-Baum

Für T_Σ ist ein Sub-Baum $t_{|p}$ an p-ter Position wie folgt definiert:

- $Pos(t_{|p}) = \{i | pi \ \in Pos(t)\}$

- $\forall q \in Pos(t_{|p}$ ist $t_{|p}(q) = t(pq)$

Wir schreiben $t[u]_p$ für den Baum, der entsteht, wenn man in t den sub-Baum $t_{|p}$ durch n ersetzt.

Beispiel: $f(f(a, b), b)$ bzw.

f
f b
a b

$t_{|1} = f(a, b)$

f
a b

$t_{|2} = t(1.2) = b$

$u = g(b, a)$

g , dann ist
b a

$t[u]_{|1} = f(g(b, a), b)$

f
g b
b a

1.7 Definition Baumautomat

Ein Baumautomat $\mathcal{A}$ ist ein 4-Tupel (Q, Σ, F, Δ), wobei:
$Q \ldots$ endliche Menge an Zusänden
$\Sigma \ldots$ Rangalphabet, wobei $\Sigma \cup Q \neq \emptyset$
$F \cdots \subseteq Q$ Finalzustände
$\Delta \ldots$ Menge von Regeln

$r : f(q_1 \ldots q_n) \to q$
für $q, q_1, \ldots, q_n \in Q$, für $a^{(0)} \in T_\Sigma : a \to q$

Beispiel:
$\mathcal{A} = \{\{q_a, q_b, q_f\}, \{a^{(0)}, b^{(0)}, f^{(2)}\}, \{q_f\}, \Delta\}$
mit $\Delta = \{a \to q_a, b \to q_b, f(q_a, q_b) \to q_a, f(q_a, q_b) \to q_f\}$

1.8 Definition Lauf/Run

Sei $\mathcal{A} = (Q, \Sigma, F, \Delta)$ ein Baumautomat und $t \in T_\Sigma$. Ein Lauf r für t von $\mathcal{A}$ ist ein Term mit

- $Pos(r) = Pos(t)$
- Ist $t(p) = a$ ein Blatt, dann ist $r(p) = q_a$, nur wenn $(a \to q_a) \in \Delta$
- Ist $t(p) = f^{(m)}$, dann ist $r(p) = q$, wenn $(f(q_1, \ldots, q_n) \to q) \in \Delta$ und $r(p_i) = q_i$, $i \in \{1, \ldots, n\}$

Ein Lauf ist erfolgreich, wenn $r(\varepsilon) \in F$. Der Automat $\mathcal{A}$ akzeptiert t, falls es einen erfolgreichen Lauf für t von $\mathcal{A}$ gibt.
Wir bezeichnen mit $L(\mathcal{A}) = \{t \in T_\Sigma | \mathcal{A} \text{ akzeptiert } t\}$ die von $\mathcal{A}$ erkannte Baumsprache. Eine Sprache $L \subseteq T_\Sigma$ heißt erkennbar, falls ein Baumautomat $\mathcal{A}$ existiert mit $L = L(\mathcal{A})$.

Um einzelne Schritte von Baumautomaten zu formalisieren, betrachten wir die *move relation* $\to_\mathcal{A}$, definiert wie folgt:
Gegeben sei $\mathcal{A} = (Q, \Sigma, F, \Delta)$, dann ist $t \to_\mathcal{A} t'$ mit $t, t' \in T_{\Sigma \cup Q}$, falls

- $t(p) = f^{(n)}$
- $t(pi) = q_i$ für $i \in \{1, \ldots, n\}$ und p_i sind Blätter
- $(f(q_1, \ldots, q_n) \to q) \in \Delta$
- und $t' = t[q]_p$

Mit $\to_\mathcal{A}^*$ bezeichnen wir die transitive Hülle von $\to_\mathcal{A}$.

1.9 Lemma

Sei $\mathcal{A} = (Q, \Sigma, F, \Delta)$ ein Baumautomat. Dann ist $L(\mathcal{A}) = \{t \in T_\Sigma | t \to_\mathcal{A}^* q \text{ mit } q \in F\} (= Z)$

Beweis: „$L(\mathcal{A}) \subseteq Z$“:

Wir zeigen: Es existiert ein Run r für t von $\mathcal{A}$ mit $r(\varepsilon) = q$, dann ist $t \to_\mathcal{A}^* q$

Inuktionsannahme:
$t = a^{(0)} \in T_\Sigma$. Dann gilt $a \in L(\mathcal{A})$, falls ein Lauf r existiert mit $r(a) = q_a$ und $(a \to q_a) \in \Delta$. Dann folgt $a \to_\mathcal{A}^* q_a$.
Sei nun $t = f(t_1, \ldots, t_n)$

Induktionsvoraussetzung:
Falls für $t_1, \ldots, t_n$ Läufe r_i existieren mit $r_i(\varepsilon) = q_i$, dann gilt auch $t_i \to_\mathcal{A}^* q_i$ mit $i \in \{1, \ldots, n\}$

Induktionsschritt:
zu zeigen: Es existiert ein Lauf r für t mit $r(\varepsilon) = q$, dann $t \to_\mathcal{A}^* q$.
Sei also r ein Lauf mit $r(\varepsilon) = q$. Dann ist $r(i) = q_i$, $i \in \{1, \ldots, n\}$, mit $(f(q_1, \ldots, q_n) \to q) \in \Delta$.
Laut Induktionsvoraussetzung gilt nun, $t_i \to_\mathcal{A}^* q_i$, $i \in \{1, \ldots, n\}$.
Damit $t = f(t_1, \ldots, t_n) \to_\mathcal{A}^* f(q_1, t_2, \ldots, t_n) \to_\mathcal{A}^* \cdots \to_\mathcal{A}^* f(q_1, \ldots, q_n)$
Des weiteren haben wir die regel $f(q_1, \ldots, q_n) \to q$, das heißt $f(q_1, \ldots, q_n) \to_\mathcal{A}^* q$.

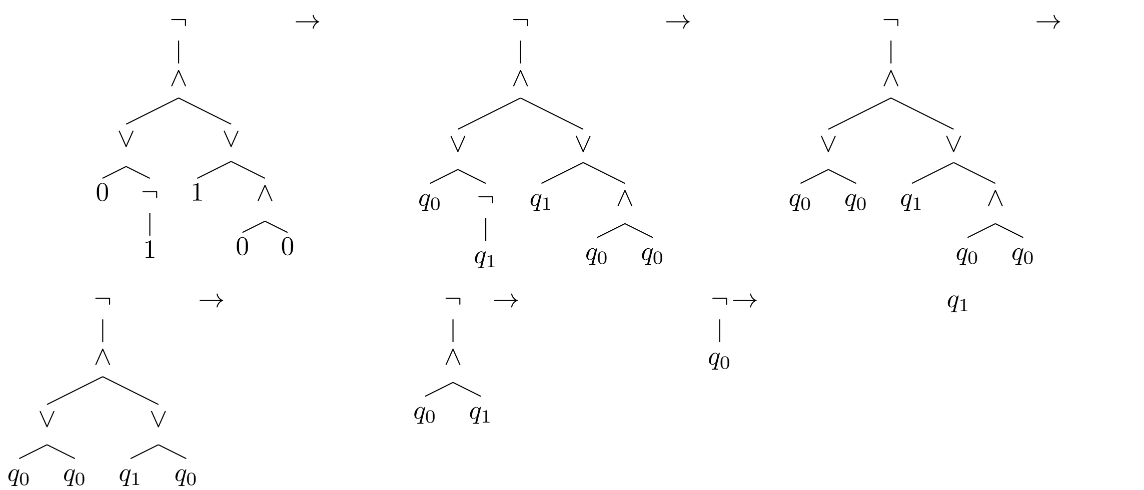
Insgesamt also $t \to_\mathcal{A}^* q$

Beweis: „$L(Z \subseteq \mathcal{A})$“: analog

Einige Beispiele für Baumautomaten:

1. Sei $B = (\{q_0, q_1\}, \{0^{(0)}, 1^{(0)}, \neg^{(1)}, \wedge^{(2)}, \vee^{(2)}\ \{q_1\}, \Delta\}$ mit
$\Delta = \{0 \to q_0, 1 \to q_1,$
$\neg(q_0) \to q_1, \neg(q_1) \to q_0,$
$\wedge\,(q_0, q_0) \to q_0, \wedge(q_0, q_1) \to q_0, \wedge(q_1, q_0) \to q_0, \wedge(q_1, q_1) \to q_1$
$\vee\,(q_0, q_0) \to q_0, \vee(q_0, q_1) \to q_1, \vee(q_1, q_0) \to q_1, \vee(q_1, q_1) \to q_1\}$

