Theoretical Computer Science 3 (1977) 51–59.

SCHRANKEN FUR BALANCED TREES BEI AUSGEWOGENEN VERTEILUNGEN

Günter HOTZ

Universität des Saarlandes, Saarbrücken, Bundes Republik Deutschland

Communicated by M. Nivat
Received October 1975
Revised December 1975

We study k-nary balanced trees (B.T.).Under weak conditions for the probability distribution and a realistic measure for the elementary operations for trees we can show that two stage ternary B.T. are better then binary B.T. by approximately the factor $\log_2 3$.

1. Einleitung

Die praktische Bedeutung von Verfahren, aus großen Datenmengen rasch eine gesuchte Information zu gewinnen, ist bekannt. Kennt man die Wahrscheinlichkeit, mit der auf die Informationen einer Datenbank zugegriffen werden soll, dann möchte man diese Datenbank so organisieren, daß die mittlere Zugriffszeit ein Minimum wird. Für den Idealfall, daß die "Leitdaten" einer Bank in jeweils nur einer einzigen Speicherzelle einer Rechenmaschine stehen, hat dieses Problem bereits eine gründliche Untersuchung erfahren. Einen Algorithmus zur Konstruktion optimaler Zugriffsbäume findet man bei Knuth [3]. Balanced Trees liefern zwar nicht immer optimale Zugriffsbäume, sind aber doch recht gut, wie die Abschätzungen von Mehlhorn [4] und Bayer [2] zeigen. Das Ziel dieser Notiz besteht in zweierlei:

(1) Wir geben unter schwachen Voraussetzungen über die Wahrscheinlichkeitsverteilung eine sehr einfache Abschätzung der mittleren Zugriffszeit bei Balanced Trees.

(2) Wir zeigen, daß man unter Berücksichtigung der mit ihrer Länge wachsenden Zugriffszeit auf Daten durch die Verwendung "zweistufiger Balanced Trees" einen wesentlichen Vorteil gegenüber Balanced Trees erreichen kann.

2. Definition und Problemstellung

Sei U eine Menge, $\leqslant$ eine vollständige Ordnung auf U und $x < y$ für $x, y \in U$ genau dann erfüllt, wenn $x \leqslant y$ und $x \neq y$ gilt. $p : U \to [0, 1]$ sei eine Wahrscheinlichkeitsverteilung.

$\mathfrak{B}$ sei ein binärer Baum mit U als Knotenmenge. Jede Strecke s des Baumes sei genau einem Zeichen $z(s) \in \{<, >\}$ zugeordnet. Und zwar seien Strecken, die vom selben Knoten ausgehen, stets verschiedenen Zeichen zugeordnet.

Ist $u \in U$ und s eine Strecke des Baumes mit u als Anfangspunkt, dann ist $\mathfrak{B}(u, z(s))$ der Unterbaum von $\mathfrak{B}$, der von u aus über s erreicht werden kann.

$\mathfrak{B}$ heißt wie üblich Suchbaum über $(U, \leqslant)$, falls für alle $u \in U$ gilt: u' ist Knoten von $\mathfrak{B}(u, z(s)) \Rightarrow u'z(s)u$.

Jeder Weg w in $\mathfrak{B}$ bestimmt eindeutig ein Element $u \in U$, so daß wir auch sagen können, daß ein Suchbaum über $(u, \leqslant)$ eine Kodierung c von U in ein ternäres Alphabet nämlich in $\{<, >, =\}$ darstellt mit folgenden Eigenschaften:

(1) Jedes Wort des Kodes hat die Form $\{<, >\}^* \cdot \{=\}$.

(2) $u < v \Leftrightarrow c(u) < c(v)$, wenn die Anordnung der Kodeworte lexikographisch erfolgt mit $<$ geht $=$, und $=$ geht $>$ voran.

Die Zugriffszeit für ein Wort $u \in U$ bei festem Suchbaum ist gerade $L(c(w)) =$ Länge$(c(w))$.

Das Ziel bei der Anlage von Suchbäumen besteht darin

$$M(c, p) = \sum_{w \in U} p(w) L(c(w))$$

zu einem Minimum zu machen. Bekanntlich gilt für alle c bei ternärem Alphabet

$$H(p)/\log 3 \leqslant M(c, p)$$

wo $H(p)$ die Entropie von (U, p) ist. Da die Kodierung durch die Bedingungen (1) und (2) stark eingeschränkt ist, kann man nicht erwarten, daß man $H(p)/\log 3$ in jedem Fall durch $M(c, p)$ durch geeignete Wahl von c gut approximieren kann. Es geht also im folgenden um die Abschätzung von $M(p) = \operatorname{Min}_c M(c, p)$.

