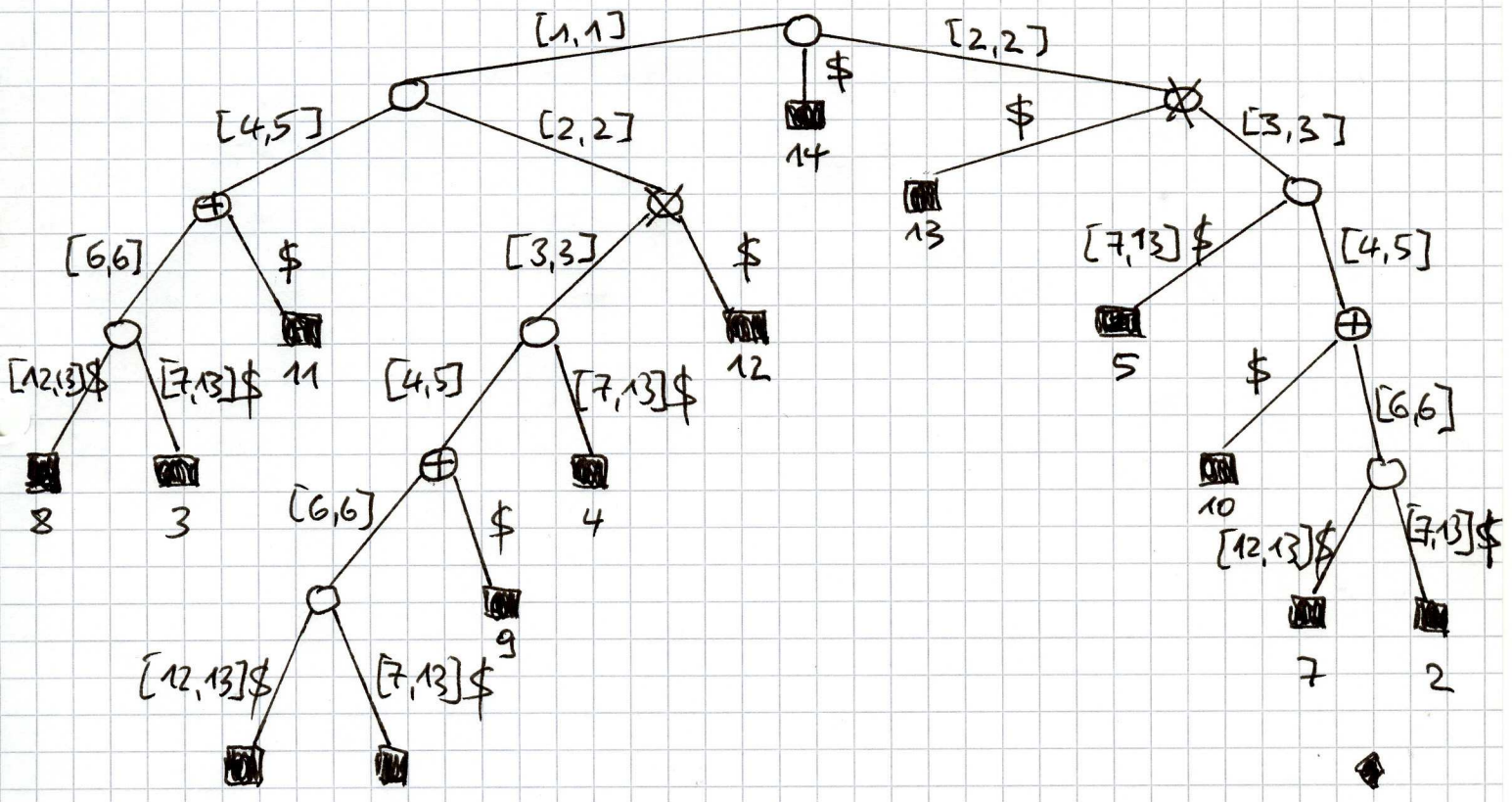
2. Kompakte auf Suffixe basierende Strukturen

In vielen praktische Anwendungen ist der benötigte Speicherplatz eine kritische Randbedingung. Suffixbäume haben die Tendenz, isomorphe Teilbäume, die sich nur durch die Nummerierung ihrer Blätter unterscheiden, zu enthalten.