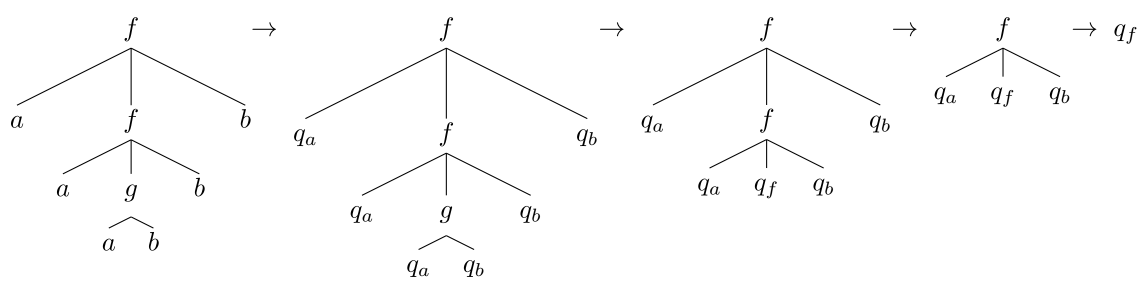
Beispiellauf:



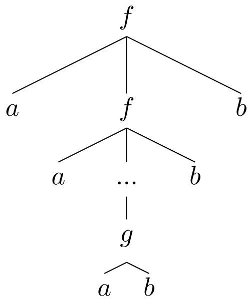
2. $(a^n b^n light)$

Betrachten $\mathcal{A} = (\{q_a, q_b, q_f\}, \{a^{(0)}, b^{(0)}, f^{(3)}, g^{(2)}\}, \{q_f\}, \Delta)$
mit $\Delta =$
$\{a \to q_a, b \to q_b, g(q_a, q_b) \to q_f, f(q_a, q_f, q_b) \to q_f\}$

Beispiellauf:



$\mathcal{A}$ akzeptiert also alle Bäume der Form:



3. Simulation eines Wortautomaten: (siehe Übung)

Betrachtet man $\Sigma = \{a^{(0)}, f^{(2)}, g^{(1)}\}$. Dann ist $L = \{f(g^i(a), g^i(a)) | i \geq 0\}$ nicht erkennbar.

1.10 Definition Determinismus

Ein Automat $\mathcal{A}(Q, \Sigma, F, \Delta)$ heißt deterministisch, falls: aus $f(q_1, \ldots, q_n) \to q$ und $f(q_1, \ldots, q_n) \to q'$ folgt $q = q'$

1.11 Satz

Sei $\mathcal{A} = (Q, \Sigma, F, \Delta)$ ein Baumautomat, dann existiert ein deterministischer Baumautomat $\mathcal{A}_d$, so dass $L(\mathcal{A}) = L(\mathcal{A}_d)$.

Beweis: Setze $\mathcal{A}_d = (Q_d, \Sigma, F_d, \Delta_d)$
mit $Q_d = 2^Q$ $(*)$
und $f(s_1, \ldots, s_n) \to s \in \Delta_d$
$\Leftrightarrow s = \{q \in Q | \exists q_1 \in s_1 \ldots q_n \in s_n : (f(q_1, \ldots, q_n) \to q) \in \Delta\}$
und $F_d = \{s \in Q_d | s \cap F \neq \emptyset\}$.

Wir zeigen:
1. $\mathcal{A}$ ist deterministisch
2. $L(\mathcal{A}) \subset L(\mathcal{A}_d)$
3. $L(\mathcal{A}_d) \subset L(\mathcal{A})$

1. ist klar, denn $(*)$ ist mit einer Äquivalenz definiert.
2. „$L(\mathcal{A}) \subseteq L(\mathcal{A}_d)$“:

Wir zeigen hierzu: Ist $Z = \{q | t \to_{\mathcal{A}}^* q\}$, dann $t \to_{\mathcal{A}_d}^* z$.

Induktionsannahme:
Angenommen $a \to_{\mathcal{A}} q_a$, dann ist $q_a \in \{q \in Q | q \to_{\mathcal{A}}^* q\}$, das heißt

$a \rightarrow^*_{\mathcal{A}} q_a \Leftrightarrow q_a \in \{q \in Q | a \rightarrow^*_{\mathcal{A}} q\}$
$\Leftrightarrow q_a \in \{q \in Q | (a \rightarrow q) \in \Delta\}$
also $a \rightarrow^*_{\mathcal{A}} q_a \Leftrightarrow q_a \in \{q \in Q | (a \rightarrow q) \in \Delta\}$, das heißt
$z := \{q_a \in Q | a \rightarrow^*_{\mathcal{A}} q_a\} = \{q \in Q | (a \rightarrow q) \in \Delta\} =: s$

Nun ist $(a \rightarrow s) \in \Delta_d$ per Definition, also auch $(a \rightarrow z) \in \Delta_d$, damit: $a \rightarrow^*_{\mathcal{A}_d} z$.

Betrachten wir nun $t = \sigma(t_1, \ldots, t_n)$

Induktionsvoraussetzung:
$t_i \rightarrow^*_{\mathcal{A}_d} z_i$ mit $Z_i = \{q \in Q | t_i \rightarrow^*_{\mathcal{A}} q\}$
Das heißt, es existieren Läufe r_i für t_i von $\mathcal{A}_d$ mit $r_i(\varepsilon) = z_i$

Induktionsschritt:
zu zeigen: $t \rightarrow^*_{\mathcal{A}} z$ mit $Z = \{q \in Q | t \rightarrow^*_{\mathcal{A}} q\}$
Das heißt, es existiert ein Lauf r für t von $\mathcal{A}_d$ mit $r(\varepsilon) = z$
Das heißt, $\exists r$:

- $r(\varepsilon) = z$
- $r(i) = z_i$
- $\sigma(z_1, \ldots, z_n) \rightarrow z \in \Delta_d$

Setze nun $r_{|i} = r_i$, damit ist insbesondere $r(i) = r_i(\varepsilon) = z := \{q | t_i \rightarrow^*_{\mathcal{A}_d} q\}$

Es bleibt also zu zeigen: $\exists$ Regel $\sigma(z_i, \ldots, z_m) \rightarrow z \in \Delta_d$.
Es ist nun $z \in Z \Leftrightarrow t \rightarrow^*_{\mathcal{A}} z$
$\Leftrightarrow \exists q_i \in Q : t_i \rightarrow^*_{\mathcal{A}} q_i, \sigma(q_1, \ldots, q_m) \rightarrow z \in \Delta$
$\Leftrightarrow \exists z_i \in Z_i$ und $\sigma/z_1, \ldots, z_m) \rightarrow z \in \Delta$
Also $Z = \{z \in Q | \exists z_i \in Z_i : (\sigma/z_1, \ldots, z_m) \rightarrow z) \in \Delta)$ also per Definition $\sigma/z_1, \ldots, z_m) \rightarrow z \in \Delta_d$

2. „$L(\mathcal{A}_d) \subseteq L(\mathcal{A})$“:

Sei $t \in T_\Sigma$ mit $t \notin L(\mathcal{A})$, dann ist $Z \cap F = \{q \in Q | t \rightarrow^*_{\mathcal{A}} q\} \cap F = \emptyset$
Laut 2. ist $t \rightarrow^*_{\mathcal{A}_d} z$ (und $\mathcal{A}$ ist deterministisch) Wegen $Z \cap F = \emptyset$ ist $Z \notin F_d$, also $t \notin L(\mathcal{A}_d)$

Wir vereinbaren die Abkürzungen: NBA/NTA für nichtdeterministischer Baumautomat und DBA/DTA für deterministischer Baumautomat.
Wie im Wortfall ist die Konstruktion exponentiell, das heißt wir benötigen expontntiell viele Zustände ($Q_d = 2^{|Q|}$). Und wie im Wortfall lässt sich das im Allgemeinen nicht vermeiden.

Beispiel: Betrachtet man $\Sigma = \{f^{(1)}, g^{(1)}, a^{(0)}\}$ und sei $L_n = \{f \in T_\Sigma | t(\underbrace{1 \ldots 1}_{\text{n-mal}}) = f\}$

Ein NTA benötigt $n+2$ Zustände:
$\mathcal{A} = (Q, \Sigma, F, \Delta)$ mit $Q = \{q, q_1, \ldots, q_{n+1}\}$, $F = \{q_{n+1}\}$
mit Übergängen $\Delta = \{a \to q, f(q) \to q, g(q) \to q, f(q) \to q_1,$
$f(q_i) \to q_{i+1}, g(q_i) \to q_{i+1}\}$ für $i \in \{1, \ldots, n\}$

Man kann zeigen: Ein DTA $\mathcal{A}'$ mit $L(\mathcal{A}') = L_n$ hat mindestens 2^{n+1} Zustände.

1.12 Definition vollständing und reduziert

Ein Automat $(\mathcal{A} = Q, \Sigma, F, \Delta)$ heißt:

- vollständig, falls für jedes $f^{(n)} \in \Sigma$ und alle $q_1, \ldots, q_n \in Q$ eine Regel $f(q_1, \ldots, q_n) \to q \in \Delta$ existiert.
- reduziert, falls für jeden Zustand $q \in Q$ ein Term $t \in T_\Sigma$ exisitert mit $f \to_{\mathcal{A}}^* q$

1.13 Satz

Sei $\mathcal{A}$ ein Baumautomat. Dann existiert ein vollständiger, reduzierter Baumautomat $\mathcal{A}'$ mit $L(\mathcal{A}) = L(\mathcal{A}')$.