3. Balanced Trees

Sei

$$C = c(U) \subset \{-1, 1\}^* \cdot \{0\}$$

unser Kode, worin wir für $<$, $=$, bzw. $>$ nun $-1, 0$, bzw. 1 geschrieben haben. Wir identifizieren C mit $\mathfrak{B}$.

Ist C ein "Suchbaum" für U und sind

$$C_{-1} = \{w \mid (-1)w \in C\},$$

$$C_0 = \{0\},$$

$$C_1 = \{w \mid (+1)w \in C\},$$

dann ist

$$C = (-1)C_{-1} \cup C_0 \cup (1)C_1,$$

eine Zerlegung von C in paarweise fremde Mengen, so daß gilt

(3) $p(C) = p(C_{-1}) + p(C_0) + p(C_1)$,

(4) C_{-1}, C_0, C_1 sind ebenfalls Suchbäume und zwar für

$$U_{-1}(u_0) = \{u \in U \mid u < u_0\},\ \{u_0\},\ U_1(u_0) = \{u \mid u > u_0\} \text{ mit } u_0 = C^{-1}(0).$$

Definition. C ist ein *Balanced Tree* (B.T.). Es gilt

(1) $|p(U_{-1}(u_0)) - p(U_1(u_0))| \leq |p(U_{-1}(v)) - p(U_1(v))|$ für alle $v \in U$.

(2) Jeder Unterbaum von C ist ein (B.T.).

(3) Jeder einelementige Baum ist ein (B.T.).

Mehlhorn [4] konnte durch eine elegante elementare Abschätzung zeigen, daß assymptotisch (für große U) gilt

$$M(C) \leq H(p) + 2 \quad \text{für } C \text{ ist ein (B.T.).}$$

Kürzlich zeigte Bayer [2], daß im gleichen Sinne gilt

$$H(p) - \log\log n \leq M(C),$$

so daß die Leistungsfähigkeit der (B.T.) als gut geklärt angesehen werden kann.

Das Ziel dieser Notiz besteht in folgendem: Es soll unter einer Voraussetzung über die Verteilung p eine verbesserte obere Schranke angegeben werden. Dies ist dadurch gerechtfertigt, daß der Beweis hierfür sehr kurz ist. Weiter soll gezeigt werden, daß es sinnvoll ist, auch k-näre Balanced Trees zu betrachten.

4. Einige einfache Abschätzungen

Sei C ein Suchbaum für U und sei

$$C = (-1)C_{-1} \cup C_0 \cup (1)C_1$$

die oben betrachtete Zerlegung: Wir ordnen den zwei Unterbäumen C_{-1} und C_1 wieder Wahrscheinlichkeitsverteilungen zu, indem wir für $C_{-1} \neq \emptyset$ und $C_1 \neq \emptyset$ setzen

$$p^{(1)}(u) = \frac{p(u)}{q_1} \quad \text{für} \quad u \in U_{-1}(u_0),$$

$$p^{(2)}(u) = \frac{p(u)}{q_2} \quad \text{für} \quad u \in U_1(u_0),$$

und

$$q_1 = p(U_{-1}(u_0)), \qquad q_2 = p(U_1(u_0)).$$

Nun gilt offenbar

$$M(C,p) = q_1(M(C_{-1},p^{(1)})+1)+q_2(M(C_1,p^{(2)})+1)+p(U_0)$$
$$= q_1M(C_{-1},p^{(1)})+q_2M(C_1,p^{(2)})+1. \tag{1}$$

Wir verwenden die für die Entropie bekannte Gleichung

$$H(p) = q_1H(p^{(1)})+q_2H(p^{(2)})+H(p(U_0),q_1,q_2) \tag{2}$$

und erhalten durch Subtraktion von (1) und (2),

$$M(C,p)-H(p)= \tag{3}$$
$$= q_1(M(C_{-1},p^{(1)})-H(p^{(1)}))+q_2(M(C_1,p^{(2)})-H(p^{(2)}))+1-H(p_0,q_1,q_2).$$

Für eine Verteilung p über großem U wird im Falle, daß C ein (B.T.) ist, im allgemeinen gelten für $p_0 = p(U_0)$,

$$q_1, q_2, p_0 \leqq \tfrac{1}{2}. \tag{4}$$

Wie man mittels elementaren Mitteln der Analysis zeigt, gilt

$$1-H(p_0,q_1,q_2) \leqq 0 \tag{5}$$

unter der Voraussetzung (4).