Beispiel:

Suffixbaum für x = abaababaabaab$

1 2 3 4 5 6 7 8 9 10 11 12 13 14



Teilbaum, dessen Wurzel mit x markiert ist, kommt zweimal vor. Teilbaum, dessen Wurzel mit + markiert ist, kommt dreimal vor.

Ziel:
Entwicklung von Datenstrukturen, die aufgrund derartiger Übereinstimmungen Platz einsparen.

2.1 Gerichtete, azyklische Wortgraphen

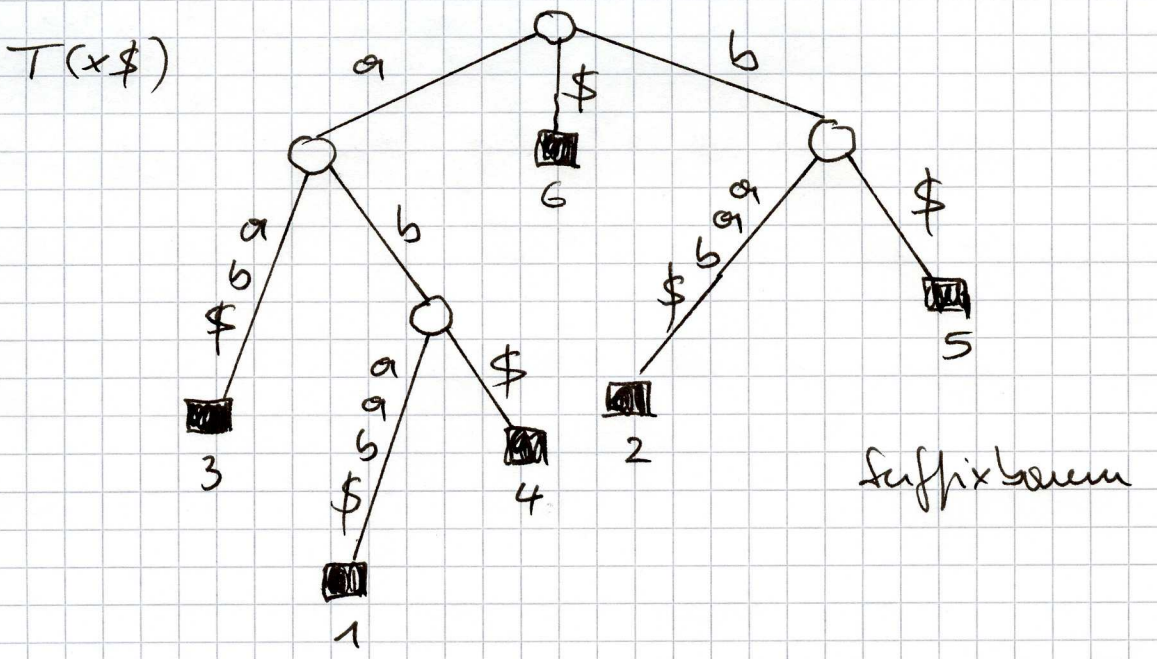
A. Blumer, J. Blumer, D. Haussler, A. Ehrenfeucht, M.T. Chen, J. Seiferas, The smallest automaton recognizing the subwords of a text, TCS 40 (1985), 31-55.

Der gerichtete azyklische Wortgraph eines Textes ist das Resultat einer Kompressionstechnik, die die oben beschriebene Redundanzen eliminiert. Gegeben einen Suffixbaum für einen Text x kann dieser in Linearzeit konstruiert werden.

