Der Wortgraph eines Texts

Hans Leiß

26.1.99, 24.5.04

(i) Der Index eines Texts $t \in \Sigma^*$

(ii) Die Endäquivalenz $\sim_t$ von Teilwörtern

(iii) Der Teilwortautomat $\mathcal{B}_t = \Sigma^* / \sim_t$

(iv) Die On-Line-Konstruktion von $\mathcal{B}_t$ in $O(|t|)$

Literatur: A.Blumer,J.Blumer,D.Haussler,M.T.Chen,J.Seiferas: The smallest automaton recognizing the subwords of a text. *Theoretical Computer Science 40* (1985), 31–55

Es ist in der Mathematik oft sehr viel schwieriger, eines anderen Meditata zu verstehen als selbst zu meditieren. G.C.Lichtenberg

Der Index eines Textes $t \in \Sigma^*$

Sei $t \in \Sigma^*$ ein Text. Definiere die Menge der *Teilwörter* von t,

$$Subw(t) := \{\, w \in \Sigma^* \mid \exists u, v \in \Sigma^*\, t = uwv \,\}$$

und den *Index* von t:

$$index(t) \quad := \quad \{\, (w, i) \mid \exists x, y \in \Sigma^*\ (t = xwy \wedge i = |x|) \,\}$$

Die Menge der *Endpositionen von* $u \in \Sigma^*$ *in* t,

$$Pos(u, t) := \{\, |xu| \mid x, y \in \Sigma^*\ t = xuy \,\},$$

liefert eine Äquivalenzrelation $\sim_t$ auf Σ^*:

$$\begin{array}{rcl} u \sim_t v & :\Longleftrightarrow & Pos(u,t) = Pos(v,t), \\ [u]_t & := & \{\, v \in \Sigma^* \mid u \sim_t v \,\}. \end{array}$$

Beachte:

$$(u, i) \in index(t) \iff i + |u| \in Pos(u, t).$$

Die Äquivalenzklassen

Auf $[u]_t$ ist die Suffix-Relation $\leq_{\mathit{Suffix}}$ eine totale Ordnung. Daher hat man

$$[u]_t = \begin{cases} \Sigma^* - Subw(t) & u \notin Subw(t), \\ \Sigma^* \cdot \min [u]_t \cap \mathit{Suffix}(\max [u]_t), & \text{sonst.} \end{cases}$$

Der Teilwortautomat $\mathcal{B}_t = (Q_t, \Sigma, q_t, \delta_t, F_t)$

Proposition 1 Sei $\mathcal{B}_t = (Q_t, \Sigma, q_t, \delta_t, F_t)$ mit

$$\begin{array}{lcl} Q_t & := & \Sigma^*/\sim_t, \\ q_t & := & [\varepsilon]_t, \\ \delta_t([u]_t, a) & := & [ua]_t, \\ F_t & := & \{\,[u]_t \mid u \in \mathit{Subw}(t)\,\}. \end{array}$$

Dann ist δ_t wohldefiniert und $L(\mathcal{B}_t) = \mathit{Subw}(t)$.

Beweis: δ ist wohldefiniert: sei $u \sim_t v$ und $a \in \Sigma$.

Fall $u \notin \mathit{Subw}(t)$: wegen $\mathit{Pos}(v,t) = \mathit{Pos}(u,t) = \emptyset$ ist $v \notin \mathit{Subw}(t)$ und folglich auch $ua, va \notin \mathit{Subw}(t)$. Also ist $ua \sim_t va$.

Fall $u \in \mathit{Subw}(t)$: Dann ist $v \in \mathit{Subw}(t)$ und etwa $u \leq_{\mathit{suffix}} v$. Dann kommt ua in t höchstens als Suffix von va vor. Also $\mathit{Pos}(ua,t) = \mathit{Pos}(va,t)$, d.h. $ua \sim_t va$.

$L(\mathcal{B}_t) = \mathit{Subw}(t)$: Für alle $w \in \Sigma^*$ ist $\delta(q_t, w) = [w]_t$. Nach der Wahl von F_t folgt daraus $L(\mathcal{B}_t) = \mathit{Subw}(t)$. $\square$

Bem.: Mit $F := \{\,[w]_t \mid w \in \mathit{Suffix}(t)\,\}$ erhält man den *minimalen* deterministischen Automaten für $L := \mathit{Suffix}(t)$: wegen

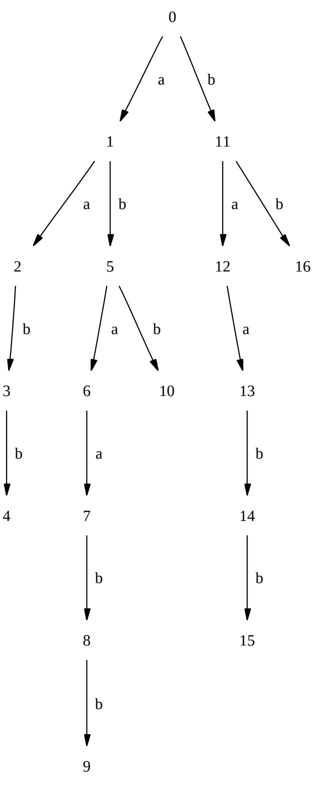
$$v \sim_t u \iff \forall w \in \Sigma^* (vw \in L \leftrightarrow uw \in L)$$

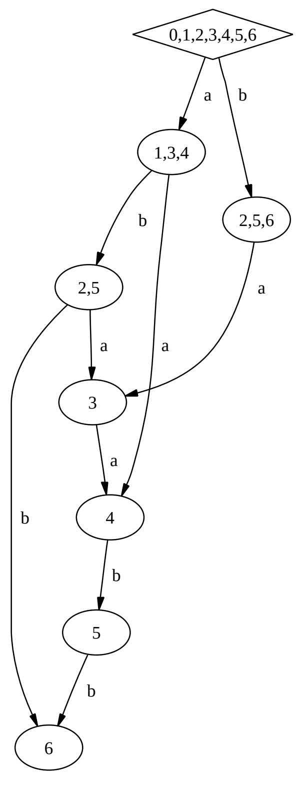
ist $\sim_t$ die Äquivalenzrelation $\sim_L$ aus dem Satz von Nerode.