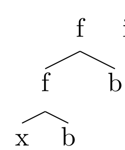
Für Wortautomaten gibt es das Pumping-Lemma, das die Gedächtnislosigkeit der Automaten formalisiert. Formal besagt es: Ist L eine reguläre Wortsprache, dann existiert ein $n \in \mathbb{N}$, so dass sich $w \in L$ mit $|w| > n$ zerlegen lässt in $w = xyz$, $y \neq \varepsilon$ und $\forall i \geq 0$ ist $xy^iz \in L$.

Baumautomaten haben auch kein Gedächtnis, also erwarten wir ein analoges Resultat. Dazu müssen wir formalisierten, was „aufgepumpt “werden soll.

1.14 Definition Kontext

Es sei Σ ein Rangalphabet und $x^{(0)} \notin \Sigma$. Es sei $C \in T_{\Sigma \cup \{x\}}$. Falls es genau eine Position $p \in Pos(C)$ gibt mit $C(p) = x$, dann heißt C ein Kontext.

Beispiel:

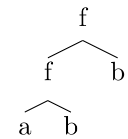


ist ein Kontext.

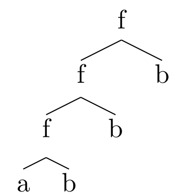
Wir schreiben $T_\Sigma(x)$ für die Menge aller solcher Kontexte.

Ist $C \in T_\Sigma(x)$ mit $C(p) = x$, dann schreiben wir $C[u]$ statt $C[u]_p$ für den Baum, der entsteht, wenn wir x durch u ersetzen.
Wir schreiben $C^0 = x$, $C^1 = C$, $C^n = C^{n-1}[C]$

Beispiel: Betrachtet $t =$



Setze $u = f(a,b)$ und $C = f(x,b)$.
Dann ist $t = C[u]$ und $C^2[u] =$

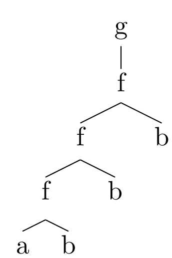


1.15 Pumping-Lemma

Sei $L \subseteq t_\Sigma$ erkennbar, dann existiert ein $k \in \mathbb{N}$, so dass:
Für alle $T \in L$ mit $ht(t) > k$ gibt es einen Kontext $C \in T_\Sigma(x)$, einen nicht-trivialen Kontext $C' \in T_\Sigma(x)$ und einen Term $u \in T_\Sigma$ mit $t = C[C'[u]]$ und $C[(C')^n[u]] \in L$ für alle $n \geq 0$.

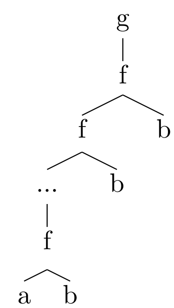
Beweis: Sei L erkennbar, das heißt $\exists$ Baumautomat $\mathcal{A} = (Q, \Sigma, F, \Delta)$ mit $L = L(\mathcal{A})$. Setze $|Q| = k$ und betrachte $t \in L$ mit $ht(t) > k$. Betrachte nun einen Lauf r und einen Pfad in t, der länger als k ist. Nun gibt es $p_1, p_2 \in Pos(r)$ mit $r(p_1) = r(p_2) = q \in Q$. Sei nun $u = t_{|p_2}$ der Sub-Baum von t bei p_2 und $u' = t_{|p_1}$. Dann existiert C' mit $C'[u] = u'$ und es existiert C mit $t = C[C'[u]]$. Es ist wegen $t \in L$
$C[C'[u]] \rightarrow^*_{\mathcal{A}} C[C'[q]] \rightarrow^*_{\mathcal{A}} C[q] \rightarrow^*_{\mathcal{A}} q_f \in F$, also auch
$C[(C')^n[u]] \rightarrow^*_{\mathcal{A}} C[(C')^n[q]] \rightarrow^*_{\mathcal{A}} CC[(C')^{(n-1)}[q]] \rightarrow^*_{\mathcal{A}} \cdots \rightarrow^*_{\mathcal{A}} C[q] \rightarrow^*_{\mathcal{A}} q_f \in F$. q.e.d.

Beispiel: Betrachte den Baumautomaten $\mathcal{A} = (\{q_a, q_b, q_g, q_f\}, \{a^{(0)}, b^{(0)}, g^{(1)}, f^{(2)}\}, \{q_f\}, \Delta)$ mit
$\Delta = \{a \rightarrow q_a, b \rightarrow q_b, f(q_a, q_b) \rightarrow q_g, g(q_g) \rightarrow q_f\}$



$u = f(a,b)$, $u' = C'[u] = f(f(a,b),b)$
$C = g(f(x,b))$, $C' = f(x,b)$

$C[(C')^n[u]] =$



Die Sprache $L = \{f(g^i(a), g^i(a)) | i \geq 0\}$ kann nicht erkennbar sein, denn für große i würde man ein k finden, so dass ein gegebener Baumautomat auch $f(g^{i+lk}(a), g^i(a))$ für alle $l \geq 0$ akzeptiert.

1.16 Korollar

Für $\mathcal{A} = (Q, \Sigma, F, \Delta)$ ist $L(\mathcal{A}) \neq \emptyset \Leftrightarrow \exists t \in L$ mit $ht(t) \leq |Q|$:

- $L(\mathcal{A})|$ nicht endlich $\Leftrightarrow \exists t \in L$ mit $|Q| < ht(t) \leq 2|Q|$

1.17 Abschlusseigenschaften

Erkennbare Sprachen sind abgeschlossen unter Vereinigung, Schnitt und Komplement. Das heißt, sind L_1 und L_2 erkennbar, dann auch $L_1 \cup L_2$, $L_1 \cap L_2$ und L_1^c (in T_Σ).

Beweis:
Seien $\mathcal{A}_1$ und $\mathcal{A}_2$ vollständige DTA. Betrachte für die Vereinigung
$\mathcal{A}_\cup = (Q_1 \times Q_2, \Sigma, F_1 \times Q_2 \cup Q_1 \times F_2, \Delta_1 \times \Delta_2)$ mit
$\Delta_1 \times \Delta_2 = \{f((q_1, q_1'), \ldots, (q_n, q_n')) \to (q, q') | f(q_1, \ldots, q_n) \to q \in \Delta_1, f(q_1', \ldots, q_n') \to q' \in \Delta_2\}$
Dann akzeptiert $\mathcal{A}_\cup$ die Sprache $L(\mathcal{A}_1) \cup L(\mathcal{A}_2)$.

Für $L(\mathcal{A}_1) \cap L(\mathcal{A}_2)$ betrachte den Automaten
$\mathcal{A}_\cap = (Q_1 \times Q_2, \Sigma, F_1 \times F_2, \Delta_1 \times \Delta_2)$

Für T_Σ $L(\mathcal{A}_1) = L(A_1)^c$ betrachte
$\mathcal{A}_C = (Q_1, \Sigma, Q_1 \ F_1, \Delta_1)$.
Der Automat A_C akzeptiert $L(A_1)^c$.

Beispiel:

Betrachte $\Sigma = \{a^{(0)}, g^{(1)}, f^{(2)}\}$ und $L = \{f(g^i(a), g^j(a)) | i \leq j\}$
Dann ist L nicht erkennbar, denn: Wäre L erkennbar, dann auch L' mit $L' = \{f(g^i(a), g^j(a)) | i \geq j\}$, also auch $L \cap L'$ ↯.

Bemerkung: Wenn $\mathcal{A}$ deterministisch und vollständig ist, dann können wir eine Übergangsfunktion $\delta : T_\Sigma \to Q$ definieren mit $\delta(t) = q$, falls $t \to_{\mathcal{A}}^* q$.

Wiederholung - Äquivalenzrelation:

Eine Äquivalenzrelation $\sim$ auf einer Menge M ist eine Relation mit

- $\forall m \in M : m \sim m$
- $\forall m, n \in M : m \sim n \Rightarrow n \sim m$
- $\forall l, m, n \in M : l \sim m, m \sim n \Rightarrow l \sim n$

Insbesondere: Ist $\sim$ eine Äquivalenzrelation auf M, so induziert $\sim$ eine Partition auf und umgekehrt, das heißt Mengen $(M_i)_{i\in I}$ mit $M_i \cup M_j = \emptyset$ für $i \neq j$ und $M = \bigcup\limits_{i\in I} M_i$

1.18 Definition Kongruenz

Eine Äquivalenzrelation $\equiv$ auf T_Σ heißt Kongruenz, falls für alle $f^{(n)} \in \Sigma$:

$v_1 \equiv u_1, \dots, v_n \equiv u_n \Rightarrow f(v_1, \dots, v_n) \equiv f(u_1, \dots, u_n)$.
Beispiel:
Die Relation $t \equiv t'$, falls t und t' die gleiche Anzahl Blätter modulo 2 haben.