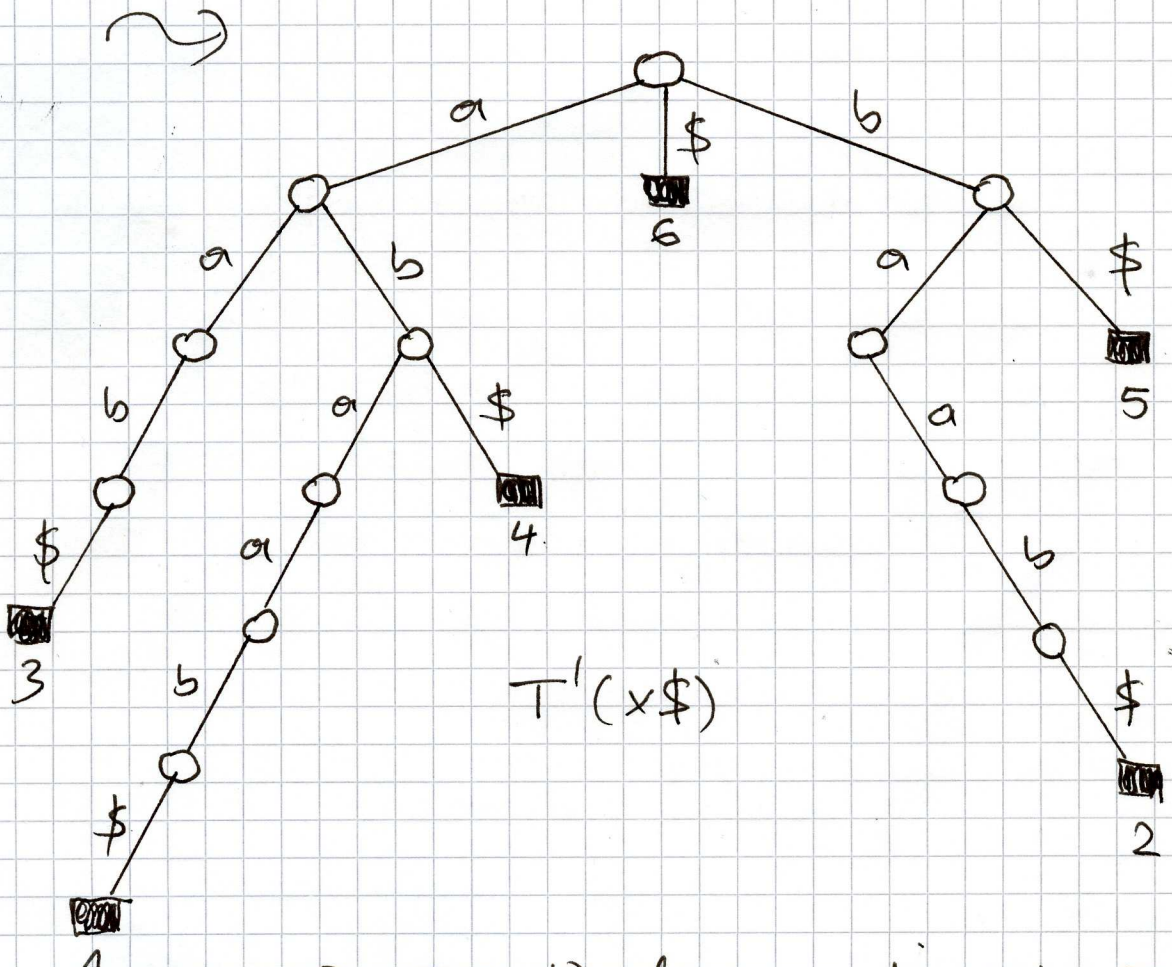
Die Entwicklung einer Methode zur Elimination von isomorphen Teilbäumen in einem Suffixbaum setzt die Analyse der Umstände, in denen derartige isomorphe Teilbäume Suffixbäume vorkommen, voraus. Hierzu sorgen wir durch anhängen des Sonderzeichens $, dass der resultierende Textstring x$ präfixfrei ist. Die Analyse ist einfacher, wenn wir diese nicht direkt am Suffixbaum, sondern an dem korrespondierenden Trie, der für jeden Teilstring von x$ einen Knoten enthält, vornehmen.

Beispiel (*):

$x\$ = abaab\$$



Suffixbaum



$T'(x\$)$

Zu $T(x\$)$ korrespondierender Trie.

◇

Sei u ein nichtleerer Teilstring von $x\$$. Dann bezeichnet Π_u die Folge von Positionen in x, in denen ein Vorkommen von u endet.

Beispiel (~~*~~):

$$\Pi_{ab} = 2, 5$$

◇

Wir vereinbaren

$$\Pi_\varepsilon := 1, 2, \ldots, n+1,$$

wobei $n := |x|$.

Bemerkung:

Die Terminatorenfolgen spiegeln die Struktur des Suffixbaumes $T(x\$)$ wider. Sei u ein nichtleerer Teilstring von $x\$$, der in exakt k Positionen $j_1, j_2, \ldots, j_k$ in x vorkommt. Dann gilt:

- $\Pi_u = j_1 + |u| - 1, j_2 + |u| - 2, \ldots, j_k + |u| - 1.$
- Die Positionen $j_1, j_2, \ldots, j_k$ sind die Nummern derjenigen Blättern im zum Präfix u korrespondierenden Teilbaum von $T'(x\$)$.