Beispiel $t = abaabb$

$\sim_t$ -Klasse	Wortmenge	Positionsmenge
$[\epsilon]$	$\{\epsilon\}$	$\{0, 1, 2, 3, 4, 5, 6\}$
$[a]$	$\{a\}$	$\{1, 3, 4\}$
$[b]$	$\{b\}$	$\{2, 5, 6\}$
$[aa]$	$\{aa, baa, abaa\}$	$\{4\}$
$[ab]$	$\{ab\}$	$\{2, 5\}$
$[ba]$	$\{ba, aba\}$	$\{3\}$
$[bb]$	$\{bb, abb, aabb,$ $baabb, abaabb\}$	$\{6\}$
$[aaa]$	$\Sigma^* \setminus Subw(t)$	$\emptyset$
$[aab]$	$\{aab, baab, abaab\}$	$\{5\}$

Trie $\mathcal{T}_t$ und Automat $\mathcal{B}_t$ ohne reject = $[aaa]$





Eigenschaften des Wortgraphen von t

($\mathcal{B}_t$ ohne Zustand `reject`)

(i) Er ist der gerichtete azyklische Graph, der aus dem Suffix-Trie durch Identifizieren isomorpher Teilbäume entsteht.

(ii) Die Anzahl seiner Knoten und Kanten ist linear in $|t|$: für $t \neq \epsilon$ hat er $< 2|t|$ Knoten und $< 3|t|$ Kanten.

(iii) Er eignet sich zur Subwortsuche, aber nicht zur Informationsspeicherung an den Knoten.

(iv) Er kann als Index (d.h. mit Zuständen als Positionsmengen) in $O(|t|^2)$ und als Automat in $O(|t|)$ Schritten konstruiert werden.

Beweis von (ii): Seien $E_q \subseteq \{0, \ldots, |t|\}$ die Endpositionen zu $q \in Q_t$. Zwei bzgl. $\subseteq$ unvergleichbare E_p, E_q sind disjunkt, also ist $(Q_t, \supseteq)$ ein Baum, mit Wurzel $\{0, \ldots, |t|\}$.

Die Blätter E_q sind disjunkt, also gibt es höchsten $|t|$ davon. Es gibt $|t| + 1$ viele $q = [w]$ wo w ein Präfix von t ist.

On-line-Konstruktion von $\mathcal{B}_t$

Theorem 1
Man kann $\mathcal{B}_t$ in $O(|t|^2)$ Schritten aus t konstruieren.

Beweis:

<u>Fall $t \equiv \varepsilon$</u>: $\mathcal{B}_\varepsilon = (Q_\varepsilon, \Sigma, \delta_\varepsilon, [\varepsilon]_\varepsilon, F_\varepsilon)$ mit

$$\begin{array}{lcl} Q_\varepsilon & = & \{[\varepsilon]_\varepsilon, \Sigma^* - [\varepsilon]_\varepsilon\}, \\ F_\varepsilon & = & \{[\varepsilon]_\varepsilon\}, \\ \delta_\varepsilon([\varepsilon]_\varepsilon, a) & = & \delta_\varepsilon(\Sigma^* - [\varepsilon]_\varepsilon, a) = \Sigma^* - [\varepsilon]_\varepsilon, \end{array}$$

wobei $[\varepsilon]_\varepsilon = \{\varepsilon\}$.

Offenbar ist $L(\mathcal{B}_\varepsilon) = Subw(\varepsilon)$ und $Pos(\varepsilon, \varepsilon) = \{0\}$.

<u>Fall $t \equiv sa$</u>: Wir müssen die Klassen $[u]_s \in Q_s$ zu den Äquivalenzklassen $[u]_{sa}$ abändern und δ_{sa} definieren.

Beachte: $Pos(u, sa) \in \{Pos(u, s), Pos(u, s) \cup \{|sa|\}\}$.

Sei $\mathcal{B}_s$ mit $L(\mathcal{B}_s) = Subw(s)$ in $O(|s|^2)$ Schritten konstruiert. Für Teilworte v von s sei auch die Funktion

$$f_s(v) := \begin{cases} \max\{\, w \in Suffix(v) \mid w \not\sim_s v \,\}, & \text{falls } v \neq \varepsilon, \\ *, & \text{sonst} \end{cases}$$

schon in einer Tabelle abgespeichert, so daß wir auf $f_s(v)$ in $O(|s|)$ Schritten zugreifen können.

Beachte: $[u]_s = [v]_s \;\Rightarrow\; f_s(u) = f_s(v)$

Wir bilden daraus den Automaten $\mathcal{B}_{sa}$ und die Funktion f_{sa}.

Bestimme die Äquivalenzklassen $[u]_{sa}$

Haben u, v dieselben Endpositionen in sa, so auch in s. Daher:

$$[u]_{sa} \subseteq [u]_s. \tag{1}$$

Fall I $\mathit{Suffix}(sa) \cap [u]_s = \emptyset$:

$$[u]_{sa} = [u]_s, \tag{2}$$

da nur die Suffixe von sa Vorkommen in sa haben, die nicht in s liegen. (Aber i.a. $[u]_{sa} \neq [u]_s - \mathit{Suffix}(sa)$, z.B. bei $u = a = s$.)

Fall II $\mathit{Suffix}(sa) \cap [u]_s \neq \emptyset$:

Falls $\varepsilon \in \mathit{Suffix}(sa) \cap [u]_s$, so ist $u \equiv \varepsilon$ und

$$[u]_{sa} = [\varepsilon]_{sa} = \{\varepsilon\} = [\varepsilon]_s = [u]_s. \tag{3}$$

Ansonsten zerfällt $[u]_s$ in die $\sim_{sa}$-Klassen

$$[pa]_{sa}, \quad \text{für } pa \in \mathit{Suffix}(sa) \cap [u]_s, \tag{4}$$

$$[u]_s - \bigcup \{\, [pa]_{sa} \mid pa \in \mathit{Suffix}(sa) \cap [u]_s \,\}. \tag{5}$$

Bestimme $[pa]_{sa}$ für $p \in \mathit{Suffix}(s)$

Wir durchlaufen die Suffixe p von s mit abnehmender Länge.