Damit erhält man

$$M(C,p)-H(p) \leqq$$
$$\leqq q_1(M(C_{-1},p^{(1)})-H(p^{(1)}))+q_2(M(C_1,p^{(2)})-H(p^{(2)})). \tag{6}$$

Definition. p heißt über $(U, \leqq)$ *wohlverteilt* $\Leftrightarrow$

(W1) Ist C ein B.T. zu $(U, \leqq, p)$ so gilt bei obiger Bezeichnung (4).

(W2) Jeder Unterbaum von C erfüllt (W1).

(W3) p ist über jede einelementige Menge wohlverteilt.

p ist also wohlverteilt, wenn kein einzelnes Element von U relativ zu seinen Nachbarn zu schwer ist.

Nun sieht man aus (6) sofort die Gültigkeit von

Satz 1. *Ist p über $(U, <)$ wohlverteilt und ist C ein (B.T.) zu $(U, <, p)$, dann gilt*

$$M(C,p) \leqq H(p).$$

Man sieht, daß dieser Satz noch assymptotisch gilt, wenn man die Bedingung (4) für kleine Unterbäume aufgibt.

5. k-näre Suchbäume

Das Kodierungstheorem legt eine Verallgemeinerung der bis hierhin behandelten Fragestellungen auf k-näre Kodes nahe. Es stellt sich die Frage, wie dann die

entsprechenden Abschätzungen lauten und ob diese Verallgemeinerung für Suchbäume sinnvoll ist. Wir verallgemeinern zuerst unseren Satz 1.

Sei $C \subset [1:k]^+$ ein endlicher Kode und sei

$$K_i = (i) \cdot C_i \quad \text{mit} \quad C_i = \{w \in [1:k]^* \mid (i) \cdot w \in C\}.$$

Offensichtlich gilt

$$K_i \cap K_j = \emptyset \quad \text{für} \quad i \neq j,$$

und

$$\dot{\bigcup_{i=1}^{k}} K_i = C.$$

Ist $p : C \to [0, 1]$ eine Wahrscheinlichkeitsverteilung und $p^{(i)}$ die durch p auf C_i bzw. K_i induzierte Verteilung, dann gilt für die mittlere Wortlänge $M(C, p)$ des Kodes:

$$M(C, p) = \sum_{i=1}^{k} q_i (M(C_i, p^{(i)}) + 1)$$

$$= \sum_{i=1}^{k} q_i M(C_i, p^{(i)}) + 1,$$

worin wir für $C_i = \{\varepsilon\}$ (ε leeres Wort in $[1:k]^*$), $M(C_i, 1) = 0$ verwenden. Weiter ist $q_i = p(C_i)$ gesetzt.

Verwendet man wieder die wohlbekannte Zerlegung

$$H(p) = H(q_1, \ldots, q_k) + \sum_{i=1}^{k} q_i H(p^{(i)}),$$

dann erhält man durch Subtraktion

$$M(C, p) - H(p) = \sum_{i=1}^{k} q_i (M(C_i, p^{(i)}) - H(p^{(i)})) + 1 - H(q_1, \ldots, q_k) \tag{7}$$

in Analogie zu (3). Nun gilt für

$$q_i \leqslant \frac{1}{m} \quad \text{für } i = 1, \ldots, k \text{ und } m \leqslant k, \tag{8}$$

entsprechend zu (5),

$$1 - \frac{H(q_1, \ldots, q_k)}{\log m} \leqslant 0. \tag{9}$$

Wir definieren nun rekursiv: p heißt über dem k-nären Kode *m-wohlverteilt*, wenn (W1), (W2) und (W3) gelten.

(W1) $p(C_i) \leqslant 1/m$ für $i = 1, \ldots, k$.

(W2) Die durch p auf C_i induzierte Verteilung ist m-wohlverteilt.

(W3) p ist über jedem Kode mit m oder weniger als m Elementen wohlverteilt.

Für m-wohlverteilte C gilt offensichtlich (9). Also schließen wir aus (7) und dem Kodierungstheorem auf den

Satz 2. *Ist C ein k-närer Kode, auf dem p m-wohlverteilt ist, dann gilt*

$$\frac{H(p)}{\log k} \leqq M(C,p) \leqq \frac{H(p)}{\log m}.$$

Sind n-näre Suchbäume für $n < 2$ sinnvoll?

Unter einem n-nären Suchbaum verstehen wir einen Graphen, dessen Knoten wie in Fig. 1 aussehen. Hierin sind $u_1, \ldots, u_{n-1}$ für jeden Knoten des Baumes fest ausgewählte Elemente und x ist ein Element aus U, das in dem Baum aufgesucht werden soll. Die je zwei verschiedenen Knoten zugeordenten Elementmengen sind paarweise fremd und jedes Element von U ist irgendeinem Knoten zugeordnet. Die Knoten, die einer Abfrage $x = u_i$ entsprechen, sind Endpunkte des Baumes.

Durch jeden solchen Baum wird jedem Element $u \in U$ eindeutig ein Weg von der Wurzel des Baumes zu dem mit $= u$ markierten Knoten des Baumes zugeordnet. Das heißt, daß dieser Baum zusammen mit den Markierungen U bijektiv auf einen k-nären Kode mit $k = 2n - 1$ abbildet.

Nun verlangen wir schließlich, daß diese Abbildung die Ordnung von $(U, \leqq)$ in die natürliche Ordnung des k-nären Kodes überträgt.

Ein solcher markierter Baum heißt ein m-närer Suchbaum für $(U, \leqq)$.

Eine *Elementaroperation* in einem Suchbaum bestehe in der Ermittlung der zu einem gegebenen $x \in U$ gehörigen von einem festen Knoten ausgehenden Strecke.

Bewerten wir die Elementaroperationen in allen Suchbäumen unabhängig von n gleich, dann gibt uns Satz 2 eine Auskunft über die mittlere Suchzeit bei m-wohlverteilten Wahrscheinlichkeitsmaßen p. Bei großen U scheint jede Wahl von $m \leqq n$ keine allzugroße Einschränkung darzustellen. Könnte man also diese Bewertung der Elementaroperationen annehmen, erhielte man gegenüber den binären Suchbäumen unter schwachen Voraussetzungen über die Wahrscheinlichkeitsverteilungen wesentliche Verbesserungen der mittleren Suchzeiten. Allerdings erscheint diese Gleichbewertung der Elementaroperationen nicht realistich. Die Elementaroperationen aber proportional zu m anzusetzen, heißt darauf zu verzichten, realistische Voraussetzungen ins Spiel zu bringen, wie wir unten sehen werden.

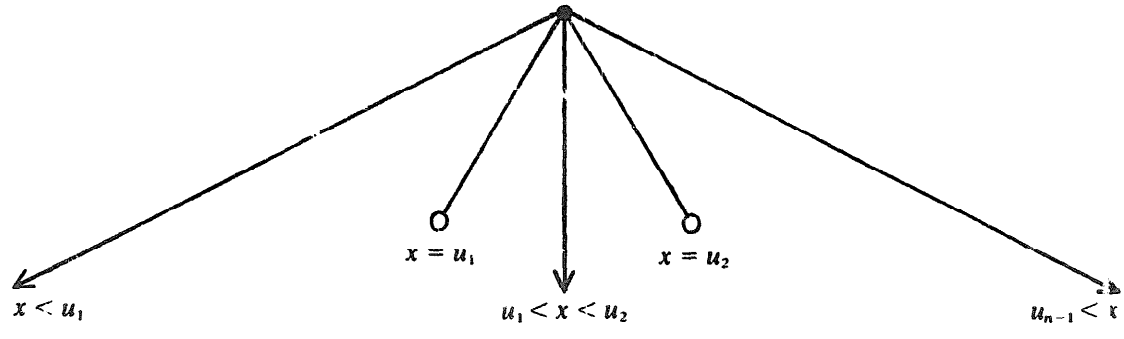

Fig. 1.

6. Zweistufige Zugriffsbäume

Wir nehmen nun an

$$U \subset A^*, \quad A \text{ endliche Menge.}$$

Ist P Knoten eines m-nären Suchbaumes zu $(U, \leqslant)$ und sind

$$U(P) = \{u_1, \ldots, u_{m-1}\}$$

die dem Knoten P zugeordneten Elemente von U, dann setzen wir die Kosten der Elementaroperation im Knoten P proportional zu

$$K(P) = \sum_{i=1}^{m-1} \text{Länge}(u_i)$$

an, worin Länge(u_i) die Länge des Wortes u_i über A angibt.

Weiter wollen wir annehmen, daß die Ordnung von U, die lexographische Ordnung über einer Ordnung von A ist.

Der Ansatz, $K(P)$ als Kosten der Elementaroperationen anzusehen ist dadurch gerechtfertigt, daß in einer Rechenanlage die Anzahl de Speicherzugriffe in erster Näherung proportionell zu $K(P)$ ist.

Wir wollen unter diesen Annahmen einen idealistischen Fall untersuchen, der zeigt, daß bei diesen Voraussetzungen ternäre Suchbäume den binären Suchbäumen sehr überlegen sind.

Sei $U \subset A^{2k}$ und $2k$ die Wortlänge einer Rechenanlage, auf der der Suchbaum implementiert werden soll. Eine direkte Implementierung des binären B.T. C_2 und eines ternären Suchbaumes C_3, so daß p über C_3 3-wohlverteilt ist, ergibt für die mittlere Suchzeit $S(C_2, p)$ bzw. $S(C_3, p)$ bei $K(P) = 1$ bzw. $K(P) = 2$,

$$S(C_2, p) = M(C_2, p),$$

$$S(C_3, p) = 2M(C_3, p),$$

$$H(p) - \log\log n \leqslant S(C_2, p) \leqslant H(P),$$

$$2\frac{H(p)}{\log 5} \leqslant S(C_3, p) \leqslant 2\frac{H(p)}{\log 3}.$$

Dies zeigt, daß hier die ternären Suchbäume keinen Vorteil vor den B.T. versprechen.

Wir betrachten nun die folgende Zerlegung:

$$U = \bigcup_{v \in V} v \cdot U_v$$

mit V, $U_v \subset A^*$ für $v \in V$ und

$$\text{Länge}(v) = \text{Länge}(w) = k \quad \text{für } v \in V,\ w \in U_v.$$

Die Anordnung von A übertragen wir lexikographisch auf V und U_v ($v \in V$), so daß für $v \neq v'$ gilt

$$vw \leqq v'w' \Leftrightarrow vw_1 \leqq v'w_1' \quad \text{für } w, w_1 \in U_v,\ w', w_1' \in U_v'.$$

Wir definieren

$$q : V \to [0, 1]$$

indem wir setzen

$$q(v) = p(vU_v) \quad \text{für} \quad v \in V$$

und

$$p_v : U_v \to [0:1]$$

durch

$$p_v(u) = p(vu)/q(v).$$

Nun gilt, wie aus der Informationstheorie bekannt ist,

$$H(p) = H(q) + \sum_{v \in V} q(v) H(p_v).$$

Bauen wir nun einen ternären Suchbaum für $(V, \leqq, q)$ auf und ternäre Suchbäume für $(U_v, \leqq, p_v)$, dann können wir die zu jeweils einem Knoten gehörigen Elemente von V bzw. U_v in einem Maschinenwort abspeichern und also für diese Suchbäume $K(p) \leqq 1$ ansetzen. Damit erhalten wir

Satz 3. *Bei der oben angegebenen Zerlegung von $U \subset A^{2k}$ und bei einer Verteilung p mit 3-wohlverteilten q und p_v ergibt sich für die mittlere Suchzeit S*

$$\frac{H(p)}{\log 5} \leqq S \leqq \frac{H(p)}{\log 3}.$$

Dieser Satz zeigt, daß so angelegte ternäre Suchbäume um den Faktor log 3 den B.T. überlegen sind.

7. Schlußbemerkung

(1) Die Voraussetzung m-wohlverteilt ist bei großen U und nicht zu sehr überwiegenden Einzelelementen erfüllt.

(2) Unsere Voraussetzung $U \subset A^{2k}$ ist idealistisch. Aber der Effekt unseres Verfahrens ist in diesem Falle so deutlich, daß zu erwarten ist, daß auch eine genaue Analyse realistischer Fälle einen Vorteil für die ternären Suchbäume ergibt.

(3) Die für m-näre Suchbäume erklärte Elementaroperation rechtfertigt in Verbindung mit Satz 3 für Rechenanlagen einen Hardware-Zusatz der zumindest

für $m = 2$ in einem Rechenschritt diese Operation durchführt. Ein solcher Zusatz ist nicht aufwendig und, wie gezeigt, für viele Anwendungen von Vorteil.

Literatur

[1] R. B. Ash, *Information Theory* (Wiley, New York).
[2] P. J. Bayer, Improved bounds on the costs of optimal and balanced binary search trees, S.B., Massachusetts Institute of Technology, Cambridge, Mass. (1973).
[3] D. Knuth, Optimum binary search trees, *Acta Informat.* 1 (1971) 14–25.
[4] K. Mehlhorn, Nearly optimal binary search trees, TB A 75/05, Universität des Saarlandes, Saarbrücken (1975).