Π_ε enthält die Nummern aller Blätter, die zum Präfix ε korrespondieren; d.h., die Nummern aller Blättern in $T'(x\$)$.

Für die Konstruktion des gerichteten, azyklischen Wortgraphen interessieren wir uns für Teilstrings, die identische Terminatorenfolgen haben.

Seien u_1 und u_2 zwei Teilstrings von $x\$$. Falls $\Pi_{u_1} = \Pi_{u_2}$, dann heißen u_1 und u_2 **π-äquivalent**.

Übung
Zeigen Sie, dass die π-Äquivalenz eine Äquivalenzrelation ist.

Lemma 2.1
Seien u_1 und u_2 zwei nichtleere Teilstrings von $x\$$ mit $|u_1| \leq |u_2|$. Die Teilstrings u_1 und u_2 sind genau dann π-äquivalent, wenn u_1 nur als Suffix von u_2 in x vorkommt.

Beweis:

„$\Rightarrow$“

Annahme: $\Pi_{u_1} = \Pi_{u_2}$

$\Rightarrow$

u_1 und u_2 terminieren in exakt denselben Positionen von $x\$$.

Also impliziert $|u_1| \leq |u_2|$, dass u_1 stets ein Suffix von u_2 ist.

„$\subseteq$"

Wenn u_1 stets ein Suffix von u_2 ist, dann müssen die Terminatorenfolgen von u_1 und u_2 gleich sein. ■

Lemma 2.2

Seien u_1 und u_2 zwei Teilstrings von $x\$$ mit $|u_1| \leq |u_2|$. Dann gilt

$$\Pi_{u_1} \cap \Pi_{u_2} = \emptyset \quad \text{oder} \quad \Pi_{u_2} \subseteq \Pi_{u_1}.$$

Beweis:

Falls $u_1 = \varepsilon$, dann gilt offensichtlich $\Pi_{u_2} \subseteq \Pi_{u_1}$.

Annahme: $u_1 \neq \varepsilon$.

$\Pi_{u_1} \cap \Pi_{u_2} \neq \emptyset \Rightarrow u_1$ ist ein Suffix von u_2

Dann ist aber auch jedes Element von Π_{u_2} in Π_{u_1} enthalten. Also gilt $\Pi_{u_2} \subseteq \Pi_{u_1}$. ■

Ein Knoten in $T'(x\$)$ repräsentiert den zur Kantenmarkierung des Pfades von der Wurzel zum Knoten korrespondierenden String. Wir identifizieren nachstehend Knoten und den durch den Knoten repräsentierten String. Für einen Knoten v in $T'(x\$)$ bezeichnet T'_v den Teilbaum mit Wurzel v von $T'(x\$)$.

Lemma 2.3

Seien u_1 und u_2 zwei beliebige Knoten in $T'(x\$)$. T'_{u_1} und T'_{u_2} sind genau dann isomorph wenn u_1 und u_2 Π-äquivalent sind.

Beweis:

Falls $u_1 = u_2$, dann ist die Behauptung trivialerweise erfüllt.

Annahme: $u_1 \neq u_2$

Da ε nur zu sich selbst Π-äquivalent ist, können wir $u_1 \neq \varepsilon$ und $u_2 \neq \varepsilon$ annehmen.

Annahme: $\Pi_{u_1} = \Pi_{u_2}$.

O.b.d.A. können wir annehmen, dass $|u_1| \leq |u_2|$

Lemma 2.1 $\Rightarrow$ u_1 kommt nur als Suffix von u_2 in x vor.

$\Rightarrow$ T'_{u_2} ist auch der Teilbaum für den Suffix u_1 von u_2

$\Rightarrow$ T'_{u_2} und T'_{u_1} sind isomorph.

Annahme: T'_{u_1} und T'_{u_2} sind isomorph.

$\Rightarrow$ $\exists$ Suffix $u_1 x'\$$ von $x\$$

$(\Leftrightarrow)$

$\exists$ Suffix $u_2 x'\$$ von $x\$$

$\Rightarrow$

u_1 und u_2 müssen in x stets in derselben Position enden.