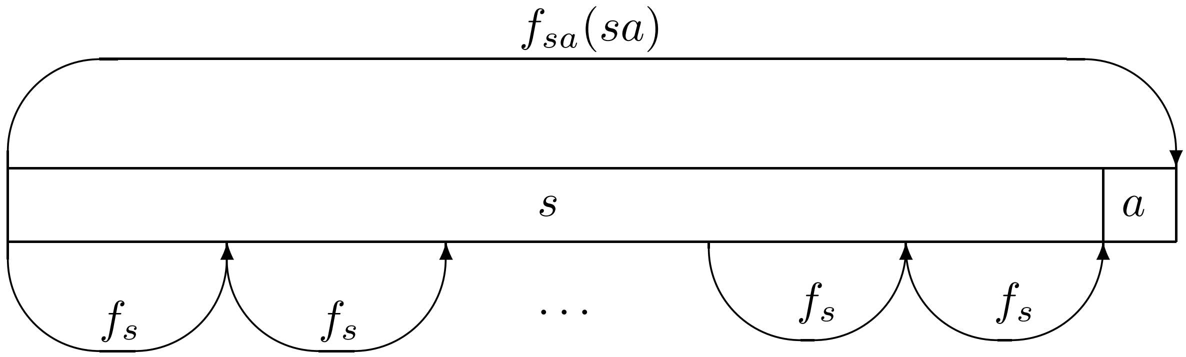
Solange $\delta_s([p]_s, a) \notin F_s$ ist, ist $pa \notin \mathit{Subw}(s)$ und daher

$$[pa]_{sa} = [sa]_{sa} = \mathit{Suffix}(sa) - \mathit{Subw}(s). \tag{6}$$

Da dasselbe für jedes Suffix $p' \sim_s p$ gilt, können wir hierbei von p zu $f_s(p)$ springen. Wir durchlaufen also nur die *Suffixkette*

$$\mathit{SK}(s) := \{\, f_s^k(s) \;\mid\; k \leq |s| \wedge \forall i \leq k \;\, \delta_s([f_s^i(s)]_s, a) \notin F_s \,\}$$

Fall II.1 $\varepsilon \in SK(s) = \{\, f_s^k(s) \mid \forall i \leq k\ \delta_s([f_s^i(s)]_s, a) \notin F_s \,\}$



Dann ist a kein Teilwort von s, also für jedes $p \in \mathit{Suffix}(s)$:

$$[pa]_{sa} = \mathit{Suffix}(sa) - \{\varepsilon\} \tag{7}$$
$$f_{sa}(pa) = \varepsilon. \tag{8}$$

Also wird nur $[sa]_s = \mathit{reject}_s := \Sigma^* - \mathit{Subw}(s)$ aufgespalten in

$$[sa]_{sa} = \mathit{Suffix}(sa) - \{\varepsilon\} \quad \text{und} \quad \mathit{reject}_{sa} := \Sigma^* - \mathit{Subw}(sa).$$

Für δ_{sa} ergänze δ_s um

$$[sa]_{sa} \xrightarrow{\quad b \in \Sigma \quad} \mathit{reject}_{sa}$$

und ersetze für $p \in \mathit{Suffix}(s)$

$$[p]_s \xrightarrow{\quad a \quad} \mathit{reject}_s$$

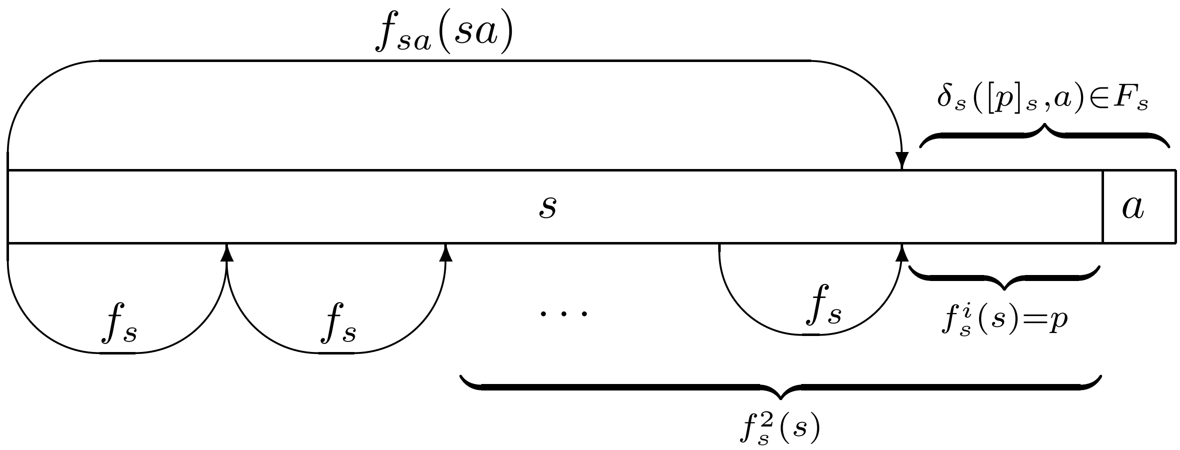
durch

$$[p]_{sa} = [p]_s \xrightarrow{\quad a \quad} [sa]_{sa}$$

und sonst überall reject_s durch reject_{sa}.

Fall II.2 $\varepsilon \notin SK(s)$**.** Dann sei

$$p := \max\{\, p \in \mathit{Suffix}(s) \mid \delta_s([p]_s, a) = [pa]_s \in F_s \,\}.$$



Wegen $pa \in \mathit{Subw}(s)$ und $sa \notin \mathit{Subw}(s)$ gilt $pa \not\sim_{sa} sa$. Für $p <_{\mathit{suffix}} r \leq_{\mathit{suffix}} s$ ist daher

$$\begin{aligned} [ra]_{sa} &= \mathit{Suffix}(sa) - \mathit{Suffix}(pa) && (9) \\ f_{sa}(ra) &= pa. && (10) \end{aligned}$$

Bisher wurde $[sa]_s = \mathit{reject}_s = \Sigma^* - \mathit{Subw}(s)$ aufgespalten in

$$\begin{aligned} [sa]_{sa} &= \mathit{Suffix}(sa) - \mathit{Suffix}(pa), \\ \mathit{reject}_{sa} &:= \Sigma^* - \mathit{Subw}(sa). \end{aligned}$$

Für δ_{sa} brauchen wir die Übergänge

$$[sa]_{sa} \xrightarrow{\quad b \in \Sigma \quad} \mathit{reject}_{sa}$$

Es sind noch die $[ra]_{sa}$ für $r \in \mathit{Suffix}(p)$ zu bestimmen.