- Außerdem: $t \equiv t' \Leftrightarrow ht(t) = ht(t')$
- Nicht: gleiche Höhe modulo 2

1.19 Definition

Eine Kongruenz $\equiv$ hat endlichen Index, falls $\equiv$ endlich viele Äquivalenzklassen indiziert.

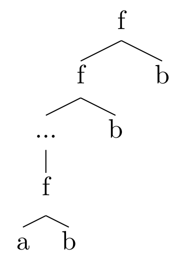
1.20 Lemma

Sei Σ ein Rangalphabet. Dann ist $\equiv$ genau dann eine Konguenz auf T_Σ, wenn $\equiv$ eine Äquivalenzrelation ist mit $u \equiv v \Rightarrow C[u] \equiv C[v]$ für alle Kontexte.

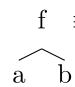
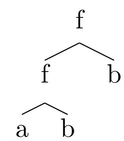
Beweis:
„$\Rightarrow$“Induktion:
Induktionsannahme: $C = x$, dann ist $u \equiv v \Rightarrow C[u] \equiv C[v]$ klar.
Sei nun $C = f(C_1, \dots C_n)$. Sei $x = C[ip] = C_i[p]$.
Dann ist $C[u]_{ip} = f(C_1, \dots, C_{i-1}, C_i[p], C_{i+1}, \dots, C_n) = C[u]_{ip} = f(C_1, \dots, C_{i-1}, C_i[v], C_{i+1}, \dots, C_n)$

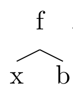
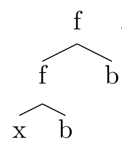
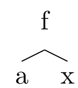
„$\Leftarrow$ “: Angenommen $u \equiv v$ und $C[u] \equiv C[v]$ für alle Kontexte.
Sei $f^{(n)} \in \Sigma$. Dann ist:
$f(u_1, \dots u_n)$
$= C^1[u_1] \equiv C^1[v_1] = f(v_1, u_2, \dots, u_n)$
$= \dots$
$= C^1[u_n] \equiv C^1[v_n] = f(v_1, \dots, v_n)$
Betrachte nun eine Sprache $L \subseteq T_\Sigma$ von Bäumen. Wir definieren $\equiv_L$ als:
$u \equiv_L v \Leftrightarrow \forall C \in T_\Sigma(x) : C[u] \in L \Leftrightarrow C[v] \in L$.

Beispiel: Betrachte alle Bäume der Form



$L = \{f(f(\ldots f(a,b),b)\ldots,b)\}$

Dann gilt:  $\equiv$ 

$C =$ , $C' =$ , $C'' =$ 

1.21 Theorem (Myhill-Nerode)

Die folgenden Aussagen sind äquivalent:

a) L ist erkennbar
b) L ist die Vereinigung von Äquivalenzklassen einer Kongruenz mit endlichem Index
c) $\equiv_L$ hat endlichen Index

Beweis:
„a $\Rightarrow$ b “: Sei $\mathcal{A}$ vollständiger DTA mit $L(\mathcal{A}) = L$. Sei $\mathcal{A} = (Q, \Sigma, F, \Delta)$.
Definiere $u \equiv_{\mathcal{A}} v \Leftrightarrow \delta(u) = \delta(v)$.
Offensichtlich hat $\equiv_{\mathcal{A}}$ höchstens $|Q|$-viele Äquivalenzklassen. Außerdem ist $\equiv_{\mathcal{A}}$ eine Kongruenz. Nun ist L Vereinigung aller Klassen $[u]_{\equiv_{\mathcal{A}}}$ mit $\delta(u) \in F$.

„b $\Rightarrow$ c “: Sei $\sim$ eine Kongruenz mit dnlichem Index. Sei $u \sim v$. Wegen Lemma 1.20 gilt $C[u] \sim C[v] \forall C \in T_\Sigma(x)$. Nun ist aer L die Vereinigung von Äquivalenzklassen von $\sim$, das heißt $C[u] \in L \Leftrightarrow C[v] \in L$. Insbesondere ist also $u \equiv_L v$
Wir haben gezeigt: $v \in [u]_\sim \Rightarrow v \in [u]_{\equiv_L}$, also $[u]_\sim \leq [u]_{\equiv_L}$
(Also ist $\sim$ eine Verfeinerung von $\equiv_L$)
Insbesondere hat $\equiv_L$ kleinern Index als $\sim$, also endlichen.

„c $\Rightarrow$ a “: Die Zustände $Q_{\min}$ sind die Äquivalenzklassen bezüglich $\equiv_L$. (Damit ist $Q_{\min}$ endlich).
Wir definieren Regeln
$f([u_1], \ldots, [u_n]) \to [f(u_1, \ldots, u_n)]$.
Das ist wohldefiniert, weil $\equiv_L$ eine Kompetenz ist. Finalzustände $F_{\min}$ sind $\{[u]_{\equiv_L} | u \in L\}$.

Dann akzeptiert $\mathcal{A}_{\min} = (Q_{\min}, \Sigma, F_{\min}, \Delta_{\min})$ die Sprache L.

Beispeiel:
Betrachte $\Sigma = \{a^{(0)}, g^{(1)}, f^{(2)}\}$ und
$L = \{f(g^i(a), g^i(a)) | i \geq 0\}$

Betrachte $g^i(a)$ und $g^j(a)$ mit $i \neq j$. Dann ist $C^i = f(x, g^i(a))$ ein Kontext mit $C^i[g^i(a)] \in L$, aber $C^i[g^j(a)] \notin L$. Da es unenlich viele $g^i(a)$ gibt, hat die Kongruenz bezüglich L unendlichen Index, also ist L nicht erkennbar.

1.22 Korollar

Ist L erkennbar, gibt es einen bis auf Umbenennung der Zustände eindeutigen, vollständigen DBA $\mathcal{A}$ mit $L = L(\mathcal{A})$. Dieser ist $\mathcal{A}_{\min}$ aus obigem Beweis.
Beweis:
Sei $L = L(\mathcal{A})$. Vorher gesehen:
$\equiv_{\mathcal{A}}$ ist Verfeinerung von $\equiv_L$
Also ist $|Q| \geq |Q_{\min}|$. Wir nehmen OBDA an: beide reduziert. Sei nun $qinQ$. Getrachte ein $u \in T_\Sigma$ mit $\delta(u) = q$. Betrachte die Funktion $\rho : Q \to Q_{\min}$ mit $\delta(u) = q \mapsto \delta_{\min}(u)$
Die Abbildung ρ ist wohldefiniert, denn falls $\delta(u) = \delta(v)$, dann $u \equiv_{\mathcal{A}} v \Rightarrow u \equiv_L v \Leftrightarrow \delta_{\min}(u) = \delta_{\min}(v)$. Außerdem ist ρ surjektiv, denn $\delta_{\min}(u)$ hat das Urbild $\delta(u)$.
Also: $|Q| = |Q_{\min}| \Rightarrow \rho$ ist Bijektion. $\Box$

1.23 Einschub - Homomorphismen von Baumsprachen

1.23.1 Allgemeine Homomorphismen

$(M, \cdot), (N, *); h : M \to N$ heißt Homomorphismus, falls
$\forall m, \hat{m} : h(m \cdot \hat{m}) = h(m) * h(\hat{m})$

1.23.2 Worthomomorphismen

$(A^*, \cdot), (B^*, \cdot); h : A^* \to B^*$ ist Homomorphismus, falls $h(w \cdot \hat{w}) = h(w) \cdot h(\hat{w})$. (zusätzlich $h(\varepsilon) = \varepsilon$)

Nutze für die Definition des Homomorphismus die induktive Definition von Wörtern aus A^*.
0.) $\varepsilon \in A^*$
1.) $a \in A^*$ $\forall a \in A$
2.) $a \cdot w \in A^*$ $\forall a \in A, w \in A^*$

Ein Wort-Homomorphismus entsteht deshalb aus einer Abbildung $\bar{h} : A \to B$ wie folgt:
0.) $h(\varepsilon) = \varepsilon$
1.) $h(a) = \bar{h}(a)$ $\forall a \in A$
2.) $h(a \cdot w) = h(a) \cdot h(w) = \bar{h}(a) \cdot h(w)$ $\forall a \in A, w \in A^*$

1.23.3 Baumhomomorphismen

T_Σ :
1.) $a \in T_\Sigma \ \forall a^{(0)} \in \Sigma$
2.) $f(t_1, \ldots, t_n) \in T_\Sigma \ \forall f^{(n)} \in T_\Sigma$

Zunächst: Schreibe $\Sigma = \bigcup\limits_{n=0}^{r} \Sigma^{(n)}$, wobei $\Sigma^{(n)} = rk^{-1}(n)$.
Sei $X = \{x_1, \ldots, x_n\}$ und $X_n = \{x_1, \ldots, x_n\}, X_0 = \emptyset$

Dann ist für ein Rangalphabet Γ auch $\Gamma \cup X_n$ ein Rangalphabet mit $rk(x_i) = 0$

Nachtrag - Substitution:
$t, s \in T_{\Sigma \cup X_n}$ (für ein $n \in \mathbb{N}$)
Sei $P \subseteq pos(t), P = \{p \in pos(t) | t(p) = x_i\}$ für ein $i \in \{1, \ldots, n\}$
Etwa $P = \{p_1, \ldots, p_m\}$
Dann ist $t_{[x_i \leftarrow s]} = t[s]_{p_1} \cdot \cdots \cdot t[s]_{p_m}$.

Definition von Homomorphismen für jeden Rang $n = 0, \ldots, r$
Wähle eine Funktion $\bar{h}_n : \Sigma^{(n)} \to T_{\Gamma \cup X_n}$
Der von $\bar{h}_0, \ldots, \bar{h}_r$ erzeugte Homomorphismus $h : T_\Sigma \to T_\Gamma$

1.) $h(a) = \bar{h}_0(a) \ \forall a \in \Sigma$
2.) $h(f(t_1, \ldots, t_n)) = \bar{h}_n(f)[x_1 \leftarrow h(t_1)] \ldots [x_n \leftarrow h(t_n)]$

Beispiel:

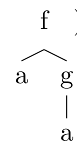
$\Sigma = \{a^{(0)}, g^{(1)}, f^{(2)}\}$
$\Gamma = \{\alpha^{(0)}, \delta^{(2)}, \tau^{(2)}\}$

$\bar{h}_0 : a \to \tau(\alpha, \alpha)$; $\bar{h}_1 : g \to \rho(\rho(\alpha, x_1), x_1)$; $\bar{h}_2 : f \to \tau(\tau(x_2, \alpha), x_2)$

Gegenbeispiel:

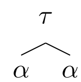
$\bar{h}_0 : a \to \tau(\alpha, x_1)$; $\bar{h}_1 : g \to \rho(\rho(\alpha, x_2), x_1)$; $\bar{h}_2 : f \to \tau(x_5, \alpha)$

Erzeugter Homomorphismus $h : T_\Sigma \to T_\Gamma$ wie oben

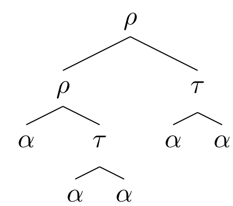
$h($ 

$)$

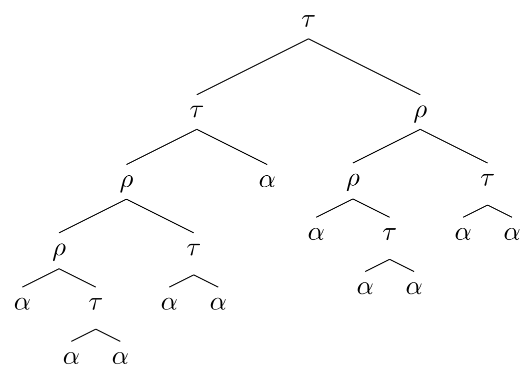
$h(a) = \bar{h}_0(a) =$



$h(g(a)) = \bar{h}_1(g)[x_1 \leftarrow h(a)] =$



$h(f(a, g(a))) = \bar{h}_2(f)[x_1 \leftarrow h(a)][x_2 \leftarrow h(g(a))]$



1.23.4 lineare Terme

Ein Term $t \in T_{\Sigma \cup X_n}$ heißt linear, falls jede Variable höchstens einmal vorkommt,
d.h. falls $\forall i \in \{1, \dots, n\} : |\{p \in pos(t) | t(p) = x_i\}| \leq 1$

1.23.5 linearer Homomorphismus

Ein Homomorphismen $h : T_\Sigma \to T_\Gamma$ erzeugt von $\bar{h}_0, \dots, \bar{h}_r$ heißt linear, falls $\forall f^{(n)} \in T_\Sigma$ gilt: $\bar{h}_n(f) \in T_{\Gamma \cup X_n}$ ist linear.

Beispiel: im Allgemeinen erhalten Homomorphismen die Erkennbarkeit nicht.
Betrachte
$\Sigma = \{f^{(1)}, g^{(1)}, a^{(0)}\}, \Gamma = \{\hat{f}^{(2)}, g^{(1)}, a^{(0)}\}$
$\bar{h}_0 : a \to a, \bar{h}_1 : g \to g(x_1), \bar{h}_1 : f \to \hat{f}(x_1, x_1)$
Für $L \subseteq T_\Sigma, L = \{f(g^i(a)) | i \in \mathbb{N}\}$ ist $h(L) = \{\hat{f}(g^i(a), g^i(a)) | i \in \mathbb{N}\}$ nicht erkennbar, obwohl L erkennbar ist.

1.23.6 Satz

Sei $L \subseteq T_\Sigma$ erkennbar, $h : T_\Sigma \to T_\Gamma$ ein linearer Homomorphismen, dann ist $h(L) \subseteq T_\Gamma$ erkennbar.

Beweisskizze:
Sei $A = (Q, \Sigma, F, \Delta)$ ein reduzierter DFTA mit $L(A) = L$

- Seien $(\bar{h}_n)_{n=0}^r$ die erzeugenden Funktionen von h
- Definiere NFTA $A' = (Q', \Gamma, F', \Delta)$ wie folgt:
  für jede Regel $\rho : f(q_1, \dots, q_n) \to q \in \Delta :$
  Setze $Q^\rho = \{q_p^\rho | q \in pos(\bar{h}_n(f))\}$ und
  Δ^ρ für jedes $p \in pos(\bar{h}_n(f))$
  falls $(\bar{h}_n f)(p) = g^{(k)} \in \Gamma^{(k)}$ für ein k,
  $g(q_{p\cdot 1}^\rho, \dots, q_{p\cdot k}^\rho) \to q_p \in \Delta^\rho$
  falls $(\bar{h}_n f)(g) = x_i$ für ein i
  $q_i \to q_p^\rho \in \Delta^\rho$
  $q_\varepsilon^\rho \to q \in \Delta^\rho$
  OBdA Q^ρ paarweise disjukt auch mit Q

- $Q' = \bigcup\limits_{\rho \in \Delta} Q^\rho \cup Q$
- $\Delta = \bigcup\limits_{\rho \in \Delta} \Delta^\rho \cup Q$
- $F' = F$

1.23.7 Satz

Sei $h : T_\Sigma \to T_\Gamma$ beliebiger Homomorphismus und $L \subseteq T_\Gamma$ erkennbar. Dann ist auch $h^{-1}(L) \subseteq T_\Sigma$ erkennbar.

Beweis:
Sei $A' = (Q', \Gamma, F', \Delta')$ ein vollständiger DFTA mit $L(A') = L$.
Definiere $A = (Q, \Sigma, F, \Delta)$ wie folgt:
$Q = Q', F = F'$
$f(q_1, \dots, q_n) \to q \in \Delta$
$\Leftrightarrow \bar{h}_n(f)[x_1 \leftarrow q_1] \cdot \dots \cdot [x_n \leftarrow q_n] \to_{\mathcal{A}'}^* q$

Beweis der Korrektheit (strukturelle Induktion):
Zeige die stärkere Aussage: für $q \in Q = Q'$ gilt
$t \to_{\mathcal{A}}^* q \Leftrightarrow h(t) \to_{\mathcal{A}'}^* q$
$(h^{-1}(s) \to_{\mathcal{A}}^* q \Leftrightarrow s \to_{\mathcal{A}'}^* q)$

Induktionsannahme:
Sei $t = a \in \Sigma^{(0)}$. Dann gilt $h(a) \to_{\mathcal{A}'}^* q \Leftrightarrow \bar{h}_0(a) \to_{\mathcal{A}'}^* q \Leftrightarrow a \to q \in \Delta \Leftrightarrow a \to_{\mathcal{A}}^* q$

Induktionsvoraussetzung:
Die Aussage gelte für Terme mit Höhe $\leq k$

Induktionsschritt: Dann gilt sie auf für t mit Höhe $k+1$.
Sei $t = f(t_1, \ldots, t_n)$ mit $ht(t_i) \leq k \; \forall i \in \{1, \ldots, m\}$

$t \to_{\mathcal{A}}^* q \Leftrightarrow \exists q_1, \ldots, q_n : t_i \to_{\mathcal{A}}^* q_1$ und $f(q_1, \ldots, q_n) \to q \in \Delta$
$\Leftrightarrow \exists q_1, \ldots, q_n : h(t_i) \to_{\mathcal{A}'}^* q_i$ und $f(q_1, \ldots, q_n) \to q \in \Delta$
$\Leftrightarrow \exists q_1, \ldots, q_n : h(t_i) \to_{\mathcal{A}'}^* q_i$ und $\bar{h}_n(f)[x \leftarrow q_i] \cdot \cdots \cdot [x_n \leftarrow q_n] \to_{\mathcal{A}'}^* q$
$\Leftrightarrow \bar{h}_n(f)[x_1 \leftarrow h(t_1)] \cdot \cdots \cdot [x_n \leftarrow h(t_n)] \to_{\mathcal{A}'}^* q$
$\Leftrightarrow h(f(t_1, \ldots, t_n))(= h(t)) \to_{\mathcal{A}'}^* q$

1.24 Top-Down Baumautomaten

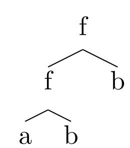
Bisher: Bottom-Up TA - laufen Bäume von den Blättern zu der Wurzel nach oben.
Nun: Top-Down TA - umgekehrt

Definition:
Ein nicht-deterministischer Top-Down Baumautomat ist ein Tupel $\mathcal{A} = (Q, \Sigma, I, \Delta)$, wobei:

- Q eine enliche Menge Zustände
- Σ ein Rangalphabet
- $I \subseteq Q$ Initialzustände
- Δ eine endliche Menge Regeln der Form
  $q(f(x_1, \ldots, x_n)) \to f(q_1(x_1), \ldots, q_n(x_n))$
  bzw. für $n = 0$: $q(a) \to a$

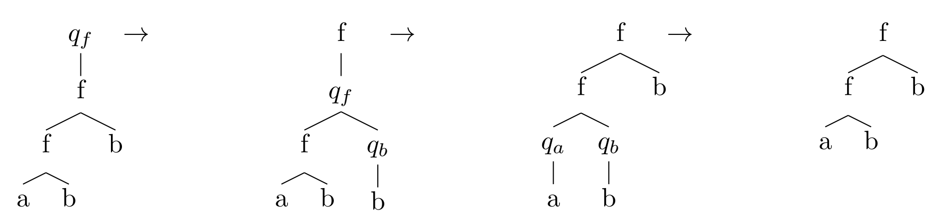
ist.

Beispiel:
Wir betrachten wieder $\Sigma = \{a^{(0)}, b^{(0)}, f^{(2)}\}$ und $t =$



Setze
$Q = \{q_f(f(x_1, x_2)) \to f(q_f(x_1), q_b(x_2)), q_f(f(x_1, x_2)) \to f(q_a(x_1), q_b(x_2)), q_a(a) \to a, q_b(b) \to b\}$

Run für t (intuitiv):



Betrachte folgende Übergangsrelation:
$C[q(f(t_1,\dots,t_n))] \to C[f(q_1(t_1),\dots,q_n(t_n))]$, $C \in T_{\Sigma\cup Q}(x)$,
falls $q(f(x_1,\dots,x_n)) \to f(q_1(x_1),\dots,q_n(x_n)) \in \Delta$

$\to^*$ transitive Hülle

$L(\mathcal{A}) = \{t \in T_\Sigma | q(t) \to^*, q \in I\}$

1.25 Satz

Eine Sprache $L \subseteq T_\Sigma$ ist genau dann erkennbar, wenn es einen nichtdeterministischen Top-Down TA $\mathcal{A}$ gibt mit $L = L(\mathcal{A})$.

Bemerkung:
Top-Down Automaten sind nicht determinisierbar. Deterministische Top-Down TA erkennen „path closed“ Sprachen.

2 Grammatiken

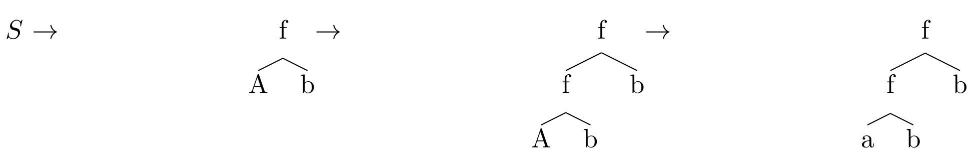
2.1 Definition - Grammatik

Eine Grammatik ist ein Tupel $G = (S, N, \Sigma, R)$. Dabei ist:

- S Startsymbol ($S = S^{(0)}, S \in N$)
- N (Rangalphabet) nichtterminale Symbole
- Σ (Rangalphabet) terminale Symbole ($\Sigma \cap N = \emptyset$)
- R Regeln der Form $\alpha \to \beta$ mit:
  $\alpha, \beta \in T_{\Sigma\cup N\cup X}$ ($X \cap (\Sigma \cup N) = \emptyset$ Variablen),
  α enthält mindestens ein Nichtterminal-Symbol

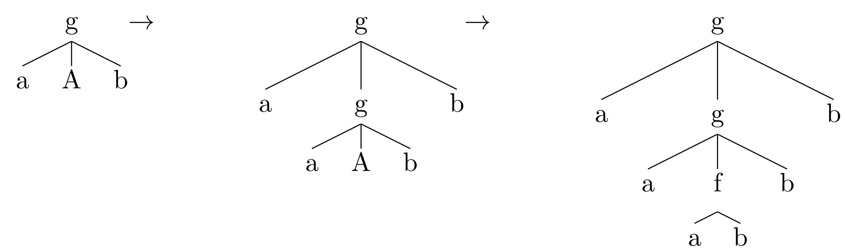
Eine reguläre Grammatik enthält nur Regelnd der Form $A \to B$, wobei A den Rang 0 hat und $b \in T_{\Sigma\cup N}$. Insbesondere enthält N nur Symbole mit Rang 0.

Beispiel: Betrachte $G = (S, \{S, A\}, \{a^{(0)}, b^{(0)}, f^{(2)}\}, R)$ mit
$R = \{S \to f(A, b), A \to a, A \to f(A, b)\}$
zum Beispiel haben wir:



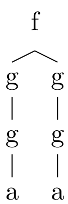
Beispiel: Betrachte $G = (S, \{S, A\}, \{a^{(0)}, g^{(3)}, f^{(2)}\}, R)$ mit
$R = \{S \to A, A \to f(a, b), A \to g(a, A, b)\}$
zum Beispiel:

$S \to A \to$



Bemerkung - kontextfreie Baumgrammatik:
$F(x_1, \ldots, x_n) \to t$, $t \in T_{\Sigma \cup N \cup \{x_1, \ldots, x_n\}}$
z.B. $S \to F(a, a), F(x, x) \to F(G(x), G(x)), F(x, x) \to f(x, x), G(x) \to g(x)$

Damit können wir erzeugen:
$S \to F(a, a) \to F(G(a), G(a)) \to F(G(G(a)), G(G(a))) \to \cdots \to f(g(g(a)), g(g(a)))$



Betrachte nun folgende Ableitungsrelation für reguläre Grammatiken:
Wir schreiben $s \to_G$ genau dann wenn ein Kontext $C \in T_{\Sigma \cup N}(x)$ existiert, sodass
$s = C[A], t = C[\alpha], A \to \alpha \in R$ für $G = (S, N, \Sigma, R)$.
Mit $\to_G^*$ bezeichnen wir die transitive Hülle von $\to_G$.

2.2 Definition

Ist G eine reguläre Grammatik, dann heißt $L(G) = \{t \in T_\Sigma | S \to_G^* t\}$ die von G akzeptierte Sprache.
Eine Sprache $L \subseteq T_\Sigma$ heißt regulär, falls $L = L(G)$ für eine reguläre Grammatik.
Betrachte reguläre Grammatik $G = (S, N, \Sigma, R)$ Wir bezeichnen mit $L_G(A)$, $A \in N$, die von G erzeugte Sprache mit A als Startsymbol.

2.3 Definition - reduziert

Sei $G = (S, N, \Sigma, R)$ eine reguläre Grammatik und $A \in N$. Dann heißt A

- erreichbar, falls ein Kontext $C \in T_\Sigma(x)$ existiert, so dass $S \to_G^* C[A]$
- produktiv, falls $L_G(A) \neq \emptyset$

G heißt reduziert, falls alle $A \in N$ erreichbar und produktiv sind.

2.4 Satz

Ist G eine reguläre Grammatik mit $L(G) = L$, dann existiert eine reduzierte reguläre Grammatik G' mit $L(G') = L$.

Nun: zu G eine „Normalform “konstruieren.

2.5 Definition - Normalisierung

Eine reguläre Grammatik G heißt normalisiert, falls alle Regeln aus R die Form

- $A \to f(A_1, \ldots, A_n)$, $A, A_1, \ldots, A_n \in N$, $v \in \Sigma^{(n)}$
- $A \to a$, $A \in N$, $a \in \Sigma$

2.6 Satz

Ist G eine reguläre Grammatik mit $L(G) = L$, dann existiert eine normalisierte reguläre Grammatik G' mit $L(G') = L$.

Beweis: Wir ersetzen Regeln der Form $A \to f(s_1, \ldots, s_n)$ durch $A \to f(A_1, \ldots, A_n)$ wobei: Ist $s_i \in N$, dann ist $A_i = s_i$, ansonsten ist A_i ein neues Symbol und wir fügen $A_i \to s_i$ hinzu.
...Iterieren...

Es bleiben übrig:
$A \to f(A_1, \ldots, A_n)$, $A \to a \in \Sigma$, $A_i \to A_j$ (letztere überbrücken)

Wir erhalten nun:

2.7 Theorem

L ist erkennbar $\Leftrightarrow$ L ist regulär.

Beweis:
„$\Rightarrow$ “Ist L erkennbar, so existiert ein nicht-deterministischer Top-Down-TA $\mathcal{A} = (Q, \Sigma, I, \Delta)$ mit $L = L(\mathcal{A})$. Betrachte eine Grammatik $G = (S, N, \Sigma, R)$ mit

- S ist ein neues Symbol
- $N = \{A_q | q \in Q\}$
- $R = \{A_q \to f(A_{q_1}, \ldots, A_{q_n}) | q(f(x_1, \ldots, x_n)) \to f(q_1(x_1)), \ldots, q_n(x_n) \in \Delta\} \cup \{S \to A_{q_i} | q_i \in I\}$

Offensichtlich ist $L(G) = L(A)$
„$\Leftarrow$ “analog (gehe von normalisierter Grammatik aus) $\square$

Ziel: Definieren Konkatenation und Kleene-Stern
Problem hierbei ist: Wir müssen erklären, wie wir Bäume zusammensetzen.

Dazu definieren wir: Substitution von Sprachen.

Betrachte $t \in T_{\Sigma \cup k}$, wobei $k = \{\square_1, \dots, \square_n\}$, $\square_i$ sind Konstanten.
Es seien $L_i \subseteq T_{\Sigma \cup k}$ Sprachen. Dann ist die Substitution $t\{\square_1 \leftarrow L_1, \dots, \square_n \leftarrow L_n\}$ induktiv wie folgt definiert:

- $\square_i\{\square_1 \leftarrow L_1, \dots, \square_n \leftarrow L_n\} = L_i$
- $a\{\square_1 \leftarrow L_1, \dots, \square_n \leftarrow L_n\} = \{a\}$, $a \neq \square_i$, $a \in \Sigma^0$
- $f(s_1, \dots, s_n)\{\square_1 \leftarrow L_1, \dots, \square_n \leftarrow L_n\} = \{f(t_1, \dots, t_n) | t_i \in s_i\{\square_1 \leftarrow L_1, \dots, \square_n \leftarrow L_n\}\}$

Außerdem setzen $L\{\square_1 \leftarrow L_1, \dots, \square_n \leftarrow L_n\} = \bigcup\limits_{t \in L} t\{\square_1 \leftarrow L_1, \dots, \square_n \leftarrow L_n\}$

Beispiel: Betrachte $k = \{\square_1, \square_2\}$

$$t = \begin{array}{c} f \\ \diagup \quad \diagdown \\ \begin{array}{c} f \\ \diagup\diagdown \\ \square_1 \ \ \square_2 \end{array} \qquad \square_2 \end{array}$$

$L_2 = \{a, b\}$

Dann ist
$$t\{\square_2 \leftarrow L_2\} = \left\{ \begin{array}{c} f \\ \diagup \quad \diagdown \\ \begin{array}{c} f \\ \diagup\diagdown \\ \square_1 \ \ a \end{array} \quad a \end{array}, \begin{array}{c} f \\ \diagup \quad \diagdown \\ \begin{array}{c} f \\ \diagup\diagdown \\ \square_1 \ \ a \end{array} \quad b \end{array}, \begin{array}{c} f \\ \diagup \quad \diagdown \\ \begin{array}{c} f \\ \diagup\diagdown \\ \square_1 \ \ b \end{array} \quad a \end{array}, \begin{array}{c} f \\ \diagup \quad \diagdown \\ \begin{array}{c} f \\ \diagup\diagdown \\ \square_1 \ \ b \end{array} \quad b \end{array} \right\}$$

Nun setzen wir für zwei Sprachen L, M:
$L \cdot_\square M = \bigcup\limits_{t \in L} t\{\square \leftarrow M\}$
Des weiteren ergibt sich der Kleene-Stern:

- $L^{0.\square} = \{\square\}$
- $L^{n+1.\square} = L^{n.\square} \cup L \cdot_\square L^{n.\square}$

$\Rightarrow L^{*.\square} = \bigcup\limits_{n \geq L^{n.\square}}$

Beispiel: Betrachte $L = \left\{ \begin{array}{c} f \\ \diagup\diagdown \\ \square \ \ \square \end{array}, a \right\}$

$L^{0.\square} = \{\square\}$
$L^{1.\square} = \{\square\} \cup L \cdot_\square \{\square\} = \{\square\} \cup \left\{ \begin{array}{c} f \\ \diagup\diagdown \\ \square \ \ \square \end{array}, a \right\}$

$L^{2.\square} = \left\{\square, \begin{array}{c} f \\ \diagup\diagdown \\ \square \ \ \square \end{array}, a \right\} \cup L \cdot_\square \left\{\square, \begin{array}{c} f \\ \diagup\diagdown \\ \square \ \ \square \end{array}, a \right\} = \dots$

Abschlusseingenschaften:

2.8 Satz

Es sei $L \subseteq T_{\Sigma \cup k}$ regulär, sowie $\Box_1, \dots, \Box_n \in k$. Dann ist $L\{\Box_1 \leftarrow L_1, \dots, \Box_n \leftarrow L_n\}$ regulär.

Beweis:
Betrachte normalisierte Grammatiken $G, G_1, \dots, G_n$ mit $L(G) = L, L(G_1) = L_1, \dots, L(G_n) = L_n$, $G = (S, N, \Sigma \cup k, R), G_i = (S_i, N_i, \Sigma \cup k, R_i)$. (alle Nichtterminale paarweise disjunkt)
Konstruiere $G' = (S, N', \Sigma \cup k, R')$ mit:

- $N' = N \cup N_1, \cup \dots \cup N_n$
- R' enthält alle Regeln in $R, R_1, \dots, R_n$, wobei $A \rightarrow \Box_i$ ersetzt werden durch $A \rightarrow S_i$

Direkt zeigen: $L\{\Box_1 \leftarrow L_1, \dots, \Box_n \leftarrow L_n\} \subseteq L(G')$
Wir zeigen „$\supseteq$ “
Induktion: über die Anzahl der Ableitungsschritte zeigen wir $A \rightarrow^*_{G'} s'$ mit $S \in T_{\Sigma \cup k}$ — s' enthält keine Nichtterminale
$\exists s$ mit $A^*_{G'}$ und $s' \in s\{\Box_1 \leftarrow L_1, \dots, \Box_n \leftarrow L_n\}$ — das heißt s" $\in L_G(A)\{\Box_1 \leftarrow L_1, \dots, \Box_n \leftarrow L_n\}$

Induktionsannahme: Angenommen $A \rightarrow^*_G$ in einem Schritt, das heißt $s' \in L$, die Regel kann nicht $A \rightarrow \Box_i$ sein (existiert nicht in G') und auch nicht $A \rightarrow s_i$ (kein $s' \in T_{\Sigma \cup k}$. Damit: $S' \in L$. Seze $s = s'$, damit enthält s kein $\Box_i$, also $\{s'\} = \{s\} = s\{\Box_1 \leftarrow L_1, \dots, \Box_n \leftarrow L_n\}$, also insbesondere $s' \in s\{\Box_1 \leftarrow L_1, \dots, \Box_n \leftarrow L_n\} \subseteq L(A)\{\Box_1 \leftarrow L_1, \dots, \Box_n \leftarrow L_n\}$
Induktionsschritt: $A \rightarrow^*_{G'} s'$: zerlege $A \rightarrow)G's_1 \rightarrow^*_{G'} s'$ Fälle für s_1 (bzw. $A \rightarrow s_1$)

- $a \rightarrow f(A_1, \dots, A_m) \Rightarrow s' = f(t_1, \dots, t_n)$. Laut Induktionsvoraussetzung: $t_i \in L(A_i)\{\Box_1 \leftarrow L_1, \dots, \Box_n \leftarrow L_n\} \Rightarrow s' \in L(A)$
- $A \rightarrow s_i \in R' \Rightarrow A \rightarrow \Box_i \in R$ (laut Konstruktion) $\Rightarrow \Box_i \in L(A) \Rightarrow s' \in L_i \Rightarrow s' \in L(A)\{\Box_1 \leftarrow L_1, \dots, \Box_n \leftarrow L_n\}$

Aussage gilt für alle Nichtterminale A_i, also auch für Startsymbol S . $\Box$

2.9 Satz

Ist $L \subseteq T_{\Sigma \cup K}$ regulär und $\Box \in K$, dann ist $L^{*,\Box}$ regulär.
Beweis: Betrachte normalisierte Grammatik $G = (S, N, \Sigma \cup K, R)$ mit $L(G) = L$.
Konstriere $G' = (S', N \cup \{S'\}, \Sigma \cup K, R')$ (mit $S' \notin N$) wie folgt:
R' enthälte alle Regeln aus R, wobei:

- $A \rightarrow \Box$ wird ersetzt durch $A \rightarrow S'$
- $S' \rightarrow S$
- $S' \rightarrow \Box$ (damit $\Box \in L(G')$)

Induktion liefert: $L(G') = L^{*,\Box}$

2.10 Definition

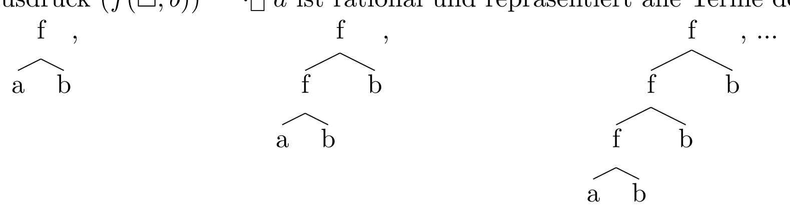
Wir formalisieren nun die rationalen Ausdrücke:
Sei Σ ein Rangalphabet, K eine Menge Konstanten mit $\square \in K$: Dann ist $Rat(\Sigma, K)$ die kleinste Menge, sodass:

- $\emptyset \in Rat(\Sigma, K)$
- $a \in \Sigma^0 \cup K' Rightarrow a \in Rat(\Sigma, K)$
- $f \in \sigma^n, E_1, \ldots, E_n \in Rat(\Sigma, K) \Rightarrow f(E_1, \ldots, E_n) \in Rat(\Sigma, K)$
- $E_1, E_2 \in Rat(\Sigma, K) \Rightarrow E_1 \cup E_2 \in Rat(\Sigma, K)$
- $E_1, E_2 \in Rat(\Sigma, K), \square \in K \Rightarrow E_1 \cdot_{\square} E_2 \in Rat(\Sigma, K)$
- $E_1 \in Rat(\Sigma, K), \square \in K \Rightarrow E_1^{*,\square} \in Rat(\Sigma, K)$

Die Ausdrücke $Rat(\Sigma, K)$ heißen rational über Σ und K. Ist E ein ratoinaler Ausdruck in $Rat(\Sigma, K)$, dann repräsentiert E eine Menge von Termen aus $T_{\Sigma \cup K}$, bezeichnet mit $||E||$.

- $||\emptyset|| = \emptyset$
- $||a|| = \{a\}$
- $f(E_1, \ldots, E_n)|| = \{f(t_1, \ldots, t_n) | t_i \in ||E_i||\}$
- $||E_1 \cup E_2|| = ||E_1|| \cup ||E_2||$
- $||E_1 \cdot E_2|| = ||E_1||\{\square \Leftarrow ||E_2||\}$
- $||E^{*,\square}|| = ||E||^{*,\square}$

Beispiel: Der Ausdruck $(f(\square, b))^{*,\square} \cdot_{\square} a$ ist rational und repräsentiert alle Terme der Form



Wir erhalten die „Baumautomaten “- Variante von Kleene:

2.11 Theorem

Rationale Ausdrücke haben die selbe Ausdrucksstärke wie bottom-up Baumautomaten.
Beweis: Ist E ein rationaler Ausdruck, so existiert laut Sätzen 2.8 und 2.9 und den Abschlusseigenschaften von Baumautomaten einen Baumautomaten $\mathcal{A}$ mit $L(\mathcal{A}) = ||E||$. Umgekehrt: Sei $\mathcal{A} = (Q, \Sigma, F, \Delta)$ ein Baumautomat. Wir zeigen: Es existiert ein rationaler Ausdruck E aus $Rat(\Sigma.K)$ mit $||E|| = L(\mathcal{A})$.
Betrachter hierzu für $1 \leq i, j \leq |Q|$ und $Z \subseteq Q$ Terme $T(i, j, Z)$ definiert wie folgt: $t \in T(i, j, Z)$, falls $t \in T_{\Sigma \cup K}$ mit einem Lauf R in t, so dass:

- $v(\varepsilon) = q_i$
- ist $p \neq \varepsilon$ und $t(p)/inZ$, dann ist $r(p) \in \{q_1, \dots, q_j\}$

Damit: $L(\mathcal{A}) = \bigcup\limits_{q_i \in F} T(i, |Q|, \emptyset)$
Induktion über j: $T(i,j,Z) \in Rat(\Sigma, K)$ $j = 0$: $t \in T(i, 0, Z)$ bedeutet, es existiert ein Lauf r

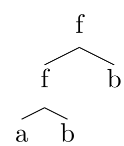
- $r(\varepsilon) = q_i$
- kein Symbol, das nicht nullstellig ist, darf gelabelt sein $\Rightarrow t = a$ oder $t = f(a_1, \dots, a_n), a_1, \dots, a_n \in \Sigma^0 \Rightarrow$ endlich viele, also existiert ein rationaler Ausdruck

Induktionsschritt: Angenommen für $j' < j$ ist $T(i, j', Z) \in Rat(\Sigma, Q)$.
Schreibe $T(i,j,Z) = T(i, j-1, Z) \cup T(i, j-1, Z \cup \{q_j\}) \cdot_{q_j} (T(j, j-1, Z \cup \{q_j\}))^{*\cdot q_j} \cdot_{q_j} T(j, j-1, Z)$

Ziel: Zusammenhang zwischen Baum- und Wortsprachen
Neben der Pfadsprache (siehe Übung) betrachten wir den sogenannten Yield-Operator, definiert wie folgt:

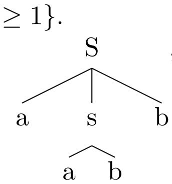
- $Yield(a^{(0)}) = a$ für eine Konstante $a^{(0)} \in \Sigma$
- $Yield(f(t_1, \dots, t_n)) = Yield(t_1) \cdot \dots \cdot Yield(t_n)$ für $f^{(n)} \in \Sigma, t_i \in T_\Sigma$

Für Sprache L gilt: $Yield(L) = \bigcup\limits_{t \in \Sigma} Yield(t)$
Beispiel: $Yield(f(f(a,b),b)) = abb$



$Yield(||g(a, \square, b)^{*\cdot\square} \cdot_\square f * (a,b)||) = \{a^n b^n | n \in \mathbb{N}, n \geq 1\}$

Wir wolen reguläre Baumsprachen mit Ableitungsbäumen von kontextfreien Wortgrammatiken „vergleichen“. Betrachte $G = (S, N, T, R)$ (kontextfreie Wortgrammatik, d.h. Regeln in R haben die Form $A \to \alpha$, wobei $A \in N, \alpha \in (N \cup T)^+$).
Betrachte zum Beispiel $G = (S, N, T, R)$ mit Regeln:
$S \to aSb, S \to ab$ mit $L(G) = \{a^n b^n | n \in \mathbb{N}, n \geq 1\}$.
Der Syntaxbaum für $aabb$ hat die Form



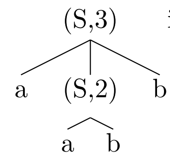
, d.h. S hat keinen festen Rang.

Betrachte daher für gegebene Grammatik G Tupel (A, m) für jedes $A \in N$, sodass $A \to \alpha \in R$ mit $|\alpha| = m$.
Zu einem Symbol $a \in T \cup N$ definieren wir die Menge der von a ausgehenden Ableitungsbäume in G, $D(G, a)$, wie folgt:

- $D(G, a) = \{a\}$ für $a \in T$
- $(a, 0)(\varepsilon) \in D(G, a)$, falls $a \to \varepsilon \in R$

- $(a,m)i(t_1,\ldots,t_m) \in D(G,a)$, falls $t_i \in (G,a_i), a_i \in T \cup N, a \to a_1,\ldots,a_m \in R$
- Ableitungsbäume für G sind $D(G) = \bigcup a \in t \cup N D(G,a)$

Beispiel: $G = (S,N,T,R)$ mit $S \to aSb, S \to ab$. Dann ist



in $D(G)$

Bemerkung: Ist G eine kontextfreie Grammatik, so ist natürlich $Yield(D(G)) = L(G)$.

2.12 Satz

1.) Ist $G = (S,N,T,R)$ eine kontextfreie Wortgrammatik, dann ist $D(G)$ reguläre Baumsprache.
2.) Ist L reguläre Baumsprache, dann ist $Yield(L)$ kontextfreie Wortsprache.
3.) Es existieren reguläre Baumsprachen, die nicht Ableitungsbäumen von kontextfreien Sprachen entsprechen.

Beweis:
1.) Erzeuge Grammatik $G' = (S,N,\Sigma,R')$ mit:

- $\Sigma = T \cup \{\varepsilon\} \cup \{(A,m) | A \in N, A \to \alpha \in R \text{ mit } |\alpha| = m\}$
- $A \to (A,0)(\varepsilon) \in R'$, falls $A \to \varepsilon \in R$
- $A \to (A,m)(A_1,\ldots,A_m) \in R'$, falls $A \to A_1 \ldots A_m \in R$

damit ist offensichtlich $Yield(L(G')) = L(G)$.
2.) Betrachte normalisierte Grammatik $G = (S,N,\Sigma,R)$ mit $L(G) = L$.
Erzeuge Wortgrammatik $G' = (S,N,\Sigma^{(0)},R')$ mit folgenden Regeln:

- $A \to A_1,\ldots,A_m$ $(A \to a) \in R'$, falls $A \to f(A_1,\ldots,A_n)$ $(A \to a) \in R$ für $f^{(m)} \in \Sigma$

3.) Übung