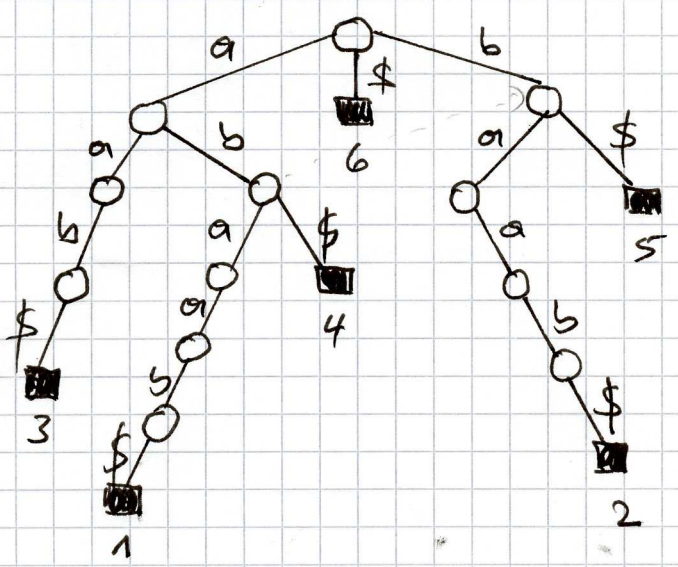
$\Rightarrow$

$$\overline{\Pi_{u_1}} = \overline{\Pi_{u_2}}.$$

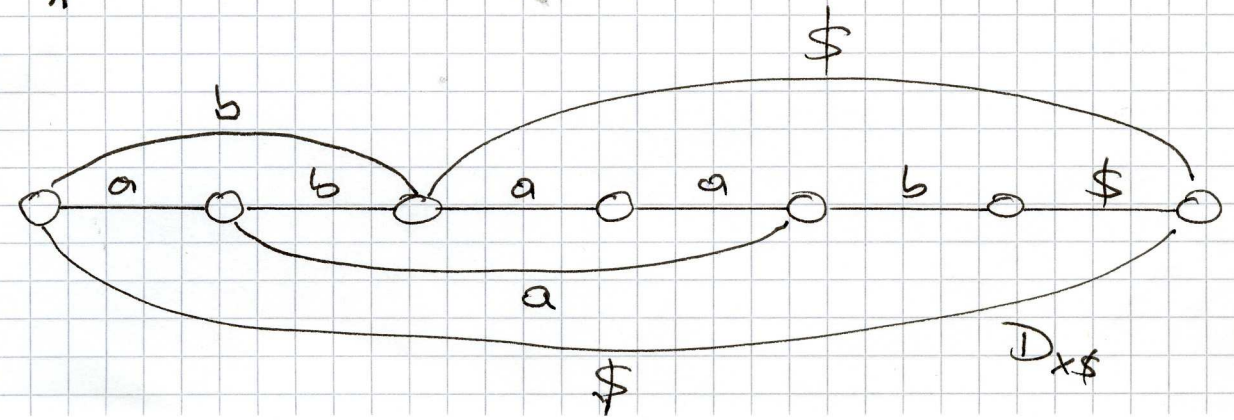
■

Der gerichtete, azyklische Wortgraph repräsentiert den minimalen endlichen Automaten, der exakt die Suffixe des zugrundeliegenden Textstrings akzeptiert. Dieser kann aus dem zum Suffixbaum T(x$) korrespondierenden Trie T'(x$) konstruiert werden. Wir erhalten diesen Graphen, indem wir in T'(x$) isomorphe Teilbäume nur einmal realisieren.

Beispiel (*): x$ = abaab$



T'(x$)



◇

Frage: Wie groß ist $D_{x\$}$ im worst case?

Der nachfolgende Satz beantwortet diese Frage.

Satz 2.1

Sei $x\$$ ein String der Länge $n+1$. Dann enthält der gerichtete azyklische Wortgraph $D_{x\$}$ höchstens $N \leq 2n+1$ Knoten und höchstens $N+n-1 \leq 3n$ Kanten.

Beweis:

Die Knoten in $D_{x\$}$ korrespondieren zu einem oder mehreren Knoten in $T'(x\$)$. Falls ein Knoten in $D_{x\$}$ zu mehreren Knoten in $T'(x\$)$ korrespondiert, dann sind die dazugehörigen Teilstrings in x π-äquivalent. Für Knoten u in $D_{x\$}$ bezeichne $k(u)$ stets den kürzesten solchen Teilstring.

Der Startknoten s von $D_{x\$}$ korrespondiert zur Wurzel des Suffixbaumes $T'(x\$)$

$\Rightarrow$

$k(s) = \varepsilon$.