Bestimme $[ra]_{sa}$ für $r \in \mathit{Suffix}(p)$

Fall II.2 a) $\max[pa]_s = pa$:

Dann ist $[pa]_s \subseteq \mathit{Suffix}(pa)$ und daher $[pa]_{sa} = [pa]_s$. Ist $ra = f_s(pa)$, so ist $\max[ra]_s = ra$ ein Suffix von pa und daher auch $[ra]_{sa} = [ra]_s$. Induktiv ebenso für $f_s^i(pa)$, also:

$$[ra]_{sa} = [ra]_s \qquad \text{für alle } r \in \mathit{Suffix}(p) \tag{11}$$

Damit kennen wir $Q_{sa} \simeq Q_s \mathbin{\dot\cup} \{[sa]_{sa}\}$.

Wir brauchen

$$f_{sa}(ra) := \begin{cases} pa & \text{für } p <_{\mathit{suff}} r \leq_{\mathit{suff}} s \\ f_s(ra) & \text{für } r \leq_{\mathit{suff}} p. \end{cases}$$

Für die Übergangsfunktion δ_{sa} ersetze für $p <_{\mathit{suff}} r \leq_{\mathit{suff}} s$

$$[r]_s \xrightarrow{\quad a \quad} \mathit{reject}_s$$

durch

$$[r]_s = [r]_{sa} \xrightarrow{\quad a \quad} [sa]_{sa}$$

und sonst reject_s durch reject_{sa}.

(Es ist $[r]_s \neq [r]_{sa}$ nur, wenn $[r]_s$ ein Suffix $r'a$ von sa enthält, das länger als pa ist. Aber dann wäre $p <_{\mathit{suffix}} r' \leq_{\mathit{suffix}} s$ mit $r'a \in \mathit{Subw}(s)$, im Widerspruch zur Wahl von p.)

Bestimme $[ra]_{sa}$ für $r \in \mathit{Suffix}(p)$

Fall II.2 b) $\max [pa]_s = wpa$ für ein $w \neq \varepsilon$:

Wegen $p = \max\{\, p \in \mathit{Suffix}(s) \mid pa \in \mathit{Subw}(s)\,\}$ ist $wpa \notin \mathit{Suffix}(sa)$, also $wpa \not\sim_{sa} pa$. Spalte $[pa]_s$ auf:

Die Elemente von $[pa]_s$, die keine Suffixe von sa sind, kommen (nach Wahl von p) in sa genau dort vor, wo sie in s vorkommen, sind also auch bezüglich $\sim_{sa}$ äquivalent. Sie bilden die Klasse

$$[\max[pa]_s]_{sa} = \mathit{Suffix}(\max [pa]_s) - \mathit{Suffix}(pa). \qquad (12)$$

Die übrigen Elemente kommen sowohl in s wie in sa nur als Suffix von pa vor, weshalb

$$[pa]_{sa} = [pa]_s \cap \mathit{Suffix}(pa) \qquad (13)$$

ist. Damit haben wir $[pa]_s$ in zwei $\sim_{sa}$-Klassen zerlegt.

Zur Bestimmung der weiteren $[ra]_{sa}$ durchlaufen wir mit f_s die nötigen Suffixe r von p. In sa hat ra ein neues Vorkommen als Suffix; daher ist

$$[ra]_{sa} \quad = \quad \begin{cases} [pa]_{sa} & \text{falls } [ra]_s = [pa]_s, \\ [ra]_s & \text{sonst.} \end{cases} \qquad (14)$$

Für f_{sa} brauchen wir:

$$f_{sa}(ra) := \begin{cases} pa, & \text{falls } p <_{\mathit{suffix}} r \leq_{\mathit{suffix}} wp, \\ f_s(ra) & \text{falls } r \leq_{\mathit{suffix}} p. \end{cases}$$

Zur Anpassung von δ_{sa} müssen die von der Aufspaltung betroffenen Kanten geändert werden:

Anpassung von δ_{sa} im Fall b)

a) Ersetze

$$[p]_s \xrightarrow{a} [pa]_s$$

durch

$$[p]_s \xrightarrow{a} [pa]_{sa}$$

b) Für jede Kante

$$[pa]_s \xrightarrow{b} [q]_s$$

füge eine neue Kante

$$[pa]_{sa} \xrightarrow{b} [q]_s$$

hinzu. Falls $[pa]_s = [s]_s$, ebenso für Kanten nach $[sa]_{sa}$.

c) Ist $r = f_s^i(p) \in \mathit{Suffix}(p)$, so ersetze

$$[r]_s \xrightarrow{a} [pa]_s$$

durch

$$[r]_s \xrightarrow{a} [pa]_{sa}$$

Ersetze sonst überall $[pa]_s$ durch $[\max[pa]_s]_{sa}$ und reject_s durch reject_{sa}.

Aufwandsabschätzung

Wir nehmen an:

- Σ variiert nicht mit der Eingabe t,
- Zustände werden als Positionsmengen dargestellt,
- $f_s : Q_s \to \mathit{Suffix}(s)$.

Dann können wir den Aufwand wie folgt beschränken:

(i) Bei geeignet gewählter Datenstruktur kann jede Grundoperation, wie

- einen neuen Zustand anlegen,
- eine Kante entfernen oder eine neue hinzufügen,
- die von einem Zustand ausgehende Kante mit vorgegebener Markierung suchen,
- Gleichheit mit dem Endzustand feststellen,
- Erweitern von f_{sa},

in $O(1)$ Schritten ausgeführt werden.

(ii) Berechnung von $f_s([p]_s)$:

(iii) Durchlaufen aller Suffixketten $\{\, f_s^i(s) \mid i = 0, 1, \ldots \}$:

Im Fall II.1 und im Fall II.2 a) werden jeweils $O(|SK(s)|)$ viele Kanten angelegt; im Fall II.2 b) $O(|\Sigma|)$ Kanten mit neuem Ursprung und $O(|SK(s)|)$ Kanten mit neuem Ziel.

Es genügt daher zu zeigen, daß

$$\Sigma\{\, |SK(s)| \mid s \leq_{\text{präfix}} t \,\}$$

linear in $|t|$ beschränkt ist.

Aufwandsabschätzung (Forts.)