Konstruktion und Lemma 2.3 $\Rightarrow$

Für unterschiedliche Knoten u_1 und u_2 von $D_{x\$}$ gilt: $k(u_1)$ und $k(u_2)$ sind nicht π-äquivalent

Somit impliziert Lemma 2.2

$\Pi_{k(u_1)} \cap \Pi_{k(u_2)} = \emptyset$ oder

$\left(\Pi_{k(u_1)} \subsetneq \Pi_{k(u_2)} \text{ oder } \Pi_{k(u_2)} \subsetneq \Pi_{k(u_1)} \right)$.

Idee:

Abarbeitung von $D_{x\$}$ in topologischer Reihenfolge mittels gleichzeitiger Konstruktion eines Baumes $B_{x\$}$, der exakt die Knoten aus $D_{x\$}$ enthält. Eine obere Schranke für die Anzahl der Knoten in $B_{x\$}$ ergibt dann eine obere Schranke für die Anzahl der Knoten in $D_{x\$}$.

Durchführung:

- Der Startknoten s von $D_{x\$}$ wird die Wurzel von $B_{x\$}$.
- Sei u derjenige Knoten in $D_{x\$}$, der gerade betrachtet wird.

  $\rightsquigarrow$

  u wird zur Wurzel desjenigen Knotens v im aktuellen Baum, für den gilt:

  i) $k(v)$ ist Suffix von $k(u)$ und

  ii) $|k(v)|$ ist maximal unter allen Knoten im aktuellen Baum, die i) erfüllen.

Da $\varepsilon = k(s)$ Suffix von $k(u)$ ist, existiert der Knoten v.

Eigenschaften von $B_{x\$}$:

1) Jeder Knoten korrespondiert zu einer nichtleeren Teilmenge von $\{1, 2, \ldots, n+1\}$.

2) Die Vereinigung der korrespondierenden Teilmengen der Söhne eines Knotens v ist in der zu v korrespondierenden Teilmenge enthalten.

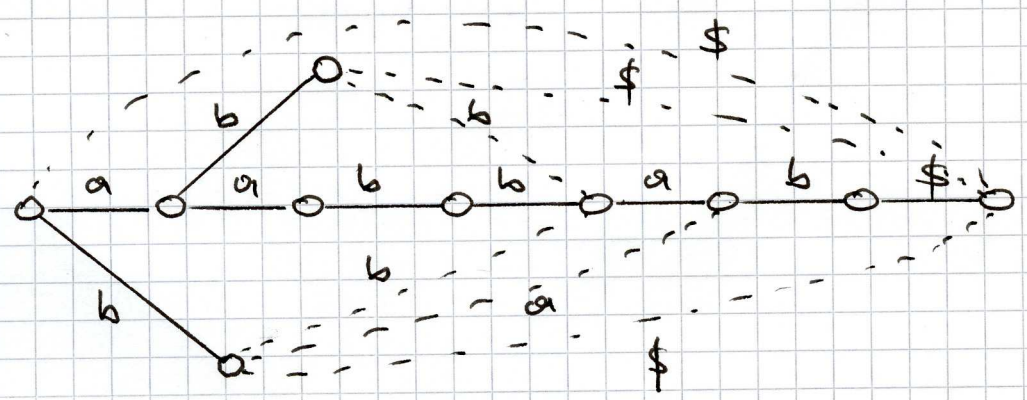
3) Befinden sich u und v auf zwei unterschiedlichen Pfaden von der Wurzel zu einem Blatt, dann sind die korrespondierenden Teilmengen disjunkt.

Grundstudium $\Rightarrow$

$B_{x\$}$ enthält $\leq 2n+1$ Knoten.

Zum Beweis der oberen Schranke für die Anzahl der Knoten in $D_{x\$}$ betrachten wir einen überspannenden Baum $ST_{x\$}$ von $D_{x\$}$, der den mit $x\$$ markierten längsten Pfad enthält.

Beispiel: $D_{aabbab\$}$, $ST_{aabbab\$}$



Nacheinander fügen wir die übrigen Kanten von $D_{x\$}$ hinzu. Da das Hinzufügen einer solchen Kante die Anzahl der Pfade von s zum Endknoten von $D_{x\$}$ mindestens um eins erhöht und die Anzahl solches Pfade n ist, ist die Anzahl der Kanten in $D_{x\$}$

$$\leq (N-1) + n.$$

■

Ziel:
Konstruktion von $D_{x\$}$ in $O(|x\$|)$ Zeit.

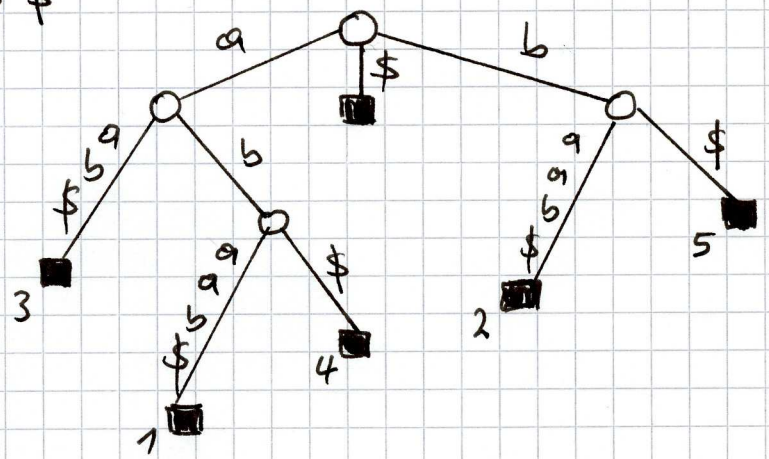
Bemerkung:
Da $T'(x\$)$ in $|x|$ quadratisch viele Knoten enthalten kann, führt der oben beschriebene Weg nicht notwendigerweise zum Ziel.

Idee:
Konstruktion von $D_{x\$}$ direkt aus dem Suffixbaum $T(x\$)$.

Betrachten wir hierzu nocheinmal das Beispiel (*).

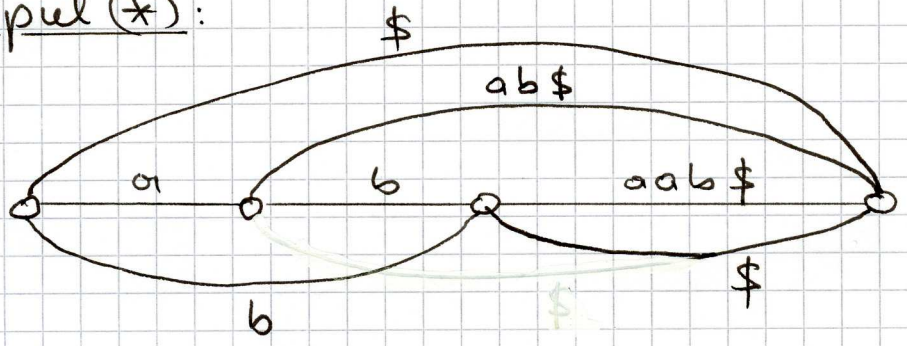
Beispiel (*):
$x\$ = abaab\$$

$T(x\$)$



Wenn wir die Methode von oben direkt auf $T(x\$)$ anwenden und dabei die Kantenmarkierungen als einzelnes Symbol interpretieren, dann erhalten wir einen gerichteten azyklischen Graphen. Dieser kann dann in den gerichteten azyklischen Wortgraphen transformiert werden.

Beispiel (*):



◇

Die Frage, die sich nun stellt ist die folgende:

Wie findet man in $T(x\$)$ effizient die isomorphe Teilbäume?

Folgendes Lemma gibt uns eine Antwort auf diese Frage:

Lemma 2.4

Seien u_1 und u_2 zwei Knoten in $T(x\$)$ mit $|m(u_1)| < |m(u_2)|$. Dann sind die Teilbäume mit Wurzel u_1 und Wurzel u_2 genau dann isomorph, wenn folgendes erfüllt ist:

i) Beide Teilbäume enthalten dieselbe Anzahl von Blättern.

ii) Es gibt von u_2 nach u_1 einen Pfad über Suffixzeiger.

Beweis:

Für $v \in T(x\$)$ bezeichne T_v den Teilbaum mit Wurzel v von $T(x\$)$.

$\Rightarrow$

Annahme: T_{u_1} und T_{u_2} sind isomorph.

$\Rightarrow$

T_{u_1} und T_{u_2} besitzen dieselbe Anzahl von Blättern.