Eine Kante $\delta([p]_s, a) = [pa]_s$ heißt *primär*, wenn

$$\max[pa] = (\max[p])a.$$

Lemma 1 Seien $w = \max[w]_s$ und $wa = \max[wa]_s$. Dann ist

$$|SK([wa]_s)| \quad \leq \quad |SK([w]_s)| - k + 1,$$

wobei k die Anzahl der nicht-primären Kanten von $SK(w)$ nach $SK(wa)$ ist.

Beweis. Jede Klasse aus $SK(wa)$ außer $[\epsilon]$ hat die Form $[pa]$ für ein $p \in \mathit{Suffix}(w)$. Daher gibt es eine Kante

$$SK(w) \ni [p]_s \xrightarrow{\quad a \quad} [pa]_s \in SK(wa)$$

Alle a-Kanten nach $[pa]$ entspringen in $SK(w)$. Zu jedem $[pa]$ ist genau eine solche a-Kante primär. Da δ_s deterministisch ist, folgt

$$|SK(wa)| - 1 \quad \leq \quad |SK(w)| - k. \qquad \square$$

Lemma 2 Sei k die Anzahl der Iterationen.

$$SK([sa]_{sa}) \quad \leq \quad SK([s]_s) - k + 2,$$

Algorithmus

Konstruiere $(\mathcal{B}_{sa}, f_{sa})$ aus (B_s, f_s) wie folgt:

1. Zerlege $[sa]_s = \mathit{reject}_s = \Sigma^* - \mathit{Subw}(s)$ in die Klassen

$$\begin{aligned} [sa]_{sa} &:= \mathit{Suffix}(sa) - \mathit{Subw}(s) \\ \mathit{reject}_{sa} &:= \Sigma^* - \mathit{Subw}(sa) \end{aligned}$$

   und definiere für δ_{sa}

$$\begin{tikzcd} {[sa]_{sa}} \arrow[r, "b \in \Sigma"] & \mathit{reject}_{sa} \arrow[loop right, "b \in \Sigma"] \end{tikzcd}$$

   Das geht in $O(1)$ Schritten.

2. Berechne $f_s(s), \ldots, f_s^i(s)$, bis $[f_s^i(s)a]_s \neq \mathit{reject}_s$ oder $f_s^i(s) = \varepsilon$ ist.

   Wir nehmen an, daß wir $f_s(v)$ für *Suffixe* v von s in einem Trie mit $O(|s|)$ Blättern gespeichert haben. Dann kostet der Übergang von $v = f_s^{i-1}(s)$ zu $f_s^i(s)$:

   - $O(|s|)$ Schritte für die Bestimmung von $f_s(v)$, und
   - $O(|s|)$ Schritte für $[f_s(v)a]_s =^? \mathit{reject}_s$ bei $f_s(v) \neq \varepsilon$.

   Macht zusammen naïv $i \times O(|s|) = O(|s|^2)$ Schritte.

3. Falls $\varepsilon = f_s^i(s)$ erreicht wird, ist $[u]_{sa} = [u]_s$ für $u \in \mathit{Subw}(s)$. Für $u \notin \mathit{Suffix}(s)$ mit $[ub]_s = \mathit{reject}_s$ setze

$$[u]_{sa} \xrightarrow{\quad b \quad} \mathit{reject}_{sa}$$

und für $p \in \mathit{Suffix}(s)$

$$\begin{tikzcd}
{[p]_{sa}} \arrow[r, "b \neq a"] \arrow[dr, "a"'] & \mathit{reject}_{sa} \\
& {[sa]_{sa}}
\end{tikzcd}$$

sowie $\qquad f_{sa}(pa) = \varepsilon.$

Das geht in $O(|s|)$ Schritten.

4. Falls $p = f_s^i(s)$ mit $[pa]_s \neq \mathit{reject}_s$ erreicht wird, ist

$$[sa]_{sa} = \mathit{Suffix}(sa) - \mathit{Suffix}(pa).$$

Fall 4.a) $pa = \max[pa]_s$:[1] Dann ist

$$[r]_{sa} = [r]_s \qquad \text{für alle } r \in \mathit{Suffix}(pa)$$

Wir brauchen

$$f_{sa}(ra) := pa \qquad \text{für } p <_{\mathit{suff}} r \leq_{\mathit{suff}} s.$$

[1] Um das festzustellen, sollte bei den $O(|s|)$ maximalen Elementen der $\sim_s$-Klassen vermerkt werden, daß sie maximal sind.

$$\begin{tikzcd}[column sep=huge]
{[r]_{sa}} \arrow[r, "a"] \arrow[dr, "b \neq a"'] & {[sa]_{sa}} \\
& \mathit{reject}_{sa}
\end{tikzcd}$$

sowie

$$f_{sa}(pa) = \varepsilon \qquad \text{für } p \in \mathit{Suffix}(s).$$

Das geht in $O(|s|)$ Schritten.

Fall 4.b) $pa <_{\mathit{suff}} \max[pa]_s$: Wir spalten $[pa]_s$ auf in

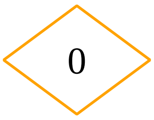
$$\begin{aligned}
[\max [pa]_s]_{sa} &= \mathit{Suffix}(\max[pa]_s) - \mathit{Suffix}(pa), \\
[pa]_{sa} &= [pa]_s \cap \mathit{Suffix}(pa)
\end{aligned}$$

und brauchen die Übergänge

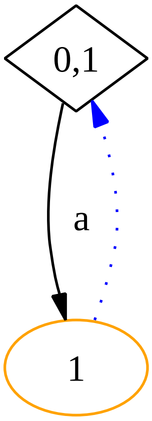
$$\begin{tikzcd}[column sep=huge]
{[r]_{sa}} \arrow[r, "a"] \arrow[dr, "b \neq a"'] & {[sa]_{sa}} \\
& \mathit{reject}_{sa}
\end{tikzcd}$$

Beispiel $t = abaabb$

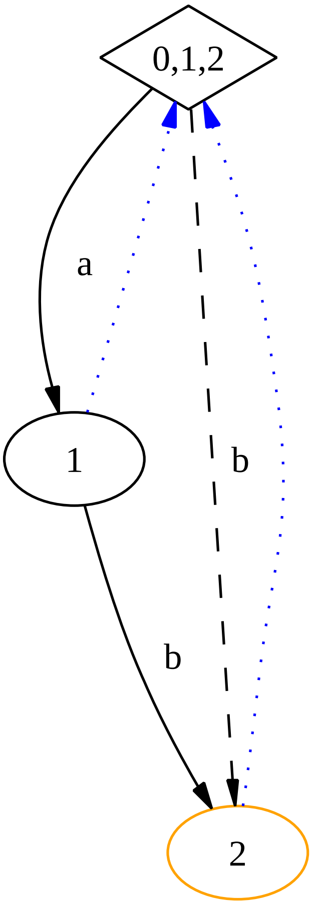
$\mathcal{B}_\epsilon =$



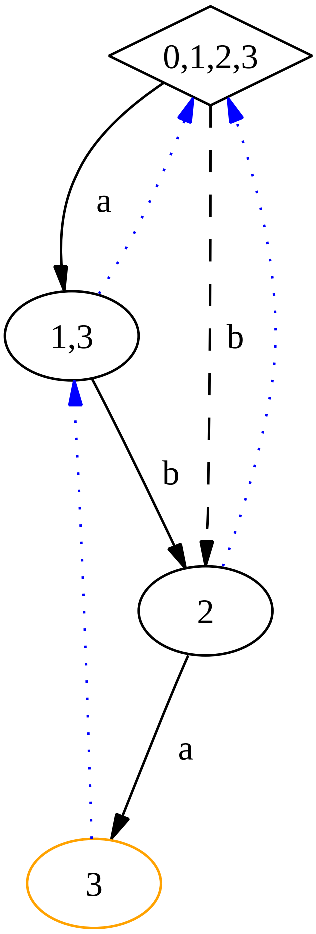
$\mathcal{B}_a =$



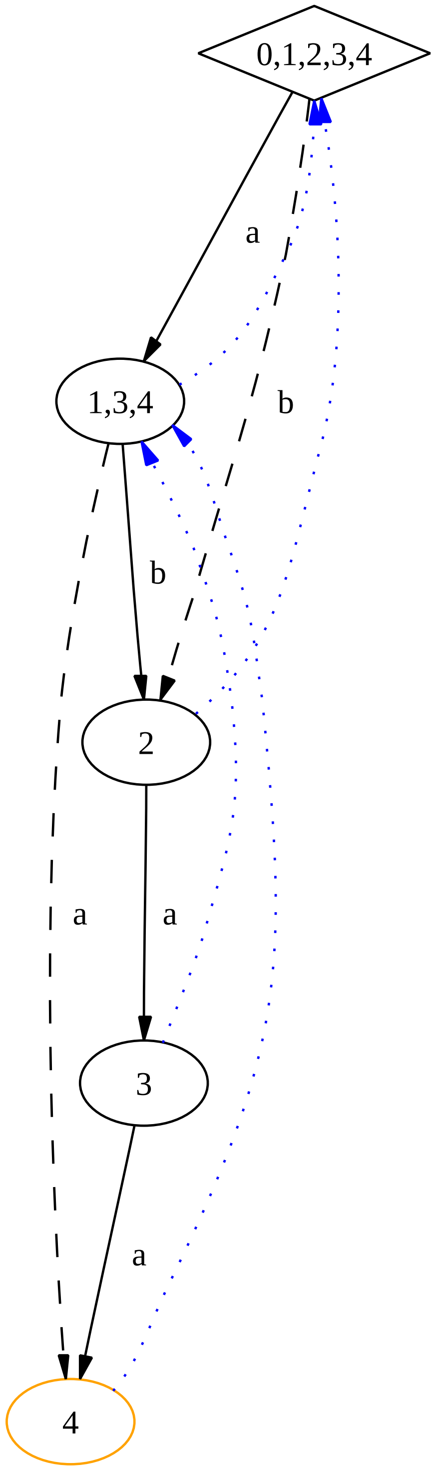
$\mathcal{B}_{ab} =$



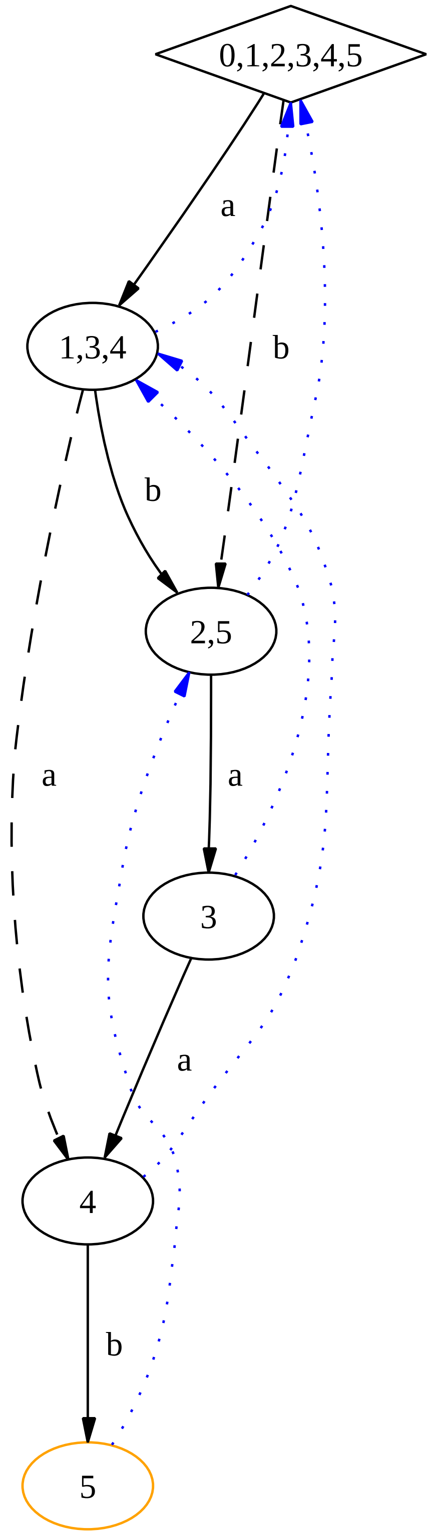
$\mathcal{B}_{aba} =$



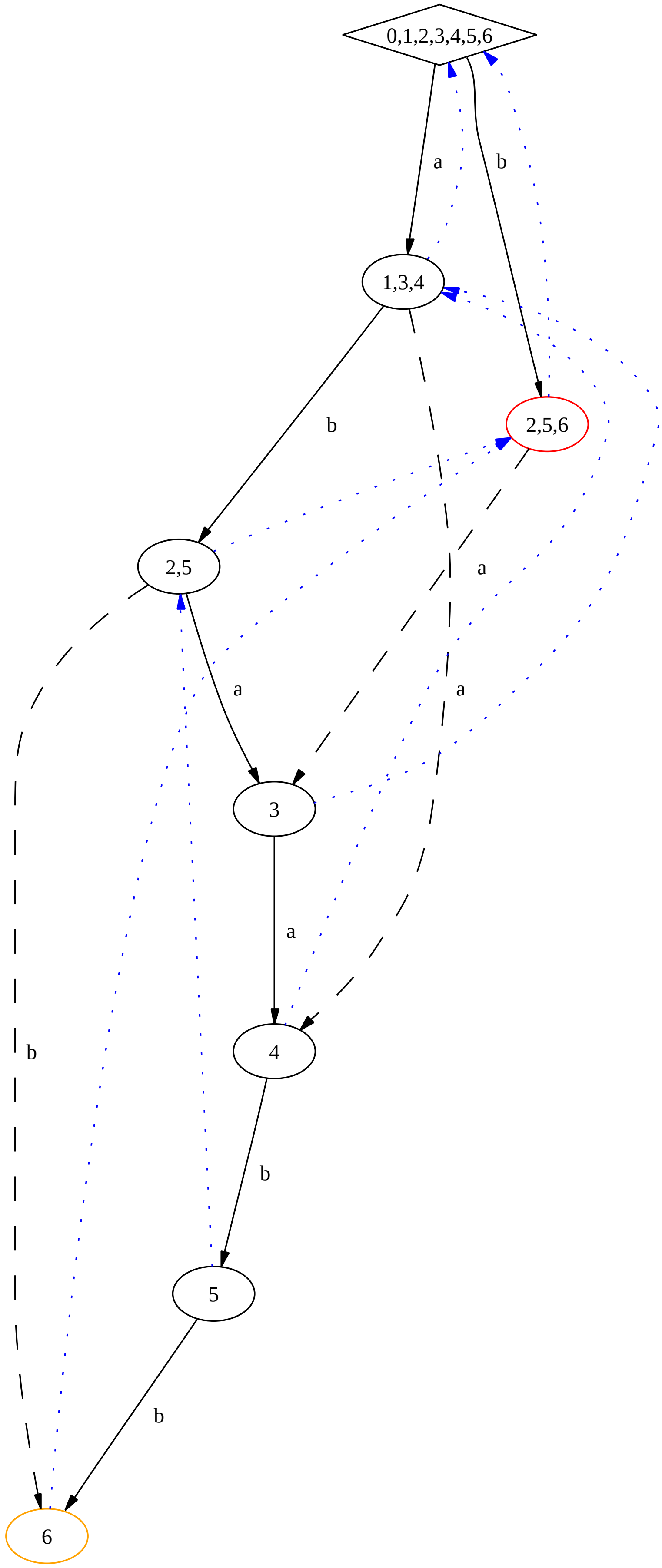
$\mathcal{B}_{abaa} =$



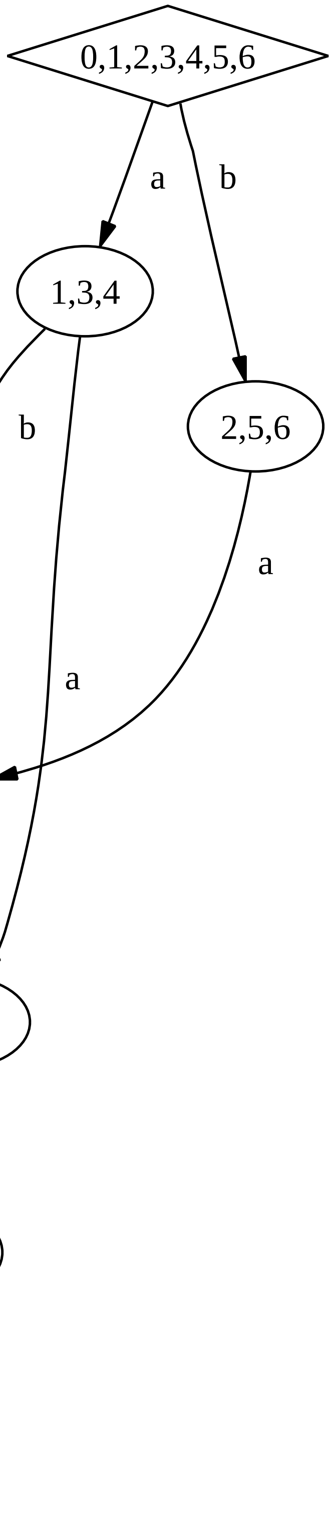
$\mathcal{B}_{abaab} =$



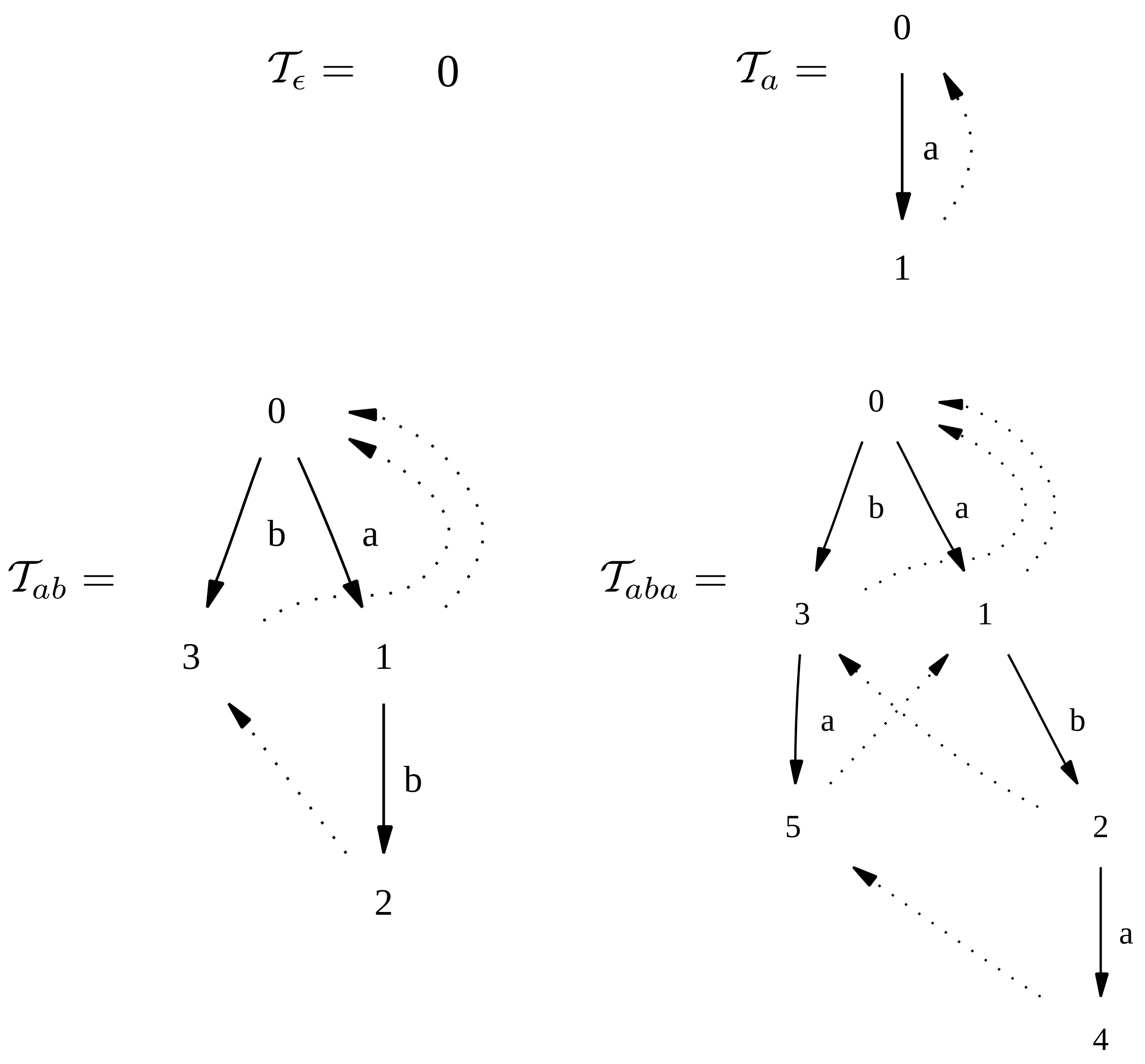
$\mathcal{B}_{abaabb} =$



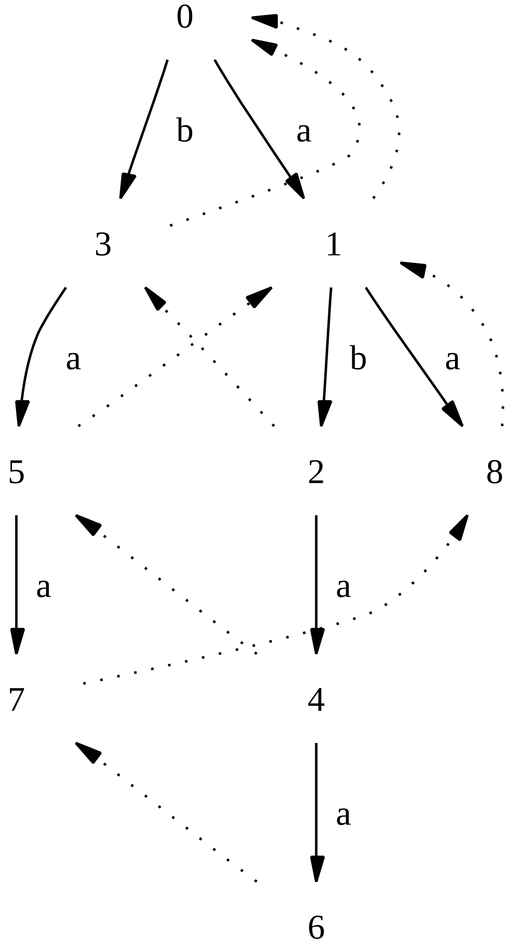
Automat $\mathcal{B}_t = \mathcal{B}_{abaabb}$



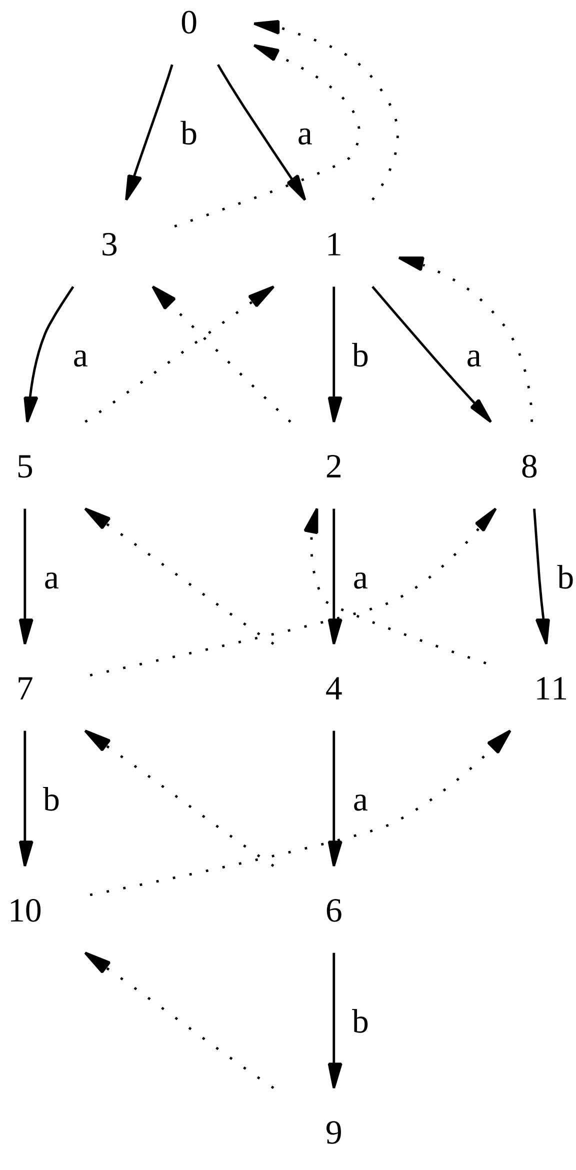
Konstruktion des Suffixtries von *abaabb*



$\mathcal{T}_{abaa} =$



$\mathcal{T}_{abaab} =$



$\mathcal{T}_{abaabb}$