Falls $m(u_1)$ ein Suffix von $m(u_2)$ ist, dann folgt aus unseren bisherigen Überlegungen bzgl. Suffixzeiger, dass es einen Pfad von u_2 nach u_1 über Suffixzeiger gibt.

z.z. $m(u_1)$ ist ein Suffix von $m(u_2)$.

Annahme: $m(u_1)$ ist kein Suffix von $m(u_2)$.

Wir werden nun beweisen, dass dann T_{u_1} und T_{u_2} nicht isomorph sind.

Betrachten wir irgendein Vorkommen von $m(u_2)$ in $x\$$, d.h.

$$x\$ = \alpha\, m(u_2)\, \beta\$.$$

$\Rightarrow$

In $T(u_2)$ gibt es einen Pfad von u_2 zu einem

Blatt mit Markierung $\beta\$$

Da $m(u_1)$ kein Suffix von $m(u_2)$ ist, kann es in T_{u_1} keinen Pfad von u_1 zu einem Blatt mit Markierung $\beta\$$ geben.

$\Rightarrow$

T_{u_1} und T_{u_2} sind nicht isomorph.

$\Rightarrow$

Obige Annahme ist falsch und somit $m(u_1)$ ein Suffix von $m(u_2)$.

$u \Leftarrow u$

Annahme:

T_{u_1} und T_{u_2} besitzen dieselbe Anzahl von Blättern und es gibt einen Pfad von u_2 nach u_1 über Suffixzeiger.

Wir beweisen, dass T_{u_1} und T_{u_2} isomorph sind, mittels Induktion über die Länge ℓ des Pfades von u_2 nach u_1 über Suffixzeiger.

$\ell = 1$: Es gibt einen Suffixzeiger von u_2 nach u_1

$\Rightarrow$

$m(u_2) = a\,m(u_1)$ für ein $a \in \Sigma$.

$\Rightarrow$

$\Pi(u_2) \subseteq \Pi(u_1)$

$\Rightarrow$

Für jeden Pfad P von u_2 zu einem

Blatt mit Markierung $m(P)$ in T_{u_2} gibt es in T_{u_1} den entsprechenden Pfad von u_1 zu einem Blatt mit derselben Struktur und derselben Markierung.

Da die Anzahl der Blätter in T_{u_1} und T_{u_2} gleich ist, gibt es in T_{u_1} keine weitere Pfade

$\Rightarrow$

T_{u_1} und T_{u_2} sind isomorph.

Annahme: Behauptung ist wahr für Pfade der Länge ℓ

$\ell \rightsquigarrow \ell+1$:

Betrachte Pfad $P = u_2 = v_1, v_2, \dots, v_{\ell+1}, v_{\ell+2} = u_1$ von u_2 nach u_1 über Suffixzeiger.

Beobachtung:
Die Anzahl der Blätter in T_{v_i}, $1 \leq i \leq \ell+2$ ist auf P monoton wachsend.

Da die Anzahl der Blätter in T_{v_1} und in $T_{v_{\ell+2}}$ gleich ist, haben alle Teilbäume

T_{v_i}, $1 \leq i \leq \ell+2$, dieselbe Anzahl von Blätter.

Induktionsannahme $\Rightarrow$

T_{v_1} und $T_{v_{\ell+1}}$ sind isomorph.

Induktionsanfang $\Rightarrow$

$T_{v_{\ell+1}}$ und $T_{v_{\ell+2}}$ sind isomorph.

Da die Isomorphie von Bäumen eine Äquivalenz-relation ist, sind somit T_{u_2} und T_{u_1} isomorph ■

Somit haben wir die Voraussetzungen für die Entwicklung eines Algorithmus zur Berechnung des gerichteten azyklischen Wortgraphen aus dem Suffixbaum in Linearzeit geschaffen.

Übung:
Entwickeln Sie einen Algorithmus, der für einen Textstring x$ aus dem Suffixbaum $T(x\$)$ den gerichteten azyklischen Wortgraphen $D_{x\$}$ in Zeit $O(|x\$|)$ berechnet.

2.2 Suffixarrays

Sei $x = a_1 a_2 \dots a_n \in \Sigma^n$ ein String über dem Alphabet Σ. Sei auf Σ eine totale Ordnung definiert. Dann definieren wir die lexikographische Ordnung von Strings über Σ auf die übliche Art und Weise.

Beispiel:

x = Mississippi

Dann ist die lexikographische Ordnung der Suffixe von Mississippi die folgende: