Mathematische Grundlagen der Künstlichen Intelligenz: K-reguläre Minimax-Bäume mit distinkten Blattwerten, die bei beliebiger Sortierung „leicht" für den α–β–Algorithmus sind

Bernhard Balkenhol
Universität Bielefeld
Fakultät für Mathematik
Postfach 8640
W – 4800 Bielefeld 1
Germany

1992

Inhaltsverzeichnis

1 Einleitung

Der α–β–Algorithmus ist eine effiziente Technik, um Spielbäume zu durchsuchen.
Knuth und Moore zeigten ([KM 75]), daß der α–β–Algorithmus in einem k–regulären Minimaxbaum der Tiefe n im günstigsten Fall $k^{\lfloor \frac{n}{2} \rfloor} + k^{\lceil \frac{n}{2} \rceil} - 1$ und im ungünstigsten Fall k^n Blattwerte liest. Sie haben in [KM 75, Seite 310] ein Beispiel eines vollständigen, binären Spielbaumes der Tiefe 4 mit paarweise verschiedenen Blattwerten angegeben, in welchem der α–β–Algorithmus maximal 14 Blattwerte liest, unabhängig davon, wie die Zweige des Baumes sortiert sind. Im allgemeinen ist die Laufzeit stark abhängig von der Ordnung (oder Sortierung) des Baumes ([Mar 86, Seite 11] und [Mar 87]).
Seit einigen Jahren gewinnen parallele Suchalgorithmen für Spielbäume immer mehr an Bedeutung. Es stellt sich die Frage, ob auf beliebigen Spielbäumen ein linearer Speed–Up erzielt werden kann. Um diese Frage beantworten zu können, untersucht man zunächst k–reguläre Minimaxbäume. In [KZ 89, Seiten 409–420] wurde im Mai 1989 die Vermutung geäußert, daß der von Karp und Zhang entwickelte Parallel–Algorithmus für zweiwertige Spielbäume auch auf vielwertigen Minimaxbäumen einen linearen Speed–Up erzielen könne. Ein halbes Jahr später (Dezember 1989) widerlegte Zhang diese Vermutung, indem er ein Gegenbeispiel präsentierte ([Zha 89]). Es ist ein offenes Problem, ob auf diesen Bäumen ein linearer Speed–Up erzielt werden kann. (Zu parallelen Spielbaum–Algorithmen siehe auch das neuere Manuskript [BKRU 90])
Die für sequentielle Algorithmen besonders guten Bäume müssen nicht für parallele Algorithmen geeignet sein. Diese Bäume könnten daher zum Testen paralleler Algorithmen benutzt werden.
Betrachtet man bei einem k–regulären Minimaxbaum die von dem α-β-Algorithmus gelesenen Blattwerte, so ist deren Anzahl offenbar abhängig von den Blattwerten selbst und von der Sortierung des Baumes.
Um eine obere Schranke für diese Anzahl zu konstruieren, betrachte man die Sortierung eines jeden Baumes, bei der die Anzahl gelesener Blattwerte maximal wird. Eine solche Sortierung heißt im folgenden **ungünstigste Sortierung**.
Die Abbruchbedingung des Alpha–Beta–Algorithmus ist: $\alpha \geq \beta$.
Betrachtet man einen Minimaxbaum, bei dem alle Blattwerte gleich sind, so wird die von Knuth und Moore angegebene Anzahl Blattwerte gelesen (sie-

he oben: untere Schranke von Knuth und Moore). Alpha kann beim Lesen eines Knotenwertes nur dann echt kleiner bleiben als Beta, wenn der gelesene Wert echt kleiner ist als Alpha oder Alpha auf einen Wert erhöht wird, der immer noch kleiner ist als Beta. Um auch hier eine obere Schranke für die Anzahl gelesener Blattwerte angeben zu können, seien alle Blattwerte distinkt. Die so gefundene obere Schranke gilt dann auch für Minimaxbäume, deren Blattwerte nicht alle paarweise verschieden sind.
Gesucht wird ein Baum mit distinkten Blattwerten, bei dem sich bei ungünstigster Sortierung eine minimale Anzahl gelesener Blattwerte ergibt. Diese Bäume heißen **optimal**.

Für den einfachsten, nichttrivialen Fall (binäre Bäume) wurde in [AB 90] eine Baumfolge konstruiert. Durch deren Analyse wurde eine obere Schranke bewiesen. Dieses Ergebnis wird in dieser Arbeit auf den k–regulären Fall ausgedehnt.

Die zentrale Frage dieser Arbeit lautet also: Wie groß ist die minimale Anzahl Blattwerte, die der α-β-Algorithmus bei einem optimalen Minimaxbaum der Tiefe n bei ungünstigster Sortierung liest?
In Kapitel 2 wird folgende obere Schranke bewiesen:

Theorem 1: Für jedes $n \geq 0$ existiert ein k - regulärer Minimax-Baum der Tiefe n mit distinkten Blattwerten, in welchem der α–β–Algorithmus, unabhängig von der Sortierung, höchstens

$$\left(\frac{(k-1)^2 \cdot n^2}{8 \cdot k} + \left(\frac{k-1}{2} - \frac{(k-1)^2}{4 \cdot k}\right) \cdot n + 1 \right) \cdot k^{\frac{n}{2}}$$

Blattwerte liest, falls n gerade ist und

$$\left(\frac{(k-1)^2 \cdot n^2}{8 \cdot k} + \left(\frac{k-1}{2} - \frac{(k-1)^2}{2 \cdot k}\right) \cdot n + 1 + \frac{3 \cdot (k-1)^2}{8 \cdot k} - \frac{k-1}{2} \right) \cdot k^{\frac{n+1}{2}} \quad ,$$

falls n ungerade ist.

Dabei wird in den Abschnitten 2.1 bis 2.3 ein Konstruktionsverfahren vorgestellt und in Abschnitt 2.4 der mit diesem Konstruktionsverfahren erstellte Baum bezüglich der Anzahl gelesener Blattwerte untersucht.

Um die Qualität dieser Aussage einschätzen zu können, wird in Kapitel 3 eine für den α–β–Algorithmus schlechte Sortierung betrachtet. Mittels dieser Sortierung wird eine untere Schranke für die Anzahl gelesener Blattwerte ermittelt, nämlich

Theorem 2: Bei jedem k–regulären Baum der Tiefe n mit distinkten Blattwerten liest der α–β–Algorithmus bei ungünstigster Sortierung mindestens

$$(3 \cdot k + 3) \cdot k^{\frac{n}{2}} - \frac{(k-1)^2}{4} \cdot n^2 - \frac{k^2 + 2 \cdot k - 3}{2} \cdot n - 3 \cdot k - 2$$

Blattwerte, falls n gerade ist und mindestens

$$(k^2 + 4 \cdot k + 1) \cdot k^{\frac{n-1}{2}} - \frac{(k-1)^2}{4} \cdot n^2 - \frac{k^2 + 2 \cdot k - 3}{2} \cdot n - \frac{k^2 + 10 \cdot k + 9}{4}$$

Blattwerte, falls n ungerade ist.

Es verbleibt das Problem, die „Lücke“ zwischen Theorem 1 und Theorem 2 zu schließen. Dazu wird in Kapitel 4 für k–reguläre Bäume ein weiteres Ergebnis (Theorem 3) bewiesen, in dem eine Aussage über die Qualität der in Theorem 1 gefundenen oberen Schranke getroffen wird.

Theorem 3: Sei $h(n)$ eine obere Schranke für die von dem α–β–Algorithmus bei einem optimalen Minimaxbaum mit gerader Tiefe n gelesene Anzahl Blattwerte. Sei $h(n)$ für gerade n von der Form $p(n) \cdot k^{\frac{n}{2}}$, wobei p(n) ein Polynom ist, und sei die obere Schranke aus Theorem 1 mit $g(n)$ bezeichnet. Gilt $h(2 \cdot n) \leq g(2 \cdot n)$ für alle $n \geq 0$, so ist $h(2 \cdot n) = g(2 \cdot n)$ für alle $n \geq 0$.

Im weiteren Verlauf dieser Arbeit werden **binäre** Minimaxbäume genauer behandelt. So wird in Kapitel 5 Abschnitt 1 ein Minimaxbaum der Tiefe 5 analysiert und gezeigt, daß die in Theorem 1 angegebene obere Schranke (für $k = 2$ und $n \geq 5$) echt größer ist, als die Anzahl gelesener Blattwerte bei einem optimalen Baum. In Abschnitt 5.2 wird dann weiter gezeigt, daß auch die durch Theorem 2 gegebene untere Schranke zu klein sein kann. Man kann also lediglich sagen, daß die Anzahl gelesener Blattwerte bei einem optimalen binären Minimaxbaum im ungünstigsten Fall größer oder gleich der durch Theorem 2 gegebenen unteren Schranke ist und echt kleiner als die durch Theorem 1 gegebene oberen Schranke.

In Theorem 4 aus Kapitel 6 Abschnitt 5 wird die obere Schranke für die Anzahl gelesener Blattwerte auf eine Größenordnung von $c \cdot 2^{\frac{n}{2}}$ reduziert, wobei c eine von der Tiefe unabhängige Konstante ist.

Theorem 4: Zu jedem $n \geq 0$ existiert ein binärer Minimaxbaum M der Tiefe n mit distinkten Blattwerten, in welchem die Anzahl der von dem α–β–Algorithmus gelesenen Blattwerte echt kleiner ist als $43 \cdot 2^{\frac{n}{2}}$, falls n gerade ist und echt kleiner als $61 \cdot 2^{\frac{n-1}{2}}$, falls n ungerade ist.

Als Fazit läßt sich festhalten, daß die „Lücke“ zwischen oberer und unterer Schranke sich im binären Fall lediglich auf den Unterschied zwischen den multiplikativen Konstanten c reduzieren läßt.
(43 statt 9 für gerade n und 61 statt 13 für ungerade n)
In Abschnitt 6.6 wird schließlich der binäre Minimaxbaum M aus Theorem 4 mit einer ganzzahligen Blattwertbelegung erzeugt.

Besonders danken möchte ich Herrn Dr. Ingo Althöfer für die Anregung zu dieser Arbeit und für seine ausgesprochen motivierende Betreuung.

2 Konstruktion eines Baumes

Theorem 1: Für jedes $n \geq 0$ existiert ein k - regulärer Minimax-Baum der Tiefe n mit distinkten Blattwerten, in welchem der α–β–Algorithmus, unabhängig von der Sortierung, höchstens

$$\left(\frac{(k-1)^2 \cdot n^2}{8 \cdot k} + (\frac{k-1}{2} - \frac{(k-1)^2}{4 \cdot k}) \cdot n + 1 \right) \cdot k^{\frac{n}{2}}$$

Blattwerte liest, falls n gerade ist und

$$\left(\frac{(k-1)^2 \cdot n^2}{8 \cdot k} + (\frac{k-1}{2} - \frac{(k-1)^2}{2 \cdot k}) \cdot n + 1 + \frac{3 \cdot (k-1)^2}{8 \cdot k} - \frac{k-1}{2} \right) \cdot k^{\frac{n+1}{2}} \quad ,$$

falls n ungerade ist.

Der Beweis wird in vier Abschnitte (2.1 bis 2.4) eingeteilt. Dabei wird in den Abschnitten 2.1 bis 2.3 ein Konstruktionsverfahren geliefert, und durch die Analyse des so erzeugten Baumes wird in Abschnitt 2.4 Theorem 1 bewiesen.

2.1 Bemerkung

Konstruiert wird ein für den α–β–Algorithmus „leichter" Minimaxbaum mit Hilfe eines Top–down–Schemas (von der Wurzel zu den Blättern). Jedem Knoten u werden dabei fünf reelle Werte a_u, b_u, c_u, d_u, e_u mit

$$a_u < b_u < c_u < d_u < e_u \tag{1}$$

zugeordnet. a_u, b_u, d_u und e_u sind lediglich Hilfsvariablen. c_u wird der spieltheoretische Wert des Knotens u (Minimaxwert).
Für eine gerade Tiefe n ist die Wurzel des Baumes ein Min–Knoten. Knoten mit geradem Abstand zu den Blättern sind Min–Knoten, Knoten mit ungeradem Abstand zu den Blättern sind Max–Knoten.
Ist n ungerade, so wird zunächst ein Minimaxbaum mit Min–Wurzel und Max–Knoten an den Blättern betrachtet. Die für diesen Baum gefundene Abschätzung gilt dann auch für einen inversen Baum, d.h. mit Max–Wurzel und Min–Knoten an den Blättern (siehe dazu 4.3). Sei

$$X_u := \{x | a_u < x < b_u \quad \vee \quad x = c_u \quad \vee \quad d_u < x < e_u\}$$

und seien $v_1, \ldots, v_k$ die Nachfolger von u.

Dann soll gelten:

1. Die Werte eines jeden Knotens u erfüllen die Bedingung (1)
2. $X_{v_i} \cap X_{v_j} = \emptyset \quad \forall \quad i \neq j$
3. $X_{v_1} \cup \ldots \cup X_{v_k} \subset X_u$

Zusammen mit der Tatsache, daß $c_u \in X_u$ für jeden Knoten u ist, wird garantiert, daß die Blattwerte distinkt sind. Die Wurzel r des Baumes sei ein Min–Knoten mit den Werten $a_r = 0$, $b_r = 1$, $c_r = 2$, $d_r = 3$ und $e_r = 4$

2.2 Rekursive Definition für Min-Knoten

Sei ein Min-knoten u mit Werten a_u, b_u, c_u, d_u, e_u gegeben. Dann erhalten die k Nachfolger folgende Werte ($a_i, b_i, \ldots, e_i$ sind die Werte des i-ten besten Nachfolgers von u):

$$
\begin{aligned}
a_1 &= a_u, \qquad b_1 = b_u \quad , \qquad\qquad c_1 = c_u \\
a_i &= \frac{5\cdot k+2-3\cdot i}{5\cdot k-4} \cdot d_u + \frac{3\cdot i-6}{5\cdot k-4} \cdot e_u \quad \forall \quad 2 \leq i \leq k \\
b_i &= \frac{5\cdot k+1-3\cdot i}{5\cdot k-4} \cdot d_u + \frac{3\cdot i-5}{5\cdot k-4} \cdot e_u \quad \forall \quad 2 \leq i \leq k \\
c_i &= \frac{5\cdot k-3\cdot i}{5\cdot k-4} \cdot d_u + \frac{3\cdot i-4}{5\cdot k-4} \cdot e_u \quad \forall \quad 2 \leq i \leq k \\
d_i &= \frac{2\cdot i-1}{5\cdot k-4} \cdot d_u + \frac{5\cdot k-3-2\cdot i}{5\cdot k-4} \cdot e_u \quad \forall \quad 1 \leq i \leq k \\
e_i &= \frac{2\cdot i-2}{5\cdot k-4} \cdot d_u + \frac{5\cdot k-2-2\cdot i}{5\cdot k-4} \cdot e_u \quad \forall \quad 1 \leq i \leq k
\end{aligned}
$$

also:

$\Big((a_u = a_1) < (b_u = b_1) < (c_u = c_1)\Big) < \Big((d_u = a_2) < b_2 < c_2\Big) < (a_3 < b_3 < c_3) < \ldots < (a_k < b_k < c_k) < (d_k < e_k) < \ldots < (d_2 < e_2) < d_1 < (e_1 = e_u)$

2.3 Rekursive Definition für Max-Knoten

Sei nun ein Max-Knoten u mit Werten $a_u, b_u, \ldots, e_u$ gegeben.
Seien $(a_i, b_i, \ldots, e_i)$ die Werte des i-ten besten Nachfolgers von u. Wir setzen:

$$
\begin{array}{lclclcll}
c_1 & = & c_u & & & & & \\
a_i & = & \frac{3\cdot k+1-2\cdot i}{3\cdot k-1} & \cdot a_u & + & \frac{2\cdot i-2}{3\cdot k-1} & \cdot b_u & \forall \quad 1 \leq i \leq k \\
b_i & = & \frac{3\cdot k-2\cdot i}{3\cdot k-1} & \cdot a_u & + & \frac{2\cdot i-1}{3\cdot k-1} & \cdot b_u & \forall \quad 1 \leq i \leq k \\
c_i & = & \frac{i-1}{3\cdot k-1} & \cdot a_u & + & \frac{3\cdot k-i}{3\cdot k-1} & \cdot b_u & \forall \quad 2 \leq i \leq k \\
d_i & = & \frac{2\cdot i-1}{2\cdot k-1} & \cdot d_u & + & \frac{2\cdot k-2\cdot i}{2\cdot k-1} & \cdot e_u & \forall \quad 1 \leq i \leq k \\
e_i & = & \frac{2\cdot i-2}{2\cdot k-1} & \cdot d_u & + & \frac{2\cdot k-2\cdot i+1}{2\cdot k-1} & \cdot e_u & \forall \quad 1 \leq i \leq k
\end{array}
$$

also:

$$
\begin{array}{l}
a_u < (a_1 < b_1) < (a_2 < b_2) < \ldots < (a_k < b_k) < c_k < \ldots < c_2 < b_u \\
< (c_1 = c_u) < d_u < (d_k < e_k) < \ldots < (d_2 < e_2) < (d_1 < e_1) < e_u
\end{array}
$$

2.4 Analyse des Baumes

Nun können beim Durchlauf des α-β-Algorithmus drei verschiedene Fälle eintreten:

Nr.	Beta
0	$e < \beta$
1	$c < \beta < d$
2	$\beta < a$

Sei nun für $0 \leq i \leq 2 \quad m_i(t)$ die größte Anzahl Abfragen, die der α-β-Algorithmus beim Durchlaufen eines Baumes der Tiefe $2 \cdot t$ vornimmt. Entsprechend sei $M_i(t) \quad \forall \quad 0 \leq i \leq 2$ die größte Anzahl Abfragen bei einer Tiefe von $2 \cdot t + 1$. Damit ist dann $m_i(0) = 1 \quad \forall \quad 0 \leq i \leq 2$.

Mit dieser Notation ergeben sich folgende Bedingungen:

1. $M_2(t) = m_2(t)$
   Beweis: Sei j der linke Teilbaum von den k vielen.
   $\Rightarrow \alpha < \beta < a_u \leq a_j$ d.h. $m_2(t)$ tritt als Fall ein.
   Nun wird $\alpha(\text{neu}) = c_j > a_u > \beta$, d.h. der α-β-Algorithmus bricht an dieser Stelle ab, unabhängig davon, welchen Wert j $(1 \leq j \leq k)$ zuerst angenommen hat.

2. $m_2(t+1) = k \cdot M_2(t)$
   Beweis: $\alpha < \beta < a_u = a_1$. Da a_1 das Minimum aller a_i, b_i, c_i, d_i, e_i genommen über alle $1 \leq i \leq k$ ist, folgt, daß β beim Durchlaufen der Teilbäume nicht verkleinert werden kann. D.h. β bleibt beim Durchlaufen der Teilbäume unverändert (unabhängig von der Reihenfolge). Da α auch nicht verändert wird, folgt für jeden der k Teilbäume $M_2(t)$ als eintretender Fall. D.h. $m_2(t+1) = k \cdot M_2(t)$.

3. Auswertung für $M_2(t)$ und $m_2(t)$

   $$M_2(t) = m_2(t) = k^t$$

   Beweis: Aus den Ergebnissen aus 1. und 2. folgt:

   $m_2(t+1) = k \cdot m_2(t) \qquad \Rightarrow m_2(t) = k^t$

   Beweis durch Induktion nach t:
   $t = 0$: Es gilt $m_i(0) = 1 \quad \forall \quad 0 \leq i \leq 2$ (siehe oben)

   $k^0 = 1 = m_2(0)$

   Induktionsschritt: $t \Rightarrow t+1$
   Annahme, die Behauptung sei für t bewiesen.

   $$\begin{aligned} \Rightarrow m_2(t+1) &= k \cdot m_2(t) \text{ (nach Voraussetzung)} \\ &= k \cdot k^t \text{ (nach Induktionsannahme)} \\ &= k^{(t+1)} \text{ q.e.d.} \end{aligned}$$

4. $M_1(t) = k \cdot m_1(t)$
   Beweis: Sei nun $\alpha < \beta$ und $c_u < \beta < d_u$.
   Es gilt : $c_j \leq c_1 = c_u < \beta < d_u < d_k \leq d_j \quad \forall \quad 1 \leq j \leq k$. Da also β unverändert bleibt und alle $c_j < \beta \quad \forall \quad 1 \leq j \leq k$, kann α nicht größer werden als β. D.h. für alle k Teilbäume tritt der Fall $m_1(t)$ ein, d.h. $M_1(t) = k \cdot m_1(t)$.

5. $m_1(t+1) = M_1(t) + (k-1) \cdot M_2(t)$
   Beweis: Da der ungünstigste Fall untersucht werden soll, kann α so gewählt werden, daß es möglichst spät zu einer Abbruchbedingung kommt. Z.B. sei $\alpha < a_u$, so daß $m_1(t)$ möglichst groß wird.

   Es gilt nun weiter $c_1 = c_u < \beta < d_u = a_2$. Solange $j > 1$ gilt $a_j > d_u > \beta$, d.h. $M_2(t)$. Tritt nun der Fall $j = 1$ auf, so gilt für diesen Fall $c_j < \beta < d_j$, d.h. $M_1(t)$. Anschließend wird β(neu) $=c_j$. Für den Fall, daß $\alpha < c_1$ ist, folgt für alle weiteren j, daß $a_j > d_u > \beta$, d.h. $M_2(t)$. Für den Fall, daß $\alpha \geq c_1$ ist, gilt : β(neu) $< \alpha$, d.h. der α-β-Algorithmus bricht ab. Um in diesem zweiten Fall $m_1(t)$ möglichst groß werden zu lassen, muß man möglichst viele Summanden haben. D.h. der ungünstigste Fall tritt dann ein, wenn $j = 1$ der letzte aller k Teilbäume ist. Es folgt also im ungünstigsten Fall für $m_1(t)$:

$$m_1(t+1) = M_1(t) + (k-1) \cdot M_2(t).$$

6. Auswertung für $M_1(t)$ und $m_1(t)$

   Aus 3., 4. und 5. folgt:

   $m_1(t+1) = k \cdot m_1(t) + (k-1) \cdot k^t$

   $\Leftrightarrow m_1(t+1) - k \cdot m_1(t) = (k-1) \cdot k^t$

   Behauptung: $m_1(t) = (\frac{k-1}{k} \cdot t + 1) \cdot k^t$

   Beweis durch Induktion nach t: Induktionsvoraussetzung: $t = 0$:

   $m_1(0) = (\frac{k-1}{k} \cdot 0 + 1) \cdot k^0 = 1 \cdot k^0 = 1$

   Induktionsschritt $t \Rightarrow t+1$:
   Annahme, die Behauptung sei bereits für t bewiesen.

$$\begin{aligned}
\Rightarrow m_1(t+1) &= k \cdot m_1(t) + (k-1) \cdot k^t \\
&= k \cdot (\tfrac{k-1}{k} \cdot t + 1) \cdot k^t + (k-1) \cdot k^t \\
&= ((k-1) \cdot t + k) \cdot k^t + (k-1) \cdot k^t \\
&= ((k-1) \cdot t + (k-1) + k) \cdot k^t \\
&= ((k-1) \cdot (t+1) + k) \cdot k^t \\
&= k \cdot (\tfrac{k-1}{k} \cdot (t+1) + 1) \cdot k^t \\
&= (\tfrac{k-1}{k} \cdot (t+1) + 1) \cdot k^{(t+1)}
\end{aligned}$$

q.e.d

Aus 4. folgt: $M_1(t) = (\frac{k-1}{k} \cdot t + 1) \cdot k^{(t+1)}$.

7. $M_0(t) = k \cdot m_0(t)$
   Beweis: Sei $\alpha < \beta$ und $e_u < \beta$. Es gilt $c_j < e_u < \beta \quad \forall \quad 1 \leq j \leq k$. Da also β unverändert bleibt, α zu Beginn kleiner ist als β und es kein c_j gibt mit $\beta < c_j$, folgt für alle k Teilbäume der Fall $m_0(t)$. D.h. $M_0(t) = k \cdot m_0(t)$.

8. $m_0(t+1) = M_0(t) + (k-1) \cdot M_1(t)$
   Zwischenbemerkung:
   Aus 3. und 6. folgt, daß $m_1(t) > m_2(t) \quad \forall \quad t > 0$
   $\Leftrightarrow k \cdot m_1(t) > k \cdot m_2(t) \Leftrightarrow M_1(t) > M_2(t)$ $(\forall t > 0)$

   Beweis von 8. Es gilt: $\alpha < \beta$, $e_u < \beta$. Sei j der linke Teilbaum $\Rightarrow e_j < \beta \Rightarrow M_0(t) \Rightarrow \beta(neu) = c_j$. Da gilt $c_j < d_u \quad \forall \quad 1 \leq j \leq k$, folgt, es können nur noch folgende drei Fälle eintreten:

   (a) $M_1(t)$ tritt als Summand auf,
   (b) $M_2(t)$ tritt als Summand auf,
   (c) der α-β-Algoritmus bricht ab.

   $m_0(t)$ wird aber nur dann maximal, wenn die Anzahl der Summanden maximal wird, und nach der Zwischenbemerkung, wenn möglichst oft $M_1(t)$ anstatt $M_2(t)$ als Summand auftritt. Sei nun die Reihenfolge der Teilbäume:
   $k, k-1, k-2, \ldots, 3, 2, 1 \Rightarrow$ Da $c_k > c_{k-1} > \ldots > c_3 > c_2 > c_1$ und im ungünstigsten Fall $\alpha < c_1$ ist (der α-β- Algorithmus beim

Durchlaufen der Teilbäume also nicht abbricht), ergeben sich nun folgende Summanden: Für $j = k$ ist $e < \beta$, also folgt $M_0(t)$. Nun wird β(neu)=c_k. Für jeden Nachfolger $j-1$ von j ($\forall \quad 2 \leq j \leq k$) folgt: $c_{j-1} < c_j = \beta < d_{j-1}$, d.h. $M_1(t)$, d.h. $m_0(t{+}1) = M_0(t){+}(k{-}1){\cdot}M_1(t)$.

9. Auswertung für $M_0(t)$ und $m_0(t)$
   Aus 4., 7. und 8. folgt:

$$\begin{aligned} m_0(t+1) &= k \cdot m_0(t) + (k-1) \cdot k \cdot m_1(t) \qquad \text{nach 6.} \\ &= k \cdot m_0(t) + (k-1) \cdot k \cdot (\tfrac{k-1}{k} \cdot t + 1) \cdot k^t \end{aligned}$$

   Behauptung:

$$m_0(t) = (\tfrac{(k-1)^2}{2\cdot k} \cdot t^2 + (k-1-\tfrac{(k-1)^2}{2\cdot k}) \cdot t + 1) \cdot k^t$$

   Beweis durch Induktion nach t: Für $t = 0$ gilt:

   $m_0(0) = 1 \cdot k^0 = 1$

   Induktionsschritt $t \Rightarrow t+1$
   Annahme, die Behauptung sei für t bewiesen.

$$\begin{aligned} \Rightarrow m_0(t+1) &= k \cdot m_0(t) + k \cdot (k-1) \cdot (\tfrac{k-1}{k} \cdot t + 1) \cdot k^t \\ &= (\tfrac{(k-1)^2}{2} \cdot t^2 + ((k-1) \cdot k - \tfrac{(k-1)^2}{2}) \cdot t + k) \cdot k^t + ((k-1)^2 \cdot t \\ &\quad +k \cdot (k-1)) \cdot k^t \\ &= (\tfrac{(k-1)^2}{2} \cdot t^2 + ((k-1) \cdot k \cdot t + (k-1) \cdot k + \tfrac{(k-1)^2}{2} \cdot t + k) \cdot k^t \\ &= m_0(t+1) + \tfrac{(k-1)^2}{2} - \tfrac{(k-1)^2}{2} \\ &= (\tfrac{(k-1)^2}{2} \cdot (t+1)^2 + ((k-1) - \tfrac{(k-1)^2}{2}) \cdot (t+1) + k) \cdot k^t \\ &= (\tfrac{(k-1)^2}{2\cdot k} \cdot (t+1)^2 + ((k-1) - \tfrac{(k-1)^2}{2\cdot k}) \cdot (t+1) + 1) \cdot k^{(t+1)} \end{aligned}$$

   q.e.d.

Für einen k - regulären Baum der Tiefe n folgt nun das Theorem für gerade n aus $m_0(\frac{n}{2})$ und für ungerade n aus $M_0(\frac{n-1}{2})$.

3 Beweis einer unteren Schranke

Theorem 2: Bei jedem k–regulären Baum der Tiefe n mit distinkten Blattwerten liest der α–β–Algorithmus bei ungünstigster Sortierung mindestens

$$(3 \cdot k + 3) \cdot k^{\frac{n}{2}} - \frac{(k-1)^2}{4} \cdot n^2 - \frac{k^2 + 2 \cdot k - 3}{2} \cdot n - 3 \cdot k - 2$$

Blattwerte, falls n gerade ist und mindestens

$$(k^2 + 4 \cdot k + 1) \cdot k^{\frac{n-1}{2}} - \frac{(k-1)^2}{4} \cdot n^2 - \frac{k^2 + 2 \cdot k - 3}{2} \cdot n - \frac{k^2 + 10 \cdot k + 9}{4}$$

Blattwerte, falls n ungerade ist.

Um Theorem 2 zu beweisen, werden zunächst acht Fälle konstruiert. Diese werden in Abschnitt 3.1 abgeschätzt und die Ergebnisse werden in Abschnitt 3.2 analysiert.
Sei T ein k-regulärer Baum. T ist für den α–β–Algorithmus schlecht geordnet, wenn gilt:

1. Die Nachfolger eines Min-Knotens sind in absteigender Reihenfolge angeordnet
2. Die Nachfolger eines Max-Knotens sind in aufsteigender Reihenfolge angeordnet

(Diese Darstellung nennt man auch „worst–first“–Darstellung). Nun können folgende acht Fälle auftreten. Sei x der jeweilige Wert des Knotens

Fall	Alpha	Beta
0	$\alpha < x$	$\beta > x$
1	$\alpha > x$	$\beta > x$
2	$\alpha < x$	$\beta < x$
3	$\alpha < x$	$\beta = +\infty$
4	$\alpha > x$	$\beta = +\infty$
5	$\alpha = -\infty$	$\beta < x$
6	$\alpha = -\infty$	$\beta > x$
7	$\alpha = -\infty$	$\beta = +\infty$

3.1 Abschätzung der Fälle

Sei $l_i(t)$ die Anzahl Abfragen von Min-Knoten der Tiefe $2 \cdot t$ und $L_i(t)$ die Anzahl Abfragen von Max-Knoten der Tiefe $2 \cdot t + 1 \; \forall \quad 0 \leq i \leq 7$.

Fall 0: Die Nachfolger eines Min-Knoten mit dem Wert x sind in absteigender Reihenfolge geordnet. Der letzte Nachfolger hat demnach den kleinsten Wert und dieser ist somit gleich x. Da nun $\alpha < x$ ist, ist α kleiner als alle Nachfolger des Min-Knotens x. β kann nun größer oder kleiner sein als der erste Wert, d.h. $L_0(t)$ oder $L_2(t)$. Falls β größer ist als der erste Wert, so ist er auch größer als alle weiteren Werte (da die Nachfolger in absteigender Reihenfolge), d.h. in den $k-1$ verbleibenden Fällen folgt $L_0(t)$. Falls β kleiner ist als der erste Wert, so kann es entweder größer oder kleiner sein als der nächste Wert, usw. D.h. solange $L_2(t)$ folgt, kann auch weiter $L_2(t)$ oder $L_0(t)$ eintreten, sobald aber $L_0(t)$ eintritt, kann auch nur $L_0(t)$ folgen. Da β jedoch nicht kleiner werden kann als x und β nach Voraussetzung größer war als x, gilt für den letzten der k Nachfolger in jedem Fall $L_0(t)$. Entsprechend gilt für einen Max-Knoten im Fall 0: solange $l_1(t)$ folgt, kann auch wieder $l_0(t)$ oder $l_1(t)$ folgen; nachdem $l_0(t)$ aufgetreten ist, kann auch nur noch $l_0(t)$ folgen und im letzten Fall gilt immer $l_0(t)$.

Fall 1: Für $l_1(t+1)$ kann im ersten Fall $L_0(t)$, $L_1(t)$ oder $L_2(t)$ eintreten. Nach $L_0(t)$ kann $L_0(t)$ oder $L_1(t)$ folgen, nach $L_2(t)$ kann $L_0(t)$, $L_1(t)$ oder $L_2(t)$ folgen. Nach $L_1(t)$ bricht der α-β-Algorithmus ab. Für $L_1(t)$ gilt in jedem der k eintretenden Fälle $l_1(t)$.

Fall 2: Für $l_2(t+1)$ gilt in jedem der k nachfolgenden Fälle $L_2(t)$. Für $L_2(t)$ gilt im ersten Fall $l_0(t)$,$l_1(t)$ oder $l_2(t)$. Nach $l_0(t)$ kann $l_0(t)$ oder $l_2(t)$ folgen, nach $l_1(t)$ kann $l_0(t)$,$l_1(t)$ oder $l_2(t)$ folgen und nach $l_2(t)$ bricht der α-β-Algorithmus ab.

Fall 3: Für $l_3(t+1)$ gilt im ersten Fall $L_3(t)$. Dann wird β reduziert. Da die Werte in absteigender Reihenfolge stehen, kann in den verbleibenden $k-1$ Fällen nur $L_0(t)$ folgen. Für $L_3(t)$ folgt für den ersten Wert $l_3(t)$ oder $l_4(t)$. Nach $l_4(t)$ kann wieder $l_3(t)$ oder $l_4(t)$ folgen, nach $l_3(t)$ nur noch $l_3(t)$. Im letzten Fall gilt jedoch immer $l_3(t)$ nach Voraussetzung, da $\alpha < x$ und da α nicht verändert wird.

Fall 4: Für $l_4(t+1)$ gilt im ersten Fall entweder $L_3(t)$ oder $L_4(t)$. Nach $L_3(t)$ kann entweder $L_0(t)$ oder $L_1(t)$ folgen, nach $L_4(t)$ bricht der α-β-Algorithmus ab. Für $L_4(t)$ gilt in allen k nachfolgenden Fällen $l_4(t)$.

Fall 5: Für $l_5(t+1)$ gilt in allen k Fällen $L_5(t)$. Für $l_5(t)$ kann im ersten Fall $l_5(t)$ oder $l_6(t)$ gelten, nach $l_6(t)$ kann $l_0(t)$ oder $l_2(t)$ gelten, nach $l_5(t)$ bricht der α-β-Algorithmus ab.

Fall 6: Für $l_6(t+1)$ gilt im ersten Fall entweder $L_6(t)$ oder $L_5(t)$. Nach $L_5(t)$ kann entweder $L_5(t)$ oder $L_6(t)$ folgen, nach $L_6(t)$ nur noch $L_6(t)$. Für den letzten der k Nachfolger gilt aber in jedem Fall $L_6(t)$. Für $L_6(t)$ gilt im ersten Fall $l_6(t)$. Für die $k-1$ verbleibenden Fälle kann nur noch $l_0(t)$ gelten.

Fall 7: Für $l_7(t+1)$ gilt im ersten Fall $L_7(t)$, für die $k-1$ nachfolgenden Fälle $L_6(7)$. Für $L_7(t)$ gilt im ersten Fall $l_7(t)$ und für die $k-1$ nachfolgenden Fälle $l_3(t)$.

3.2 Analyse des günstigsten Falles

Es gelten also folgende Beziehungen:

$$
\begin{aligned}
l_0(t+1) &= d \cdot L_0(t) + (k-d) \cdot L_1(t), \text{ für ein } d \text{ mit } 1 \leq d \leq k. \\
L_0(t) &= d \cdot l_0(t) + (k-d) \cdot l_1(t), \text{ für ein } d \text{ mit } 1 \leq d \leq k.
\end{aligned}
$$

$$
\begin{aligned}
l_1(t+1) &= a \cdot L_0(t) + L_1(t) + b \cdot L_2(t), \text{ für ein } a, \text{ und ein } b \text{ mit} \\
&\quad 0 \leq a \leq k-1, 0 \leq b \leq k-1, a+b+1 \leq k. \\
L_1(t) &= k \cdot l_1(t).
\end{aligned}
$$

$$
\begin{aligned}
l_2(t+1) &= k \cdot L_2(t). \\
L_2(t) &= a \cdot l_0(t) + b \cdot l_1(t) + l_2(t), \text{ für ein } a \text{ und ein } b \text{ mit} \\
&\quad 0 \leq a \leq k-1, 0 \leq b \leq k-1, a+b+1 \leq k.
\end{aligned}
$$

$$
\begin{aligned}
l_3(t+1) &= L_3(t) + (k-1) \cdot L_0(t). \\
L_3(t) &= d \cdot l_3(t) + (k-d) \cdot l_4(t), \text{ für ein } d \text{ mit } 1 \leq d \leq k.
\end{aligned}
$$

$$
\begin{array}{ll}
l_4(t+1) & = L_4(t) \text{ oder} \\
l_4(t+1) & = L_3(t) + d \cdot L_1(t) + (k-d-1) \cdot L_0(t), \\
 & \quad \text{für ein } d \text{ mit } 1 \leq d \leq k-1. \\
L_4(t) & = k \cdot l_4(t).
\end{array}
$$

$$
\begin{array}{ll}
l_5(t+1) & = k \cdot L_5(t). \\
L_5(t) & = l_5(t) \text{ oder } L_5(t) = l_6(t) + a \cdot l_0(t) + b \cdot l_2(t), \text{ für ein } a \text{ und} \\
 & \quad \text{ein } b \text{ mit } 0 \leq a \leq k-1, 0 \leq b \leq k-1, a+b \leq k-1.
\end{array}
$$

$$
\begin{array}{ll}
l_6(t+1) & = d \cdot L_6(t) + (k-d) \cdot L_5(t), \text{ für ein } d \text{ mit } 1 \leq d \leq k. \\
L_6(t) & = l_6(t) + (k-1) \cdot l_0(t).
\end{array}
$$

$$
\begin{array}{ll}
l_7(t+1) & = L_7(t) + (k-1) \cdot L_6(t). \\
L_7(t) & = l_7(t) + (k-1) \cdot l_3(t).
\end{array}
$$

Für alle $0 \leq i \leq 7$ gilt: Entweder $L_i(t)$ oder $l_i(t+1)$ besteht aus k Summanden. Da dies für alle gilt, folgt $l_i(t) \geq k^t$ und $L_i(t) \geq k^t$ für alle $0 \leq i \leq 7$.

$\Rightarrow l_1(t+1) = L_1(t)$, $l_2(t+1) = k \cdot L_2(t)$, $l_4(t+1) = L_4(t)$, $L_5(t) = l_5(t)$.

Damit sind die Fkt $l_1(t+1)$, $L_1(t)$, $l_2(t+1)$, $L_2(t)$, $l_4(t+1)$, $L_4(t)$, $l_5(t+1)$, $L_5(t)$ minimal und es gilt:

$(L_2(t) = L_5(t)) \leq (L_1(t) = L_4(t))$ und

$l_1(t+1) = l_2(t+1) = l_4(t+1) = l_5(t+1)$.

$$
\begin{array}{ll}
\Rightarrow l_0(t+1) & = L_0(t) + (k-1) \cdot L_2(t), \\
\quad L_0(t) & = l_0(t) + (k-1) \cdot l_1(t), \\
\quad L_3(t) & = l_3(t) + (k-1) \cdot l_4(t), \\
\quad l_6(t+1) & = L_6(t) + (k-1) \cdot L_5(t).
\end{array}
$$

Daraus entsteht folgende Tabelle:

Nr.	Min-Knoten $l_i(t+1)$	Max-Knoten $L_i(t)$
0	$L_0(t) + L_2(t) \cdot (k-1)$	$l_0(t) + l_1(t) \cdot (k-1)$
1	$L_1(t)$	$l_1(t) \cdot k$
2	$L_2(t) \cdot k$	$l_2(t)$
3	$L_3(t) + (k-1) \cdot L_0(t)$	$l_3(t) + (k-1) \cdot l_4(t)$
4	$L_4(t)$	$l_4(t) \cdot k$
5	$L_5(t) \cdot k$	$l_5(t)$
6	$L_6(t) + (k-1) \cdot L_5(t)$	$l_6(t) + (k-1) \cdot l_0(t)$
7	$L_7(t) + (k-1) \cdot L_6(t)$	$l_7(t) + (k-1) \cdot l_3(t)$

$\Rightarrow l_1(t) = l_2(t) = l_4(t) = l_5(t) = k^t$

Beweis: Es gilt $m_i(t+1) = k \cdot m_i(t) \quad \forall i \in \{1,2,4,5\}$
Für $t=0$ gilt $m_i(0) = k^0 = 1$.
Sei die Behauptung für t bewiesen:
$\Rightarrow m_i(t+1) = m_i(t) \cdot k = k^t \cdot k = k^{(t+1)}$
q.e.d.

$$\begin{aligned}
\Rightarrow l_0(t+1) &= L_0(t) + (k-1) \cdot L_2(t) \\
&= l_0(t) + (k-1) \cdot l_1(t) + (k-1) \cdot L_2(t) \\
&= l_0(t) + (k-1) \cdot l_1(t) + (k-1) \cdot l_2(t) \\
&= l_0(t) + (k-1) \cdot k^t + (k-1) \cdot k^t \\
&= l_0(t) + 2 \cdot k^{(t+1)} - 2 \cdot k^t \\
&= l_0(t) + (2 \cdot k - 2) \cdot k^t
\end{aligned}$$

$\Rightarrow l_0(t) = 2 \cdot k^t - 1$
Beweis durch Induktion nach t:
$l_0(0) = 2 \cdot k^0 - 1 = 2 \cdot 1 - 1 = 1$
Annahme, die Behauptung sei bereits für t bewiesen
$\Rightarrow l_0(t+1) = l_0(t) + 2 \cdot k \cdot k^t - 1 - 2 \cdot k^t + 1 = 2 \cdot k^t - 1 + (2 \cdot k - 2) \cdot k^t = 2 \cdot k^{(t+1)} - 1$
q.e.d.

$$\Rightarrow L_0(t) = 2 \cdot k^t - 1 + (k-1) \cdot k^t = k^{(t+1)} + k^t - 1$$

$$\begin{aligned}
l_3(t+1) &= l_3(t) + (k-1) \cdot l_4(t) + (k-1) \cdot L_0(t) \\
&= l_3(t) + k^{(t+1)} - k^t + (k-1) \cdot (k^{(t+1)} + k^t - 1) \\
&= l_3(t) + k^{(t+2)} + k^{(t+1)} - 2 \cdot k^t - k - 1 \\
&= l_3(t) + (k^2 + k - 2) \cdot k^t - k + 1
\end{aligned}$$

$$\Rightarrow l_3(t) = \tfrac{k^2+k-2}{k-1} \cdot k^t - (k-1) \cdot t - \tfrac{k^2+k-2}{k-1} + 1$$

Beweis durch Induktion nach t: $t = 0$

$$\begin{aligned}\Rightarrow l_3(0) &= \tfrac{k^2+k-2}{k-1} \cdot k^0 - (k-1) \cdot 0 - \tfrac{k^2+k-2}{k-1} + 1 \\ &= 1\end{aligned}$$

Annahme, die Behauptung sei bereits für t bewiesen:

$$\begin{aligned}\Rightarrow l_3(t+1) &= l_3(t) + (k^2+k-2) \cdot k^t - k + 1 \\ &= \tfrac{(k^2+k-2)+(k-1)\cdot(k^2+k-2)}{k-1} \cdot k^t - (k-1) - (k-1) \cdot t - \tfrac{k^2+k-2}{k-1} + 1 \\ &= \tfrac{k\ (k^2+k-2)}{k-1} \cdot k^t - (k-1) \cdot (t+1) - \tfrac{k^2+k-2}{k-1} + 1 \\ &= \tfrac{k^2+k-2}{k-1} \cdot k^{(t+1)} - (k-1) \cdot (t+1) - \tfrac{k^2+k-2}{k-1} + 1\end{aligned}$$

q.e.d.

$$\Rightarrow L_3(t) = k^{(t+1)} + \tfrac{k^2-3}{k-1} \cdot k^t - (k-1) \cdot t - \tfrac{k^2+k-2}{k-1} + 1$$

$$L_5(t) = l_5(t) = k^t$$

$$\begin{aligned}\Rightarrow l_6(t+1) &= l_6(t) + k^{(t+1)} - k^t + (k-1) \cdot l_0(t) \\ &= l_6(t) + 3 \cdot k^{(t+1)} - 3 \cdot k^t - k + 1\end{aligned}$$

$$\Rightarrow l_6(t) = 3 \cdot k^t - (k-1) \cdot t - 2$$

Beweis durch Induktion nach t: $t = 0$

$$l_6(0) = 3 \cdot k^0 - (k-1) \cdot 0 - 2 = 3 \cdot 1 - 2 = 1$$

Annahme, die Behauptung sei bereits für t bewiesen:

$$\begin{aligned}\Rightarrow l_6(t+1) &= l_6(t) + 3 \cdot k^{(t+1)} - 3 \cdot k^t - k + 1 \\ &= 3 \cdot k^t - (k-1) \cdot t - 2 + 3 \cdot k^{(t+1)} - 3 \cdot k^t - (k-1) \\ &= 3 \cdot k^{(t+1)} - (k-1) \cdot (t+1) - 2\end{aligned}$$

q.e.d.

$$\begin{aligned}\Rightarrow L_6(t) &= 3 \cdot k^t - (k-1) \cdot t - 2 + (k-1) \cdot (2 \cdot k^t - 1) \\ &= 2 \cdot k^{(t+1)} + k^t - (k-1) \cdot t - k - 1\end{aligned}$$

$$
\begin{aligned}
l_7(t+1) &= l_7(t) + (k-1) \cdot l_3(t) + (k-1) \cdot L_6(t) \\
&= l_7(t) + (k^2+k-2) \cdot k^t - (k-1)^2 \cdot t - (k^2+k-2) + k - 1 \\
&\quad +2 \cdot k^{(t+2)} - k^{(t+1)} - k^t - (k-1)^2 \cdot t - k^2 + 1 \\
&= l_7(t) + (3 \cdot k^2 - 3) \cdot k^t - 2 \cdot (k-1)^2 \cdot t - 2 \cdot k^2 + 2 \\
\Rightarrow l_7(t) &= (3 \cdot k + 3) \cdot k^t - (k-1)^2 \cdot t^2 - (k^2 + 2 \cdot k - 3) \cdot t - 3 \cdot k - 2
\end{aligned}
$$

Beweis durch Induktion nach t: $t = 0$

$$
\begin{aligned}
l_7(0) &= (3 \cdot k + 3) \cdot k^0 - (k-1)^2 \cdot 0^2 - (k^2 + 2 \cdot k - 3) \cdot 0 - 3 \cdot k - 2 \\
&= (3 \cdot k + 3) \cdot 1 - 3 \cdot k - 2 = 1
\end{aligned}
$$

Annahme, die Behauptung sei bereits für t bewiesen:

$$
\begin{aligned}
\Rightarrow l_7(t+1) &= l_7(t) + (3 \cdot k^2 - 3) \cdot k^t - 2 \cdot (k-1)^2 \cdot t - 2 \cdot k^2 + 2 \\
&= (3 \cdot k + 3) \cdot k^t - (k-1)^2 \cdot t^2 - (k^2 + 2 \cdot k - 3) \cdot t - 3 \cdot k - 2 \\
&\quad +(3 \cdot k^2 - 3) \cdot k^t - 2 \cdot (k-1)^2 \cdot t - 2 \cdot k^2 + 2 \\
&= (3 \cdot k^2 + 3 \cdot k - 3 + 3) \cdot k^t - (k-1)^2 \cdot t^2 \\
&\quad -(2 \cdot (k-1)^2 + k^2 + 2 \cdot k - 3) \cdot t - 2 \cdot k^2 - 3 \cdot k + 2 - 2 \\
&= (3 \cdot k^2 + 3 \cdot k) \cdot k^t - (k-1)^2 \cdot t^2 - (k-1)^2 \cdot 2 \cdot t - (k-1)^2 \\
&\quad +(k^2 + 2 \cdot k - 3) \cdot t - k^2 - 5 \cdot k + 1 \\
&= (3 \cdot k + 3) \cdot k^{(t+1)} - (k-1)^2 \cdot (t+1)^2 \\
&\quad -(k^2 + 2 \cdot k - 3) \cdot t - (k^2 + 2 \cdot k - 3) - 3 \cdot k - 2 \\
&= (3 \cdot k + 3) \cdot k^{(t+1)} - (k-1)^2 \cdot (t+1)^2 \\
&\quad -(k^2 + 2 \cdot k - 3) \cdot (t+1) - 3 \cdot k - 2 \\
\Rightarrow L_7(t) &= l_7(t) + (k-1) \cdot l_3(t) \\
&= (k^2 + 4 \cdot k + 1) \cdot k^t - (k-1)^2 \cdot t^2 - (2 \cdot k^2 - 2) \cdot t - k^2 - 3 \cdot k - 1
\end{aligned}
$$

Für gerade $n \geq 0$ folgt das Theorem aus $l_7(\frac{n}{2})$ und für ungerade $n > 0$ aus $L_7(\frac{n-1}{2})$.

4 Bemerkungen zu k–regulären Bäumen

4.1 Zur Qualität der Abschätzungen

Theorem 3: Sei $h(n)$ eine obere Schranke für die von dem α–β–Algorithmus bei einem optimalen Minimaxbaum mit gerader Tiefe n gelesene Anzahl Blattwerte. Sei $h(n)$ für gerade n von der Form $p(n) \cdot k^{\frac{n}{2}}$, wobei p(n) ein Polynom ist, und $g(n)$ die obere Schranke für gerade n aus Theorem 1.
Gilt $h(2 \cdot n) \leq g(2 \cdot n)$ für alle $n \geq 0$, so ist $h(2 \cdot n) = g(2 \cdot n)$ für alle $n \geq 0$.

Beweis:
Sei also $h(n) = p(n) \cdot k^{\frac{n}{2}}$ und $g(n)$ die obere Schranke aus Theorem 1.
Sei $f(n)$ die untere Schranke aus Theorem 2.
Sei nun $Grad(p) > 2$.
$\Rightarrow$ Es existiert ein t, so daß $\forall n > t$ gilt $p(n) \cdot k^{\frac{n}{2}} > g(n)$. Widerspruch, da $h(n) \leq g(n)$ für alle geraden $n \geq 0$.
$\Rightarrow Grad(p) \leq 2$.
Sei also $p(n) = a \cdot n^2 + b \cdot n + c$.
Da $g(0) = f(0), g(2) = f(2)$ und $g(4) = f(4)$ muß auch $h(n)$ für $n \in \{0, 2, 4\}$ mit $f(n)$ übereinstimmen. ($h(n)$ soll eine obere Schranke sein und $h(n) \leq g(n)$ für alle geraden $n \geq 0$)

$\Rightarrow p(0) \cdot k^0 = f(0) = 1 \Leftrightarrow (a \cdot 0^2 + b \cdot 0 + c) \cdot k^0 = 1 \Leftrightarrow c = 1$

$p(2) \cdot k^1 = f(2) = k^2 \Leftrightarrow (4 \cdot a + 2 \cdot b + 1) \cdot k = k^2 \quad \Leftrightarrow 4 \cdot a + 2 \cdot b = k - 1$

$p(4) \cdot k^2 = f(4) = 3 \cdot k^3 - 3 \cdot k^2 + k$
$\Leftrightarrow (16 \cdot a + 4 \cdot b + 1) \cdot k^2 = 3 \cdot k^3 - 3 \cdot k^2 + k$
$\Leftrightarrow (16 \cdot a + 4 \cdot b + 1) \cdot k = 3 \cdot k^2 - 3 \cdot k + 1$

$\Rightarrow 8 \cdot a \cdot k = k^2 - 2 \cdot k + 1 \quad \Leftrightarrow a = \frac{(k-1)^2}{8 \cdot k}$

$\Rightarrow b = (\frac{k-1}{2} - \frac{(k-1)^2}{4 \cdot k}) \Rightarrow p(n) \cdot k^{\frac{n}{2}} = g(n)$

$\Rightarrow g(n) = h(n)$ für alle geraden $n \geq 0$.

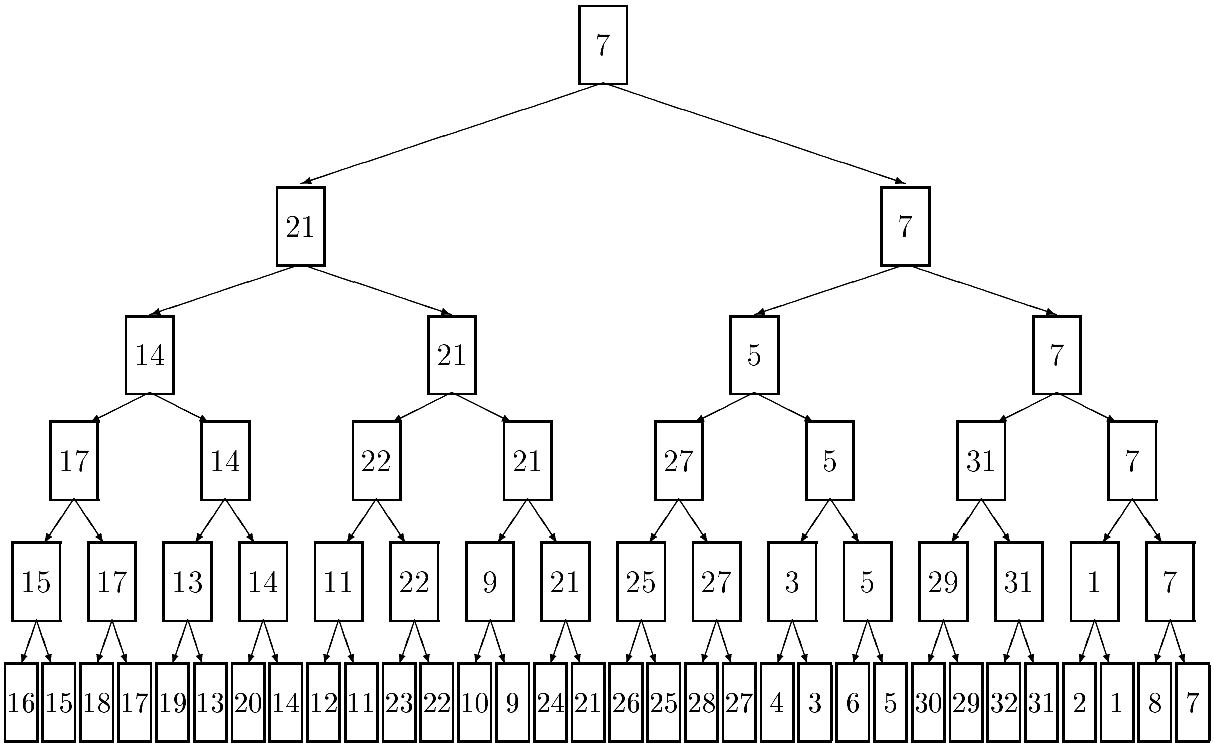
4.2 Zum Invertieren von Bäumen

Zu jedem Minimaxbaum mit Min–Wurzel, bei dem der α–β–Algorithmus m Blattwerte liest, läßt sich auch ein Minimaxbaum mit Max–Wurzel konstruieren, bei dem der α–β–Algorithmus ebenfalls m Blattwerte liest und umgekehrt.
Hat der Baum eine Min–Wurzel, so erhält man einen dazu gehörenden inversen Baum, indem man jeden Knotenwert w durch $-w$ ersetzt. Bei dem so entstandenen Baum sind aus den Min–Knoten Max–Knoten geworden und umgekehrt. Beim Durchlaufen des inversen Baumes sind die Rollen von α und β vertauscht. Die Anzahl gelesener Werte bleibt gleich.

Daher ist es, bei der Bestimmung von oberer und unterer Schranken, ausreichend, Minimaxbäume mit Min–Wurzeln zu untersuchen. Es werden deshalb in Kapitel 2 und in Kapitel 6 Konstruktionsverfahren geliefert, die Minimaxbäume mit Min–Wurzel erzeugen.
Auch innerhalb von Beweisen wird nicht mehr zwischen Bäumen mit Min–Wurzel und Max–Wurzel unterschieden. So wird zum Beispiel in Kapitel 6 Abschnitt 3 ein Induktionsbeweis geführt, in dem eine Aussage für einen Baum mit Min–Wurzel getroffen wird. Die Induktionsvorraussetzung wird dabei auf einen Baum mit Max–Wurzel angewendet.

5 Bestmöglicher Binärbaum der Tiefe 5

Im Rest dieser Arbeit werden nur noch Binärbäume betrachtet. Um nun zu zeigen, daß in der Konstruktion zu Theorem 1 noch nicht der bestmögliche Minimaxbaum konstruiert wird, betrachte man folgenden Minimaxbaum der Tiefe 5.



5.1 Analyse des Baumes

Bei dem angegebenen Baum liest der α–β–Algorithmus maximal 25 Blattwerte, unabhängig von der Sortierung des Baumes.
Beweis: Sei T der Baum und r die Wurzel des Baumes. Sei r ein Min–Knoten und A, B die beiden Teilbäume von T unterhalb von r. Seien weiter A_1, A_2 die Teilbäume von A und B_1, B_2 entsprechend die Teilbäume von B. Seien die Blattwerte in der worst–first–Darstellung des Baumes T geordnet.

Notation:

1. die Knotenbezeichnungen und ihre Werte werden im weiteren Verlauf miteinander identifiziert. Z.B. hat die Wurzel von T den Wert 7. Diese Wertzuweisung wird durch die Schreibweise $r = 7$ dargestellt.

2. $w(T)$ bezeichne den Wurzelwert des Teilbaumes T.

$\Rightarrow r = 7$, $w(A) = 21$, $w(A_1) = 14$, $w(A_2) = 21$, $w(B) = 7$, $w(B_1) = 5$, $w(B_2) = 7$.
Man betrachtet nun T in einer Tiefe von 2.

1. Annahme: B sei der zuerst gelesene Teilbaum von T.
   $\Rightarrow$ Unabhängig davon, wie A_1, A_2, B_1 und B_2 geordnet sind, wird ein Teilbaum (A_1 oder A_2) nicht mehr gelesen, denn in einer Tiefe von 5 sind $2^5 = 32$ Blattwerte vorhanden. Da die Wurzel ein Min-Knoten ist (nach Voraussetzung), gilt nach dem Durchlaufen von B: $\beta = 7$. Damit werden alle Blattwerte unterhalb eines Teilbaumes (A_1 oder A_2) nicht gelesen. Es gilt daher: bei einer Tiefe von 5 werden mindestens 8 Blattwerte nicht gelesen, d.h. es werden maximal $32 - 8 = 24 < 25$ Blattwerte gelesen.

2. Sei also nun A der zuerst gelesene Teilbaum von T.

   $\Rightarrow$ A ist ein Baum der Tiefe 4, der die Bedingung von Knuth und Moore erfüllt; d.h. von den 16 Blattwerten von A werden maximal 14 Blattwerte gelesen, unabhängig von der Anordnung der Blattwerte von A. Es gilt für r ($= w(T)$) nach dem Durchlaufen von A: $r \leq 21$.

   (a) Sei B_1 der zuerst gelesene Teilbaum von B.
   $\Rightarrow$ unabhängig davon, wie die Blattwerte von B_1 geordnet sind, werden von den Werten 26, 25, 28, 27 zwei nicht gelesen. Damit werden von 8 Blattwerten von B_1 nur 6 gelesen. Nun gilt für die Wurzel von B_2: $5 \leq w(B_2) \leq 21$.
   $\Rightarrow$ unabhängig davon, wie die Blattwerte von B_2 geordnet sind, werden von den Werten 30, 29, 32 und 31 zwei nicht gelesen. Ebenso wird von den Werten 2 und 1 einer nicht gelesen (da $w(B_2) \geq 5$, unabhängig von der Anordnung der Blattwerte von B_2). D.h. es werden von B_2: $8 - 2 - 1 = 5$ Blattwerte gelesen und damit von T: $14 + 6 + 5 = 25$.

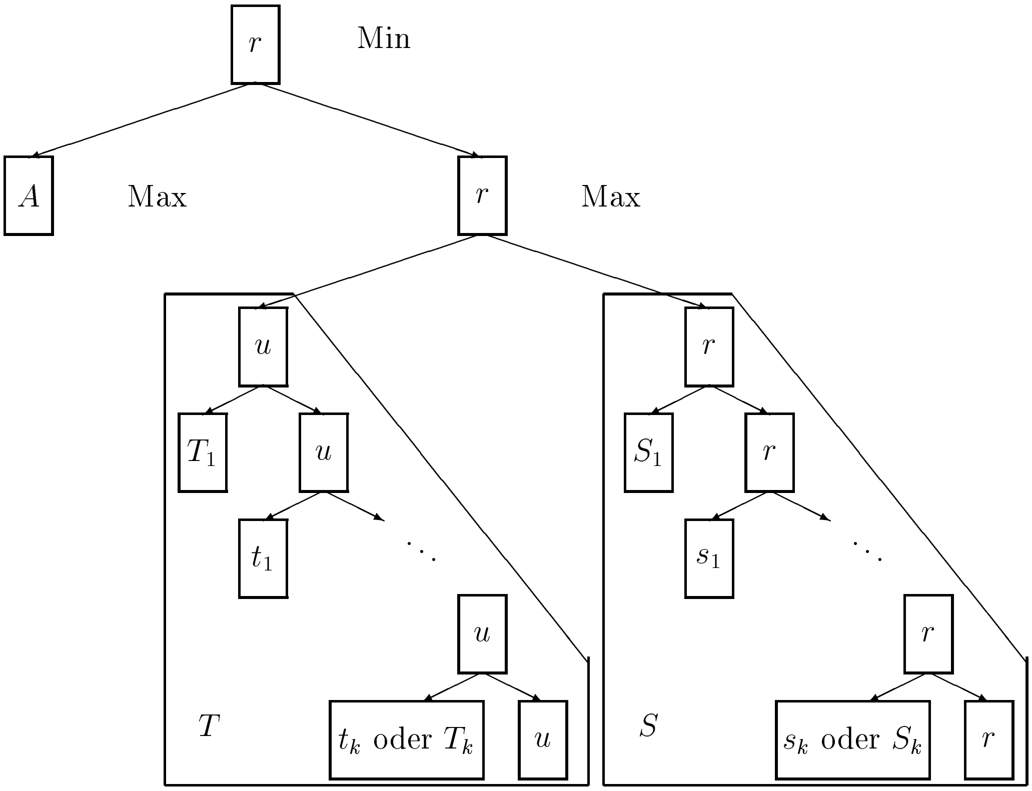
(b) Sei B_2 der zuerst gelesene Teilbaum von B.
$\Rightarrow$ Analog zu 2.(a) werden von den Blattwerten von B_2 nur 6 gelesen (von den Werten 30, 29 32, 31 werden zwei nicht gelesen). Nun gilt: $7 \leq w(B_1) \leq 21$. Analog zu 2.(a) werden dann von den Blattwerten 26, 25, 28 und 27 von B_1 zwei nicht gelesen. Da nun gilt $\alpha \geq 7$, wird von den Werten 4 und 3 nur einer gelesen; ebenso von den Werten 6 und 5.
D.h. von B_1 werden $8-2-1-1=4$ Blattwerte gelesen und damit von T: $14+6+4=24<25$.
$\Rightarrow$ Der α–β–Algorithmus liest beim Durchlaufen von T maximal 25 Blattwerte unabhängig von der Sortierung.

5.2 Bemerkung

Ersetzt man bei dem Baum T dieses Kapitels den Wert 4 durch 33 und den Wert 6 durch 34, so liest der α–β–Algorithmus beim Durchlaufen des Baumes in der worst-first Darstellung zwar 25 Blattwerte, aber werden die Blattpositionen 17 bis 25 mit den Postitionen 26 bis 32 vertauscht (d.h. eine Umordnung des Baumes unterhalb des rechten Max-Knotens unterhalb der Wurzel), so liest der α–β–Algorithmus 26 Blattwerte. Dadurch ist gezeigt, daß die worst-first Darstellung eines Baumes nicht immer die ungünstigste Sortierung darstellen muß.
Durch den Baum T ist gezeigt worden, daß für die Tiefen $1 \leq n \leq 5$ ein Baum existiert, dessen maximale Anzahl gelesener Blattwerte mit der unteren Schranke aus Theorem 2 übereinstimmt. Damit ist gezeigt, daß die Anzahl abzufragender Blattwerte für $n>4$ echt kleiner ist, als die durch Theorem 1 gegebene obere Schranke angibt.

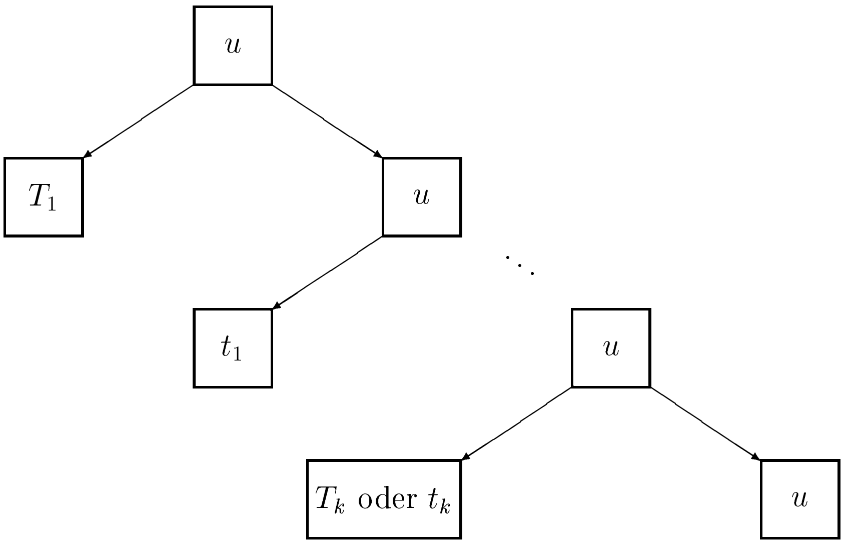
6 Konstruktionsverfahren für Binärbäume

Sei M ein binärer Minimaxbaum der Tiefe n, der wie folgt aufgebaut ist.



Seien T und S die entsprechenden Teilbäume von M.
Notationen: Sei $B(T)$ die Menge aller Blattwerte von T.
Man schreibe $B(T) < x$, wenn gilt $y < x \quad \forall \quad y \in B(T)$ und
$B(S) < B(T)$, wenn gilt $x < y \quad \forall \quad x \in B(S)$ und $y \in B(T)$.

Sei T von der Form:



Es soll weiter gelten:

$B(s_k) < B(s_{k-1}) < \ldots < B(s_1) < B(t_k) < \ldots < B(t_1) < u < r < B(A) < B(T_1) < B(T_2) < \ldots < B(T_j) < B(S_1) < \ldots < B(S_j)$
mit $j = k$, falls n gerade ist, und $j = k - 1$, falls n ungerade ist.

Die t_i und s_i seien, mit den entsprechend zugeordneten Blattwerten, wie der gesamte Baum M geordnet; die T_i und S_i seien, mit den entsprechend zugeordneten Blattwerten, wie der zu dem gesamten Baum M gehörende inverse Baum geordnet.
Sei n_T die Tiefe und u die Wurzel von T.

Sei n die Tiefe von M.
Durch die obigen Vorgaben soll nun eine obere Schranke von der Größenordnung $43 \cdot 2^{\frac{n}{2}}$, falls n gerade ist und $61 \cdot 2^{\frac{n-1}{2}}$, falls n ungerade ist, bewiesen werden. Dazu werden zunächst drei Hilfsbetrachtungen angestellt.

Für den α–β–Algorithmus sind nicht die Knotenwerte selbst, sondern nur ihre Größenordnungen zueinander, von Bedeutung. Daher ist es nicht notwendig, Werte für Alpha und Beta vorzugeben, sondern es ist ausreichend, ihre Relation zu den Blattwerten anzugeben. Eine solche Vorgabe ist z.B $B(t_i) < \alpha \leq u \leq \beta < B(T_j) \quad \forall \quad i, j$.

In Abschnitt 6.1 wird zunächst der Teilbaum T mit einer besonders günstigen Vorgabe von Alpha und Beta betrachtet. Günstig bedeutet, daß der α–β–Algorithmus bei dieser Vorgabe von Alpha und Beta relativ wenig Blattwerte von T liest. Die genaue Anzahl wird in Lemma 1 angeführt. Der Beweis dieses Lemmas wird zerlegt in die Teilbereiche:

1. Analyse der T_j,

2. Analyse der t_i.

Die Zusammenfassung der Ergebnisse wird in einer Funktion $r(n)$ festgehalten.

Hat man einen optimalen Baum und wird die in einer Tiefe n gelesene Anzahl Blattwerte des ungünstigsten Falles durch eine Funktion $m(n)$ angegeben, so stellt sich die Frage nach dem Größenzuwachs von m von zwei unterschiedlichen Tiefen. In Abschnitt 6.2 wird dieser Größenzuwachs, für den in diesem Kapitel konstruierten Baum, mit dem Größenzuwachs des günstigsten Falles in Beziehung gesetzt.

Mit diesen beiden Hilfsbetrachtungen kann nun in Abschnitt 6.3 der Teilbaum T mit der ungünstigsten Vorgabe von Alpha und Beta analysiert werden. Da der Teilbaum S vom Aufbau her mit dem Teilbaum T übereinstimmt, gelten die Ergebnisse der Analysen von T auch für S.

Somit kann nun in Abschnitt 6.4 der gesamte Baum M analysiert werden. Auch hier werden zwei Bereiche unterschieden, nämlich:

1. Der Teilbaum A wird von dem α–β–Algorithmus zuerst durchlaufen,
2. Der rechts von der Wurzel von M befindliche Teilbaum wird zuerst durchlaufen.

Die Ergebnisse dieser Betrachtungen sind zwei unterschiedliche Rekursionsvorschriften.

Es wird in Abschnitt 6.5 obige erwähnte obere Schranke bewiesen, indem sie für beide Rekursionsvorschriften bewiesen wird.

In Abschnitt 6.6 wird eine spezielle Realisierung des Baumes M mit festen Blattwerten vorgenommen, um einen binären Minimaxbaum angeben zu können, der diese obere Schranke hat.
Dieser Abschnitt dient weiter, um das im Anhang angegebene Programm leichter verstehen zu können. Dieses Programm liefert die Blattwertbelegung dieser speziellen Realisierung von M in der worst–first–Darstellung des Baumes. Mit dieser Blattwertbelegung können nun, wie bereits in der Einleitung erwähnt, parallele Algorithmen getestet werden.

6.1 Hilfsbetrachtung 1

Lemma 1: Sei $B(t_i) < \alpha \leq u \leq \beta < B(T_j) \quad \forall \quad i, j$
$\Rightarrow$ Der α–β–Algorithmus liest $2^{\frac{n_T}{2}+1} - 1$ Blattwerte von T, falls n_T gerade ist und $2^{\frac{n_T+1}{2}} + 2^{\frac{n_T-1}{2}} - 1$ Blattwerte, falls n_T ungerade ist.

Beweis: Die Wurzeln aller T_j sind Max–Knoten und die Wurzeln aller t_i sind Min–Knoten.

1. Aus $\beta < B(T_j)$ folgt, von den Nachfolgern eines jeden Max–Knotens aus den T_j wird jeweils nur einer gelesen. Sei n_{T_j} die Tiefe von T_j.
   $\Rightarrow$ der α–β–Algorithmus liest $2^{\frac{n_{T_j}}{2}}$ Blattwerte von T_j, falls n_{T_j} gerade ist und $2^{\frac{n_{T_j}-1}{2}}$ Blattwerte, falls n_{T_j} ungerade ist.
   Beweis durch Induktion nach n_{T_j}:
   Für $n_{T_j} = 0$ existiert nur ein Knoten, die Wurzel. $2^{\frac{0}{2}} = 2^0 = 1$.
   Für $n_{T_j} = 1$ gilt, da die Wurzel von T_j ein Max-Knoten ist und β kleiner ist als die Nachfolger dieses Knotens, wird von den Nachfolgern nur einer gelesen. $2^{\frac{1-1}{2}} = 2^0 = 1$.
   Annahme, die Behauptung sei für eine Tiefe von $n_{T_j} - 1$ bewiesen.
   $\Rightarrow$ In einer Tiefe von $n_{T_j} - 1$ werden $2^{\frac{n_{T_j}-1}{2}}$ Blattwerte gelesen, falls n_{T_j} ungerade ist und $2^{\frac{(n_{T_j}-1)-1}{2}}$ Blattwerte, falls n_{T_j} gerade ist. Ist n_{T_j} ungerade, so sind die Knoten der Tiefe $n_{T_j} - 1$ Max–Knoten und die Anzahl der gelesenen Knoten stimmt mit der Anzahl gelesener Knoten der Tiefe n_{T_j} überein. Ist n_{T_j} gerade, so sind die Knoten der Tiefe $n_{T_j} - 1$ Min–Knoten, d.h. die Anzahl gelesener Knoten der Tiefe n_{T_j} ist doppelt so groß wie die Anzahl gelesener Knoten der Tiefe $n_{T_j} - 1$.
   Daraus folgt die Behauptung für den ersten Fall.

2. Sei n_{t_i} die Tiefe von t_i. Aus $B(t_i) < \alpha$ folgt nun, von den Nachfolgern eines jeden Min–Knotens aller t_i wird jeweils nur einer gelesen. Von den Nachfolgern eines jeden Max–Knotens aller t_i werden alle gelesen. Analog zu 1. folgt nun, daß der α–β–Algorithmus $2^{\frac{n_{t_i}}{2}}$ Blattwerte liest, falls n_{t_i} gerade ist und $2^{\frac{n_{t_i}-1}{2}}$ Blattwerte, falls n_{t_i} ungerade ist.

Gilt nun $B(t_i) < \alpha \leq u \leq \beta < B(T_j) \quad \forall \quad i,j$, so endet T mit einem t_k, $k = \frac{n_T}{2}$, falls n_T gerade ist und mit einem T_s, $s = \frac{n_T+1}{2}$, falls n_T ungerade ist. Sei $r(n_T)$ die Anzahl gelesener Blattwerte von T in einer Tiefe von n_T. Dann gilt:
a) n_T ist gerade.

$$\begin{aligned}
\Rightarrow r(n_T) &= \sum_{\substack{i=0\\ i\ gerade}}^{n_T-2} 2^{\frac{i}{2}} \quad + \sum_{\substack{j=1\\ j\ ungerade}}^{n_T-1} 2^{\frac{j-1}{2}} \quad +1 \\
&= \sum_{i=0}^{\frac{n_T}{2}-1} 2^i \quad + \sum_{j=0}^{\frac{n_T}{2}-1} 2^j \quad +1 \\
&= (2^{\frac{n_T}{2}}-1)+(2^{\frac{n_T}{2}}-1)+1 \\
&= 2^{\frac{n_T}{2}+1}-1
\end{aligned}$$

b) n_T ist ungerade.

$$\begin{aligned}
\Rightarrow r(n_T) &= \sum_{\substack{i=0\\ i\ gerade}}^{n_T-1} 2^{\frac{i}{2}} \quad + \sum_{\substack{j=1\\ j\ ungerade}}^{n_T-2} 2^{\frac{j-1}{2}} \quad +1 \\
&= \sum_{i=0}^{\frac{n_T-1}{2}} 2^i \quad + \sum_{j=0}^{\frac{n_T-3}{2}} 2^j \quad +1 \\
&= (2^{\frac{n_T+1}{2}}-1)+(2^{\frac{n_T-1}{2}}-1)+1 \\
&= 2^{\frac{n_T+1}{2}}+2^{\frac{n_T-1}{2}}-1
\end{aligned}$$

Damit ist Lemma 1 bewiesen.
Bemerkung: $r(n)$ stellt die von Knuth und Moore gefundene untere Schranke dar (siehe Einleitung).

6.2 Hilfsbetrachtung 2

Lemma 2: Sei $r(n)$ die von Knuth und Moore angegebene untere Schranke

für die Anzahl gelesener Blattwerte bei einem Minimaxbaum der Tiefe n.
Sei $m(n)$ die Anzahl gelesener Blattwerte des Minimaxbaumes M bei ungünstigster Sortierung.
Sei $f(n) = m(n) - r(n)$.
Dann gilt: f ist eine monoton wachsende Funktion.

Beweis: Es gilt $f(n) \geq 0 \quad \forall n \geq 0$.
In einer Tiefe von n werden im linken Teilbaum von M im ungünstigsten Fall $m(n-1)$ viele Blattwerte gelesen. Dies gilt, weil der linke Teilbaum rekursiv aus M selbst hervorgeht.
Für den rechten Teilbaum von M gilt dann im günstigsten Fall:
Es werden $\frac{1}{2} \cdot 2^{\frac{n}{2}}$ viele Blattwerte gelesen, falls n gerade ist und es werden $2^{\frac{n-1}{2}}$ viele Blattwerte gelesen, falls n ungerade ist.

Demnach gilt für $n > 0$:

$m(n) \geq m(n-1) + \frac{1}{2} \cdot 2^{\frac{n}{2}}$, falls n gerade ist und

$m(n) \geq m(n-1) + 2^{\frac{n-1}{2}}$, falls n ungerade ist.

Es gilt weiter:

$r(n) - r(n-1) = \frac{1}{2} \cdot 2^{\frac{n}{2}}$, falls n gerade ist und

$r(n) - r(n-1) = 2^{\frac{n-1}{2}}$, falls n ungerade ist.

$\Rightarrow m(n) \geq m(n-1) + r(n) - r(n-1) \quad \forall n > 0$

$\Leftrightarrow m(n) - r(n) \geq m(n-1) - r(n-1) \quad \forall n > 0$

$\Leftrightarrow f(n) \geq f(n-1) \quad \forall n > 0$

$\Leftrightarrow f$ ist monoton wachsend.
q.e.d.

6.3 Hilfsbetrachtung 3

Lemma 3: Seien α und β nun beliebig vorgegeben. Sei nun jeder der Teilbäume T_i und t_j so angeordnet, daß, wenn ihre Blattwerte von links nach rechts gelesen werden, der α–β–Algorithmus die meisten Blattwerte liest. Dann gilt: Der α–β–Algorithmus liest beim Durchlaufen von T die meisten Blattwerte, wenn die Teilbäume von links nach rechts gelesen werden, d.h. in der Reihenfolge T_1, t_1, T_2, t_2, …

Beweis: Sei $m(n)$ die größte Anzahl Abfragen, die beim Durchlaufen des α–β–Algorithmus eines binären Minimaxbaumes der Tiefe n auftritt. Die Wurzel von T ist ein Min–Knoten. Ist nun n_T ungerade, so sind die Blätter von T Max–Knoten. Der links von dem Blattwert u befindliche Teilbaum ist also ein Baum T_k mit der Tiefe $n_{T_k} = 1$ mit $k = \frac{n_T+1}{2}$. Ist nun n_T gerade, so sind die Blätter von T Min–Knoten. Der links von dem Blattwert u befindliche Teilbaum ist also ein Baum t_k mit der Tiefe $n_{t_k} = 1$ mit $k = \frac{n}{2}$.
Beweis obiger Behauptung durch Induktion nach n_T.
a) Für $n_T = 0$ ist die Behauptung richtig, da T nur aus einem Knoten besteht.
b) Für $n_T = 1$ ist die Behauptung richtig, da T nur zwei Blattwerte hat und beide gelesen werden unabhängig von der Sortierung von T.
Sei die Behauptung nun für eine Tiefe von $n_T - 1$ bewiesen.

1. T_1 wird zuerst durchlaufen.

   T_1 hat eine Tiefe von $n_T - 1$. Da α und β zu Beginn beliebig gewählt sein sollten, sei der ungünstigste Fall angenommen. D.h. von T_1 werden $m(n_T - 1)$ Blattwerte gelesen. Dann wird β entsprechend reduziert.

   (Dieser Fall kann angenommen werden, denn wenn bereits an dieser Stelle durch α oder β Ausgrenzungen vorgenommen werden können, so kann dies unabhängig von der Sortierung von T getan werden.) Der verbleibende Teil von T ist ein Teilbaum der Tiefe $n_T - 1$, der wie T aufgebaut ist. (In dem verbleibenden Teilbaum ist die Rolle der Min– und Max–Knoten im Vergleich zu T vertauscht. Da aber auch dementsprechend die Rollen der α– und β–Schranken vertauscht werden, gilt für diesen verbleibenden Teilbaum die Induktionsvoraussetzung.) Nach Induktionsvoraussetzung wird dieser verbleibende Teilbaum von T am ungünstigsten durchlaufen, wenn die Teilbäume von links nach rechts gelesen werden. Da gilt $B(T_1) > B(t_1)$ hat die β–Schranke keinen Einfluß auf t_1. Es werden also $m(n_T - 2)$ Blattwerte von t_1 gelesen. Nun wird α entsprechend erhöht. Für den restlichen Teilbaum von T gilt nun $B(t_i) < \alpha \leq u \leq \beta < B(T_j) \quad \forall \quad i,j$. Nach Lemma 1 folgt nun, daß die Anzahl gelesener Blätter des restlichen Baumes gleich $r(n_T - 2)$ ist. Es werden also $m(n_T - 1) + m(n_T - 2) + r(n_T - 2)$ Blattwerte von T gelesen.

2. Sei nun der rechte Teilbaum von T der zuerst gelesene Teilbaum. Dieser ist ein Baum von der Tiefe $n_T - 1$, für den die Induktionsvoraussetzung gilt. Da α noch keinen Einfluß auf die Blattwerte von t_1 hat, werden $m(n_T - 2)$ Blattwerte von t_1 gelesen. Anschließend wird α erhöht. Da β noch keinen Einfluß auf T_2 hat, werden nun $m(n_T - 3)$ Blattwerte von T_2 gelesen. Für den noch verbleibenden Teil gilt
$B(t_i) < \alpha \leq u \leq \beta < B(T_j) \quad \forall \quad i,j$.
Nach Lemma 1 folgt nun, daß die Anzahl gelesener Blätter des restlichen Baumes gleich $r(n_T - 3)$ ist. Es gilt nun $\beta = u$. Da aber $B(T_1) > u$, werden von T_1 $2^{\frac{n_T-1}{2}}$ Blattwerte gelesen, falls n_T ungerade ist und $2^{\frac{n_T-2}{2}}$ Blattwerte, falls n_T gerade ist. Somit werden von T: $m(n_T - 2) + m(n_T - 3) + r(n_T - 3) + 2^{\frac{n_T-1}{2}}$ Blattwerte gelesen, falls n_T ungerade ist und $m(n_T - 2) + m(n_T - 3) + r(n_T - 3) + 2^{\frac{n_T-2}{2}}$ Blattwerte, falls n_T gerade ist.

Es bleibt also noch zu zeigen:

$m(n_T - 1) + m(n_T - 2) + r(n_T - 2) \geq m(n_T - 2) + m(n_T - 3) + r(n_T - 3) + 2^{\frac{n_T-1}{2}}$, falls n_T ungerade ist und

$m(n_T - 1) + m(n_T - 2) + r(n_T - 2) \geq m(n_T - 2) + m(n_T - 3) + r(n_T - 3) + 2^{\frac{n_T-2}{2}}$, falls n_T gerade ist.

Sei n_T ungerade
$\Rightarrow m(n_T - 1) + 2^{\frac{n_T-1}{2}} + 2^{\frac{n_T-1}{2}-1} - 1 \geq m(n_T - 3) + 2^{\frac{n_T-1}{2}} + 2^{\frac{n_T-1}{2}} - 1$

$\Leftrightarrow m(n_T - 1) \geq m(n_T - 3) + \frac{1}{2} \cdot 2^{\frac{n_T-1}{2}}$
Sei n_T gerade

$\Rightarrow m(n_T - 1) + 2^{\frac{n_T}{2}} - 1 \geq m(n_T - 3) + 2^{\frac{n_T}{2}-1} + 2^{\frac{n_T-1}{2}-1} - 1 + 2^{\frac{n_T-1}{2}-2}$

$\Leftrightarrow m(n_T - 1) \geq m(n_T - 3) + \frac{1}{4} \cdot 2^{\frac{n_T}{2}}$

Die von Knuth und Moore gezeigte Abschätzung des günstigsten Falles ent-

spricht $r(n)$ (siehe Einleitung).
Nach Lemma 2 gilt $m(n-1) - r(n-1) \geq m(n-3) - r(n-3)$

$\Leftrightarrow m(n_T - 1) \geq m(n_T - 3) + r(n_T - 1) - r(n_T - 3)$
Sei n_T ungerade
$\Rightarrow m(n_T - 1) \geq m(n_T - 3) + 2^{\frac{n_T-1}{2}+1} - 1 - 2^{\frac{n_T-3}{2}+1} + 1$

$\Rightarrow m(n_T - 1) \geq m(n_T - 3) + \frac{3}{2} \cdot 2^{\frac{n_T-1}{2}}$
Sei n_T gerade
$\Rightarrow m(n_T - 1) \geq m(n_T - 3) + 2^{\frac{n_T}{2}} + 2^{\frac{n_T}{2}-1} - 1 - 2^{\frac{n_T}{2}-2} - 2^{\frac{n_T}{2}-2} + 1$

$\Rightarrow m(n_T - 1) \geq m(n_T - 3) + \frac{3}{4} \cdot 2^{\frac{n_T}{2}}$

Damit ist die Behauptung bewiesen.

6.4 Analyse des Minimaxbaumes

1. A wird zuerst durchlaufen.

   $\Rightarrow m(n-1)$ Abfragen von A werden im ungünstigsten Fall gelesen, da A ein binärer Minimaxbaum der Tiefe $n-1$ ist. Da die Wurzel von M ein Min–Knoten sein soll, wird nun β entsprechend auf den Wurzelwert von A reduziert.
   Die Auswirkungen von u auf S sind äquivalent zu den Auswirkungen von r auf T. Daher werde O.B.d.A. zuerst T und dann S gelesen. Nach Lemma 3 wird T (und ebenso S) am ungünstigsten durchlaufen, wenn die Blattwerte von links nach rechts gelesen werden. Demnach ist also T_1 der zuerst gelesene Teilbaum von T. Da gilt $B(A) < B(T_1)$ werden von den Blattwerten von T_1 $2^{\frac{n-3}{2}}$ Blattwerte gelesen, falls n ungerade ist und $2^{\frac{n-4}{2}}$ Blattwerte, falls n gerade ist (T und S sind Bäume der Tiefe $n-3$). Nun gilt für t_1: $\alpha < B(t_1) < \beta$, d.h. es werden von den Blattwerten von t_1 im ungünstigsten Fall $m(n-4)$ Blattwerte gelesen (t_1 ist ein binärer Minimaxbaum der Tiefe $n-4$). Nun wird α auf den Wurzelwert von t_1 erhöht. Es gilt also für den verbleibenden Rest von T $B(t_i) < \alpha \leq u \leq \beta < B(T_j) \quad \forall i > 1$ und $j > 1$. D.h. von dem verbleibenden Rest von T werden $r(n-4)$ (nach Hilfsbetrachtung 1) Blattwerte gelesen. Für S gilt nun $B(s_i) < \alpha \leq r \leq \beta < B(S_j)$.
   D.h. nach Hilfsbetrachtung 1 werden $r(n-2)$ Blattwerte von S gelesen. Demnach gilt für M

$m(n) \leq m(n-1) + m(n-4) + r(n-2) + r(n-4) + 2^{\frac{n-3}{2}}$, falls n ungerade ist und

$m(n) \leq m(n-1) + m(n-4) + r(n-2) + r(n-4) + 2^{\frac{n-4}{2}}$, falls n gerade ist.

2. Sei der rechte Teilbaum von M der zuerst gelesene. Da gilt, daß die Auswirkungen von u auf S äquivalent sind zu den Auswirkungen von r auf T, sei wieder O.B.d.A. T der zuerst gelesene Teilbaum. Nach Lemma 3 wird T von links nach rechts am ungünstigsten durchlaufen. Also wird zunächst T_1 gelesen, dann t_1, usw. T_1 ist ein binärer Minimaxbaum der Tiefe $n-3$, daher werden von ihm im ungünstigsten Fall $m(n-3)$ Blattwerte gelesen. Dann wird die β–Schranke auf den Wurzelwert von T_1 reduziert. Diese neue β–Schranke hat jedoch keine Auswirkungen auf t_1, so daß von den Blattwerten von t_1 $m(n-4)$ im ungünstigsten Fall gelesen werden. Für den verbleibenden Teil von T gilt $B(t_i) < \alpha \leq u \leq \beta < B(T_j) \quad \forall i > 1$ und $j > 1$. D.h. von dem verbleibenden Rest von T werden $r(n-4)$ (nach Hilfsbetrachtung 1) Blattwerte gelesen. Da die Wurzel von T ein Min–Knoten ist, wird die α–Schranke auf den Wurzelwert von T erhöht, bevor S durchlaufen wird. Entsprechend gilt nun für S nach Lemma 3, daß S am ungünstigsten von links nach rechts durchlaufen wird. Da der Wurzelwert von T aber keine Auswirkungen auf S_1 hat, werden von den Blattwerten von S_1 im ungünstigsten Fall $m(n-3)$ viele gelesen. Für den verbleibenden Teil von S gilt: $B(s_i) < \alpha \leq r \leq \beta < B(S_j)$. D.h. es werden $r(n-3)$ Blattwerte von dem verbleibenden Teil von S gelesen. Nun wird β auf r reduziert. Da gilt $r < B(A)$, werden von den Blattwerten von A $2^{\frac{n-1}{2}}$ viele gelesen, falls n ungerade ist und $2^{\frac{n-2}{2}}$ viele, falls n gerade ist. Demnach gilt für M

   $m(n) \leq 2 \cdot m(n-3) + m(n-4) + r(n-3) + r(n-4) + 2^{\frac{n-1}{2}}$, falls n ungerade ist und

   $m(n) \leq 2 \cdot m(n-3) + m(n-4) + r(n-3) + r(n-4) + 2^{\frac{n-2}{2}}$, falls n gerade ist.

6.5 Beweis einer oberen Schranke

Theorem 4: Zu jedem $n \geq 0$ existiert ein binärer Minimaxbaum M der Tiefe n mit distinkten Blattwerten, in welchem die Anzahl der von dem α–β–Algorithmus gelesenen Blattwerte echt kleiner ist als $43 \cdot 2^{\frac{n}{2}}$, falls n gerade ist und echt kleiner als $61 \cdot 2^{\frac{n-1}{2}}$, falls n ungerade ist.
Beweis:
Seien M, $m(n)$ und $r(n)$ wie zuvor definiert. Die Behauptung soll nun für jede Abschätzung per Induktion nach n bewiesen werden. Es gilt $m(0) = 1$, $m(1) = 2$, $m(2) = 4$, $m(3) = 8$ und $m(4) = 14$. Also ist der Induktionsanfang bewiesen.
Induktionsschritt:

1. Sei A der zuerst gelesene Teilbaum von M.

   (a) Sei n gerade. Dann gilt

   $$\begin{aligned} m(n) &\leq m(n-1) + m(n-4) + r(n-2) + r(n-4) + 2^{\frac{n-4}{2}} \\ &= m(n-1) + m(n-4) + 2^{\frac{n}{2}} - 1 + \tfrac{1}{2} \cdot 2^{\frac{n}{2}} - 1 + 2^{\frac{n-4}{2}} \\ &= m(n-1) + m(n-4) + \tfrac{7}{4} \cdot 2^{\frac{n}{2}} - 2 \end{aligned}$$

   Annahme, die Behauptung sei bereits für $n-1$, $n-2$, $n-3$ und $n-4$ bewiesen, dann gilt

   $$\begin{aligned} m(n) &\leq m(n-1) + m(n-4) + \tfrac{7}{4} \cdot 2^{\frac{n}{2}} - 2 \\ &< 61 \cdot 2^{\frac{(n-1)-1}{2}} + 43 \cdot 2^{\frac{n}{2}-2} + \tfrac{7}{4} \cdot 2^{\frac{n}{2}} - 2 \\ &= \tfrac{61}{2} \cdot 2^{\frac{n}{2}} + \tfrac{43}{4} \cdot 2^{\frac{n}{2}} + \tfrac{7}{4} \cdot 2^{\frac{n}{2}} - 2 \\ &= 43 \cdot 2^{\frac{n}{2}} - 2 \\ &< 43 \cdot 2^{\frac{n}{2}} \end{aligned}$$

   (b) Sei n ungerade. Dann gilt

   $$\begin{aligned} m(n) &\leq m(n-1) + m(n-4) + r(n-2) + r(n-4) + 2^{\frac{n-3}{2}} \\ &= m(n-1) + m(n-4) + 2^{\frac{n-1}{2}} + \tfrac{1}{2} \cdot 2^{\frac{n-1}{2}} - 1 + 2^{\frac{n-3}{2}} \\ &\quad + \tfrac{1}{2} \cdot 2^{\frac{n-1}{2}} + \tfrac{1}{4} \cdot 2^{\frac{n-1}{2}} - 1 \\ &= m(n-1) + m(n-4) + \tfrac{11}{4} \cdot 2^{\frac{n-1}{2}} - 2 \end{aligned}$$

   Annahme, die Behauptung sei bereits für $n-1$, $n-2$, $n-3$ und $n-4$ bewiesen, dann gilt

$$\begin{aligned}
m(n) &\leq m(n-1) + m(n-4) + \tfrac{11}{4} \cdot 2^{\frac{n-1}{2}} - 2 \\
&< 43 \cdot 2^{\frac{n-1}{2}} + 61 \cdot 2^{\frac{n-1}{2}-2} + \tfrac{11}{4} \cdot 2^{\frac{n-1}{2}} - 2 \\
&= 43 \cdot 2^{\frac{n-1}{2}} + \tfrac{61}{4} \cdot 2^{\frac{n-1}{2}} + \tfrac{11}{\cdot} 2^{\frac{n-1}{2}} - 2 \\
&= 61 \cdot 2^{\frac{n-1}{2}} - 2 \\
&< 61 \cdot 2^{\frac{n-1}{2}}
\end{aligned}$$

2. Sei der rechte Teilbaum von M der zuerst gelesene.

   (a) Sei n gerade. Dann gilt

$$\begin{aligned}
m(n) &\leq 2 \cdot m(n-3) + m(n-4) + r(n-3) + r(n-4) \\
&\quad + 2^{\frac{n-2}{2}} \\
&= 2 \cdot m(n-3) + m(n-4) + 2 \cdot (2^{\frac{n-2}{2}} - 1) \\
&\quad + \tfrac{1}{4} \cdot 2^{\frac{n}{2}} + \tfrac{1}{2} \cdot 2^{\frac{n}{2}} \\
&= 2 \cdot m(n-3) + m(n-4) + \tfrac{7}{4} \cdot 2^{\frac{n}{2}} - 2
\end{aligned}$$

   Annahme, die Behauptung sei bereits für $n-1$, $n-2$, $n-3$ und $n-4$ bewiesen, dann gilt

$$\begin{aligned}
m(n) &\leq 2 \cdot m(n-3) + m(n-4) + \tfrac{7}{4} \cdot 2^{\frac{n}{2}} - 2 \\
&< 2 \cdot 61 \cdot 2^{\frac{(n}{2}-2} + 43 \cdot 2^{\frac{n}{2}-2} + \tfrac{7}{4} \cdot 2^{\frac{n}{2}} - 2 \\
&= \tfrac{61}{2} \cdot 2^{\frac{n}{2}} + \tfrac{43}{4} \cdot 2^{\frac{n}{2}} + \tfrac{7}{4} \cdot 2^{\frac{n}{2}} - 2 \\
&= 43 \cdot 2^{\frac{n}{2}} - 2 \\
&< 43 \cdot 2^{\frac{n}{2}}
\end{aligned}$$

   (b) Sei n ungerade. Dann gilt

$$\begin{aligned}
m(n) &\leq 2 \cdot m(n-3) + m(n-4) + r(n-3) + r(n-4) \\
&\quad + 2^{\frac{n-1}{2}} \\
&= 2 \cdot m(n-3) + m(n-4) + 2 \cdot (2^{\frac{n-3}{2}} + 2^{\frac{n-5}{2}} - 1) \\
&\quad + 2^{\frac{n-5}{2}} + 2^{\frac{n-1}{2}} \\
&= 2 \cdot m(n-3) + m(n-4) + \tfrac{11}{4} \cdot 2^{\frac{n-1}{2}} - 2
\end{aligned}$$

   Annahme, die Behauptung sei bereits für $n-1$, $n-2$, $n-3$ und $n-4$ bewiesen, dann gilt

$$\begin{aligned}
m(n) &\leq 2 \cdot m(n-3) + m(n-4) + \tfrac{11}{4} \cdot 2^{\frac{n-1}{2}} - 2 \\
&< 2 \cdot 43 \cdot 2^{\frac{n-1}{2}-1} + 61 \cdot 2^{\frac{n-1}{2}-2} + \tfrac{11}{4} \cdot 2^{\frac{n-1}{2}} - 2 \\
&= 43 \cdot 2^{\frac{n-1}{2}} + \tfrac{61}{4} \cdot 2^{\frac{n-1}{2}} + \tfrac{11}{\cdot} 2^{\frac{n-1}{2}} - 2 \\
&= 61 \cdot 2^{\frac{n-1}{2}} - 2 \\
&< 61 \cdot 2^{\frac{n-1}{2}}
\end{aligned}$$

6.6 Konstruktion des Minimaxbaumes

Um nun den in Theorem 4 beschriebenen Baum auf dem Computer berechnen zu können, sei folgender Hilfsbaum H konstruiert. Sei n die Tiefe des zu konstruierenden Baumes und v ein Zeilenvektor der Länge $2^{n+1} - 1$. Der binäre Baum H soll wie folgt in den Vektor v eingetragen werden:

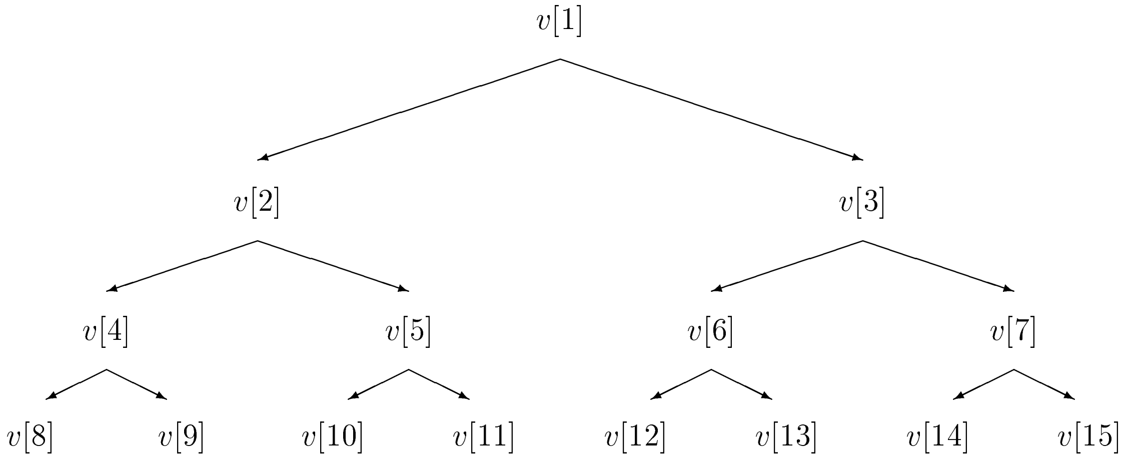
1. Da H ein vollständiger Binärbaum ist, kann jeder Knoten des Baumes durch seine Position in v identifiziert werden. Ebenso sollen Knoten mit ihren Werten identifiziert werden, d.h. hat ein Knoten die j–te Position in v und den Wert w, so soll dies durch die Schreibweise $v[j] = w$ gekennzeichnet werden.

2. Knotenwerte und ihre Positionen werden miteinander identifiziert. Sei z.B. r die Wurzel eines Baumes und der Wurzelwert 7, so wird dies durch die Schreibweise $r = 7$ dargestellt.

3. Die Wurzel des gesamten Baumes sei $v[1]$. Sei $v[i]$ ein Knoten des Baumes, so ist $v[2 \cdot i]$ linker Nachfolger von $v[i]$ und $v[2 \cdot i + 1]$ rechter Nachfolger von $v[i]$.

4. Die Knoten von H einer Tiefe n durchlaufen die Zahlen von 1 bis 2^n.

Damit ist der Aufbau von H:



Aus 3. folgt $v[1] = 1$.

Nimmt man die Knoten von H der Tiefe n als Blattwerte eines binären Minimaxbaumes der Tiefe n, so ist dadurch der Wurzelwert des Minimaxbaumes festgelegt. Dieser Wurzelwert ist natürlich von der Tiefe n abhängig, da bei einer anderen Tiefe eine andere Blattwertverteilung von H genommen wird. Man bezeichne diesen Wert als Wurzelwert der Tiefe n in H. H selbst ist kein Minimaxbaum! Sei $r(n)$ Wurzelwert der Tiefe n in H. Nimmt man die Knoten von H der Tiefe n als Blattwerte eines binären Minimaxbaumes der Tiefe n, so wird der Minimaxbaum in seiner worst–first–Darstellung erzeugt. Daher liegt der Wurzelwert der Tiefe n in H an der Position $2^{n+1} - 1$, d.h. es gilt $r(n) = v[2^{n+1} - 1]$.
Ebenso wie der Wurzelwert r ist auch der Wert für u, aus dem in Kapitel 6 angeführten Baum M, von der Tiefe abhängig. Daher sei in H der Wert durch $u(n)$ gekennzeichnet.
Obwohl auch die Teilbäume $A, T, S, T_i, t_i, S_i, s_i$ aus M in H von der Tiefe abhänging sind, sollen diese Teilbäume in H ebenso bezeichnet werden. T als Teilbaum von M ist zwar ein anderer als der Teilbaum T von H, da jedoch nur noch von dem Baum H die Rede sein wird, ist die Bezeichnung T anstatt $T(n)$ und t_i anstatt $t_i(n)$, usw., ausreichend.

Nach den in Kapitel 6 aufgelisteten Bedingungen gilt:

1. $B(s_i) < r \quad \forall \quad i$
2. $B(t_i) < r \quad \forall \quad i$
3. $u < r$

Die Anzahl der Knoten der Tiefe n im rechten Teilbaum, deren Wert kleiner ist als $r(n)$, ist $r(n) - 1$. $\Rightarrow$ Die Anzahl der Knoten der Tiefe n im rechten Teilbaum, deren Wert größer ist als $r(n)$, ist $2^{n-1} - r(n)$. $r(n)$ läßt sich interpretieren als die Anzahl Blattwerte x im rechten Teilbaum von H, für die gilt $x < B(A)$.
$\Rightarrow$ Ist n gerade, so gilt $r(n+1) = 2 \cdot r(n) - 2$, und ist n ungerade, so gilt $r(n+1) = 2 \cdot r(n)$.

$\Rightarrow v[2^{n+1} - 1] = \frac{1}{6} \cdot 2^n + \frac{2}{3}$, wenn n ungerade ist und $v[2^{n+1} - 1] = \frac{1}{6} \cdot 2^n + \frac{4}{3}$, wenn n gerade ist ($\forall \quad n > 1$).

Beweis durch Induktion nach n:
Es gilt $v[2^{n+1} - 1] = 2 \cdot v[2^n - 1]$, falls n gerade ist und

$v[2^{n+1}-1] = 2 \cdot v[2^n - 1] - 2$, falls n ungerade ist.

Weiter gilt für $n = 1$: $v[2^{1+1}-1] = v[3] = 1$

und für $n = 2$: $v[2^{2+1}-1] = v[7] = 2$.

Annahme, die Behauptung sei für n bewiesen.

1. Sei n gerade.
$$\begin{aligned} \Rightarrow v[2^{n+1}-1] &= 2 \cdot v[2^n - 1] \\ &= 2 \cdot (\tfrac{1}{6} \cdot 2^{n-1} + \tfrac{2}{3}) \\ &= \tfrac{1}{6} \cdot 2^n + \tfrac{4}{3} \end{aligned}$$

2. Sei n ungerade
$$\begin{aligned} \Rightarrow v[2^{n+1}-1] &= 2 \cdot v[2^n - 1] - 2 \\ &= 2 \cdot (\tfrac{1}{6} \cdot 2^{n-1} + \tfrac{4}{3}) - 2 \\ &= \tfrac{1}{6} \cdot 2^n + \tfrac{8}{3} - 2 \\ &= \tfrac{1}{6} \cdot 2^n + \tfrac{2}{3} \end{aligned}$$

Damit gilt für jede Tiefe i:

$v[2^{i+1}-1] = \frac{1}{6} \cdot 2^i + \frac{2}{3}$, wenn i ungerade ist und

$v[2^{i+1}-1] = \frac{1}{6} \cdot 2^i + \frac{4}{3}$, wenn i gerade ist (für $1 \leq i \leq n$).

Weiter gilt, in einer Tiefe von n durchlaufen die Blattwerte von A die Werte von $r(n)+1$ bis $r(n)+2^{n-1}$ und zwar in der inversen Anordnung zu der Anordnung der Blattwerte der Tiefe $n-1$ von H.

$\Rightarrow v[2^n - 1 + i] = 2^n + 1 - v[2^{n-1} - 1 + i] + v[2^{n+1} - 1]$ für $1 \leq i \leq 2^{n-1}$

Der Wert links von der Wurzel ist entweder ein Teilbaum S_k oder ein Teilbaum s_k. Es gilt

$$B(S_k) > B(S_i) > B(T_j) > r > u > B(t_j) > B(s_i) \quad \forall \quad i,j \qquad (1)$$

Demnach ist der Blattwert $v[2^{n+1}-2] = 1$, wenn n gerade ist und

$v[2^{n+1}-2] = 2^n$, wenn n ungerade ist. Damit gilt für H

$v[2^{i+1}-2] = 1$, wenn i gerade ist und $v[2^{i+1}-2] = 2^i$, wenn i ungerade ist

$(\forall \quad 1 \leq i \leq n)$.

Damit folgt aus (1), daß $u(n) = r(n) - 1$ d.h.

$u(n) = v[2^{i+1} - 1 - 2^{i-2}] = v[2^{i+1} - 1] - 1 \quad \forall \quad 2 \leq i \leq n$

Die Blattwerte unter S, d.h. die Blattwerte unter $v[7]$, durchlaufen die Zahlen von 1 bis $\frac{1}{2} \cdot v[2^{n+1} - 1] - 1$ und von $2^n - 2^{n-2} + \frac{1}{2} \cdot v[2^{n+1} - 1] + 1$ bis 2^n. Die Blattwerte unter T, d.h. die Blattwerte unter $v[6]$, durchlaufen die Zahlen von $\frac{1}{2} \cdot v[2^{n+1} - 1]$ bis $v[2^{n+1} - 1] - 1$ und von $2^{n-1} + v[2^{n+1} - 1] + 1$ bis $2^n - 2^{n-2} + \frac{1}{2} \cdot v[2^{n+1} - 1]$.

Demnach gilt für den Wert links von u:

$v[2^{n+1} - 2 - 2^{n-2}] = \frac{1}{2} \cdot r = \frac{1}{2} \cdot v[2^{n+1} - 1]$, falls n gerade ist und

$v[2^{n+1} - 2 - 2^{n-2}] = 2^n - 2^{n-2} + \frac{1}{2} \cdot v[2^{n+1} - 1]$, falls n ungerade ist.

$\Rightarrow v[2^{i+1} - 2 - 2^{i-2}] = \frac{1}{2} \cdot v[2^{i+1} - 1]$, falls i gerade ist und

$v[2^{i+1} - 2 - 2^{i-2}] = 2^i - 2^{i-2} + \frac{1}{2} \cdot v[2^{i+1} - 1]$, falls i ungerade ist $\forall \quad 3 \leq i \leq n$.

Hat T die Teilbäume $T_1, \ldots, T_k$, so gilt $k = \frac{n-2}{2}$.

Der erste Blattwert von links von T_1 hat in einer Tiefe von n die Position $2^n + 2^{n-1}$. Um nun den ersten Blattwert von links von T_i, $i > 1$, in einer Tiefe von n zu erhalten, muß zu diesem Wert die Anzahl aller Blattwerte aller T_j und aller $t_j \quad \forall 1 \leq j < i$ addiert werden. Damit ist die Position des ersten Blattwertes von links von T_i, $i > 1$:

$$2^n + 2^{n-1} + 3 \cdot \sum_{j=1}^{i-1} 2^{n-2 \cdot j-2} = 7 \cdot 2^{n-2} - 2^{n-2 \cdot i}$$

Diese Größe um eins reduziert sei als $bk(i, n)$ bezeichnet.

Wie bereits erwähnt, läßt sich $r(n)$ interpretieren als die Anzahl Blattwerte x im rechten Teilbaum von H, für die gilt $x < B(A)$. Da der rechte Teilbaum von $H \quad 2^{n-1}$ Blattwerte hat, ist die Anzahl Blattwerte x im rechten Teilbaum von H, für die gilt $x > B(A)$: $2^{n-1} - r(n)$. Da T und S identisch aufgebaut sind, gilt, daß ein Blattwert von S_i von der Position um 2^{n-2} größer ist als der entsprechende Blattwert von T_i und vom Wert um $\frac{1}{2} \cdot (2^{n-1} - r(n))$ größer ist als der entspechende Wert von T_i. Analog gilt, daß ein Blattwert von s_i von der Position um 2^{n-2} größer ist als der entsprechende Blattwert von t_i und vom Wert um $\frac{r(n)}{2} - 1$ kleiner ist als der entspechende Wert von t_i.

Damit gilt weiter: der kleinste Blattwert von T_i in einer Tiefe von n ist

$$r(n) + 1 + 4 \cdot \sum_{j=1}^{i} 2^{n-2\cdot j-1}$$

Dieser Wert um eins reduziert sei als $bl(i, n)$ bezeichnet.

Ebenso gilt: der kleinste Blattwert von t_i in einer Tiefe von n ist

$$r(n) - 1 - \frac{1}{2} \cdot \sum_{j=1}^{i} 2^{n-2\cdot j-1}$$

Dieser Wert um eins reduziert sei als $bu(i, n)$ bezeichnet.

Die Blattwerte von T_i in einer Tiefe von n seien in der inversen Anordnung, wie die Blattwerte von H in der Tiefe $n - 2 \cdot i - 1$ und die Blattwerte von t_i in der größenmäßig gleichen Anordnung, wie die Blattwerte von H in einer Tiefe von $n - 2 \cdot i - 2$.
Damit gilt nun für die Blattwerte von T_i in einer Tiefe n:

$$v[bk(i, n) + j] = bl(i, n) + 2^{n-2\cdot i-1} - v[2^{n-2\cdot i-1} - 1 + j] \quad \forall 1 \leq j \leq 2^{n-2\cdot i-1}$$

entsprechend für S_i:

$$v[bk(i, n) + j + 2^{n-2}] = v[bk(i, n) + j] + \frac{1}{2} \cdot (2^{n-1} - r(n)) \quad \forall 1 \leq j \leq 2^{n-2\cdot i-1}$$

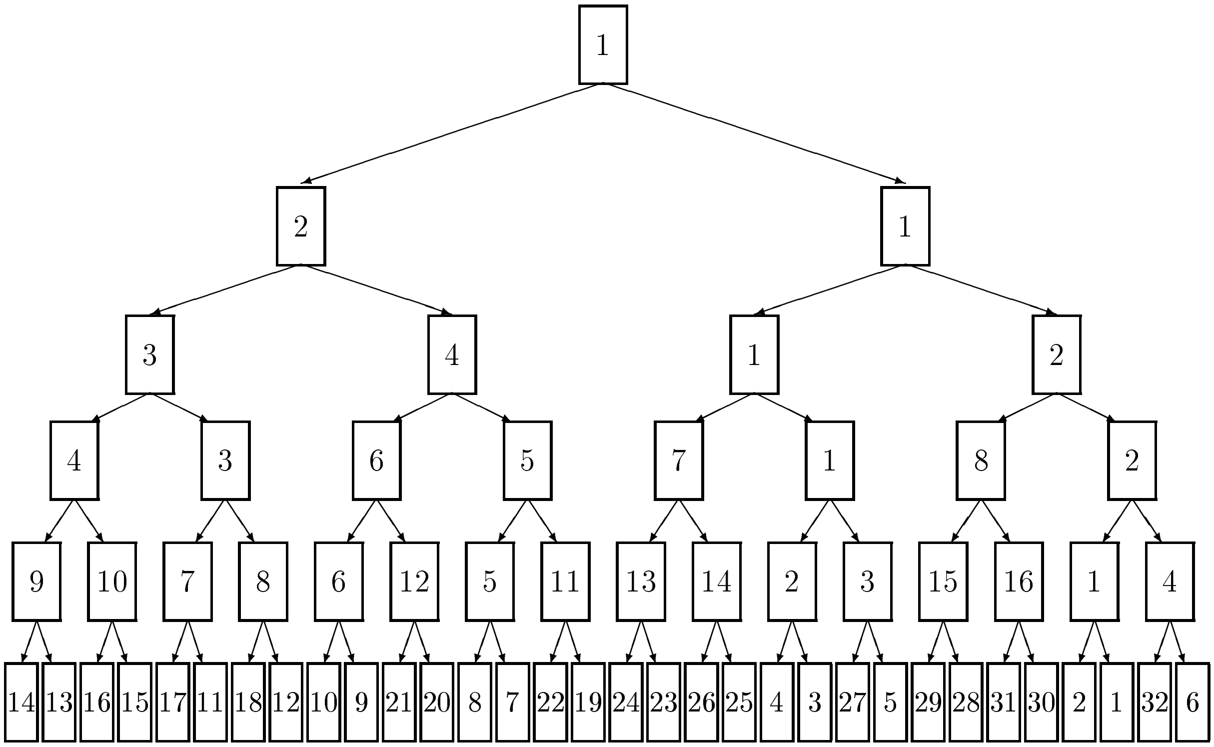
und für t_i:

$$v[bk(i, n) + j + 2^{n-2\cdot i-1}] = bu(i, n) - 1 + v[2^{n-2\cdot i-2} - 1 + j] \quad \forall 1 \leq j \leq 2^{n-2\cdot i-1}$$

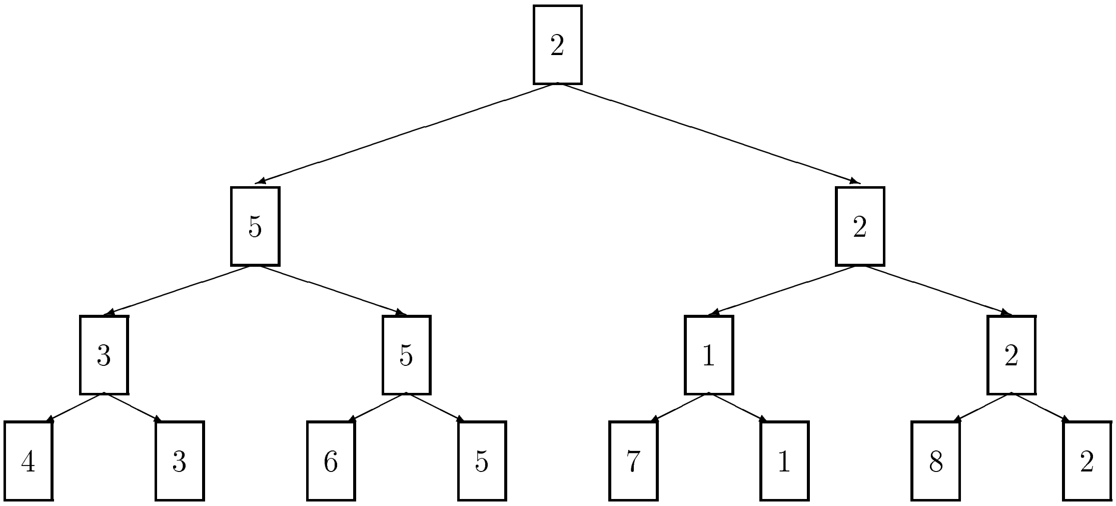
und für s_i:

$$v[bk(i, n) + j + 2^{n-2\cdot i-1}2^{n-2}] = v[bk(i, n) + 2^{n-2\cdot i-1} + j] - \frac{r(n)}{2} + 1$$

$\forall 1 \leq j \leq 2^{n-2\cdot i-1}$
Diese Aussagen gelten für alle $1 \leq i \leq k$ und für alle Tiefen n.
Damit ist der Baum H eindeutig festgelegt und hat bis zu einer Tiefe von $n = 5$ folgende Knotenwerte:



Für $n = 3$ ist der resultierende Minimaxbaum:



7 Ausblick

Die hier gemachten Bemerkungen und Vermutungen sollen helfen, anderen Wissenschaftlern einen Einstieg in diesen Themenbereich zwecks Verbesserung und Verallgemeinerung der Ergebnisse zu erleichtern.

1. Analog zu Kapitel 5 kann gezeigt werden, daß der, in der Konstruktion zu Theorem 4, behandelte binäre Minimaxbaum für alle Tiefen n mit $1 \leq n \leq 6$ bereits optimal ist. Daraus läßt sich dann weiter erkennen, daß die in Theorem 2 angegebene untere Schranke für alle Tiefen $n \geq 6$ echt kleiner ist, als die Anzahl gelesener Blattwerte, die im ungünstigsten Fall bei einem optimalen Baum gelesen werden.

   Weiterhin vermute ich, daß der in Theorem 4 behandelte Baum noch nicht den bestmöglichen Fall darstellt. Denn: Wird der Teilbaum T mit beliebigen Werten α und β durchlaufen, so ist die Anzahl gelesener Blattwerte bei einer Tiefe von n größer als die Anzahl gelesener Blattwerte von M bei einer Tiefe von n (siehe dazu Hilfsbetrachtung 2). D.h. M ist günstiger sortiert als T. Sind also die Teilbäume T und S ebenso aufgebaut wie M, mit gleichbleibenden Blattwerten, so kann ein besseres Ergebnis erzielt werden. Dies ist eventuell ab einer Tiefe von 8 möglich. Die Verbesserung liegt darin, daß nach dem Durchlaufen von T_1 Ausgrenzungen in t_1 vorgenommen werden können. Daher ist also die Anzahl gelesener Blattwerte in einem optimalen Baum ein noch offenes Problem.

2. Für $n = 21$ und für $n \geq 23$ ist die obere Schranke aus Theorem 4 echt kleiner als die obere Schranke aus Theorem 1. Wie bereits erwähnt wurde, stellt $r(n) = k^{\lfloor \frac{n}{2} \rfloor} + k^{\lceil \frac{n}{2} \rceil} - 1$ für $k = 2$ die von Knuth und Moore gefundene untere Schranke dar. Sei $l(n)$ die in Theorem 2 angegebene untere Schranke für $k = 2$, sei $f(n)$ die rekursiv definierte obere Schranke aus Abschnitt 6.4 (mit den entsprechenden Startwerten für $1 \leq n \leq 5$), sei g(n) die in Theorem 4 angegebene obere Schranke ($g(n)$ ist aus der Analyse von $f(n)$ entstanden und stellt eine obere Schranke für $f(n)$ dar.) und sei m(n) die in Theorem 1 angegebene obere Schranke für $k = 2$. Dann wird die Qualität des Ergebnisses an folgender Tabelle deutlich.

Tiefe	KM	Theorem 2	rekursive Def. von Theorem 4	Theorem 4	Theorem 1	
n	$r(n)$	$l(n)$	$f(n)$	$g(n)$	$m(n)$	2^n
0	1	1	1	43	1	1
1	2	2	2	61	2	2
2	3	4	4	86	4	4
3	5	8	8	122	8	8
4	7	14	14	172	14	16
5	11	25	25	244	28	32
6	15	40	41	344	44	64
7	23	66	69	488	88	128
8	31	100	109	688	128	256
9	47	157	176	976	256	512
10	63	230	271	1376	352	1024
11	95	350	426	1952	704	2048
12	127	502	645	2752	928	4096
13	191	749	995	3904	1856	8192
14	255	1060	1488	5504	2368	16384
15	383	1562	2264	7808	4736	32768
16	511	2192	3355	11008	5888	65536
17	767	3205	5052	15616	11776	131072
18	1023	4474	7434	22016	14336	262144
19	1535	6510	11104	31232	28672	524288
20	2047	9058	16249	44032	34304	1048576
21	3071	13141	24115	62464	68608	2097152
22	4095	18248	35131	88064	80896	4194304
23	6143	26426	51865	124928	161792	8388608
24	8191	36652	75280	176128	188416	16777216
25	12287	53021	110657	249856	376832	33554432
26	16383	73486	160122	352256	434176	67108864
27	24575	106238	234513	499712	868352	134217728
28	32767	147182	338463	704512	991232	268435456
29	49151	212701	494174	999424	1982464	536870912

3. Die im Beweis von Theorem 4 konstruierte obere Schranke dürfte sich vermutlich mittels eines ähnlichen Konstruktionsverfahrens auf den k–regulären Fall ausdehnen lassen, d.h. vermutlich läßt sich ein Ergebnis der Form $c(k) \cdot k^{\frac{n}{2}}$ erzielen, wobei $c(k)$ eine nur von k abhängige Konstante darstellen soll. Der Teilbaum T des binären Baumes hat dann im k–regulären Fall nicht nur Knoten T_i sondern $T_{i,1}, \ldots, T_{i,k-1}$ und entsprechendes für die t_j. Dabei seien die Beziehungen zwischen den $T_{i,s}$ und $T_{i,u}$ ebenso wie die Beziehungen zwischen $T_{i,s}$ und $T_{j,u}$ und diese wie die Beziehungen der T_i und S_j im binären Fall.

4. Ich bin mir der Tatsache bewußt, daß die von mir konstruierten Spielbäume in der Praxis wohl kaum auftreten werden (z.B. im Computerschach). So ist es möglich (oder sogar wahrscheinlich), daß die Probleme, die parallele Algorithmen mit meinen Bäumen haben können, in der Praxis nicht auftreten.

5. In jedem geordneten Spielbaum liest der ungerichtete und kompliziertere SSS^*–Algorithmus nur einen Teil der Blattwerte, die vom Alpha–Beta–Algorithmus gelesen werden ([Pea 84, Seite 301]). Daher können die von mir konstruierten Bäume auch geeignet sein, Parallelisierungen des SSS^*–Algorithmus zu testen.

   (siehe zur Parallelisierung des SSS^*–Algorithmus die Dissertation

   [Kra 90])

8 Literatur

[AB 90] Althöfer, I., Balkenhol, B. A game tree with distinct leaf values which is easy for the alpha–beta algorithm. Artificial Intelligence 52 (1991), 183-190.

[BKRU 90] Broder, A.Z., Karlin, A.R., Raghavan, P., and Upfal, E. On the parallel complexity of evaluating game trees. Preprint 1990.

[Hsu 90] Hsu, F.-H. Large scale parallelization of alpha-beta search: an algorithmic and architectural study with computer chess. Ph.D. thesis, Carnegie-Mellon University, Pittsburgh, 1990.

[KM 75] Knuth, D.E. and Moore, R.W. An analysis of alpha-beta pruning. Artificial Intelligence 6 (1975), 293-326.

[Kra 90] Kraas, H.-J., Zur Parallelisierung des SSS^*–Algorithmus. Dissertation, Naturwissenschaftliche Fakultät der TU Braunschweig, 1990.

[KZ 89] Karp, R.M. and Zhang, Y. On parallel evaluation of game trees. Proc. 1st ACM Symposium on Parallel Algorithms and Architectures, 1989, 409-420.

[Mar 86] Marsland, T.A. A review of game-tree pruning. Int Comp. Chess Assoc. Journal 9:1 (March 1986), 3–19.

[Mar 87] Marsland, T.A. Computer chess methods, in "Encyclopedia of Artificial Intelligence" (ed. S.C. Shapiro), Wiley, 1987, 159-171.

[Pea 84] Pearl, J. "Heuristics-Intelligent Search Strategies for Computer Problem Solving". Addison-Wesley, 1984.

[Zha 89] Zhang, Y. Parallel algorithms for combinatorial search problems, Ph.D. thesis, University of California, Berkeley, 1989.

9 Anhang

```
/*
                    M I N I M A X

            (c) 1992 by Bernhard Balkenhol
```

Programmiersprache : C
Aufruf des Programms ohne Parameter. Es wird von dem Programm eine Eingabe erwartet, diese stellt die Tiefe des zu berechnenden Baumes dar. Die Ausgabe ist eine Zahlenfolge mit Werten zwischen 1 und 2^n. Diese Folge soll, von links nach rechts und von oben nach unten gelesen, die Blattwertbelegung eines binären Minimaxbaumes der Tiefe n darstellen. Die Blattwerte sind von links nach rechts in den Baum einzutragen.

```
*/

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* Liste der Unterfunktionen */

void aus(int);
long hoch(int,int);

/*
```

Deklaration eines globalen Vektors v, der den Baum enthalten soll. Die Wurzel des Hilfsbaumes sei v[1]. Ist v[i] ein Knoten des Hilfsbaumes, so sei v[2*i] der linke Nachfolger und v[2*i+1] der rechte Nachfolger.

```
*/

int * v;

void main()  {  /* Beginn des Hauptprogramms */
```

```
/* Deklaration der Variablen */

 int    n,     /* Tiefe des zu berechnenden Baumes   */
        n1,    /* Hilfsvariable für Zwischentiefen   */
        bl,    /* für die Blattwerte von T_i         */
        br,    /* zur Positionsberechnung            */
        bu;    /* für die Blattwerte von t_i         */
        r,     /* Wurzelwert der jeweiligen Tiefe    */
        k,     /* Anzahl der Teilbäume T_i           */
        i, j,  /* Schleifenvariablen                 */
 long   s, a;  /* Hilfsvariablen                     */

/* Einlesen der Tiefe n */

n=-1;
while(n<0) {
   printf("\n Tiefe des Baumes: ");
   scanf("%d", &n);
   }

/*

Allokation des für die Tiefe n benötigten Speicherplatzes für v

*/

a=hoch(2,n+1);

v = (int *) calloc ((unsigned long) a, (unsigned int) sizeof(int));

if (v == NULL)
{
   printf ("\nZu wenig Speicherplatz\n\n");
   exit(0);
   }
```

```
/* Vorbelegung einiger Werte */

v[1]=1;
v[2]=2;
v[3]=1;
v[4]=3;
v[5]=4;
v[6]=1;
v[7]=2;

/*
```

Bei den Tiefen 0, 1 und 2 kann die Ausgabe bereits erfolgen. Anschließend kann das Programm abgebrochen werden.

```
*/

if(n<3) {
   aus(n);
   exit(0);
   }
/*
```

Tiefe n ist größer als 2

```
*/
else {
   a=1;

/*
```

Die Wurzelwerte aller Tiefen von 1 bis n werden berechnet. a entspricht in dem i–ten Schleifendurchlauf 2^i. j dient als Hilfsvariable, um die Position in v festzulegen.

```
*/
```

```
   for(i=1;i<n+1;i++) {
      a=a*2;
      j=2*a-1;
      if(i%2==1) {
         v[j]=(a+4)/6;
         v[j-1]=a;
         }
      else {
         v[j]=(a+8)/6;
         v[j-1]=1;
         }
      if(i>1)
         v[j-a/4]=v[j]-1;
      }
/*
```

Berechnung aller Tiefen von 3 bis n. Hier ist r in jedem Schleifendurchlauf der entsprechende Wurzelwert (siehe r(n) in Kapitel 6 Abschnitt 5). k ist die Anzahl der Teilbäume T_i. Auch bl, br und bu entsprechen in jedem Schleifendurchlauf den in Kapitel 6 Abschnitt 5 definierten Funktionen bl(i,n), br(i,n) und bu(i,n).

```
*/

   for(n1=3;n1<n+1;n1++) {
      if(n1%2==0) {
         s=hoch(2,n1);
         r=(s+8)/6;
         for(i=1;i<s/2+1;i++)
            v[s-1+i]=s/2+1-v[s/2-1+i]+r;
         k=(n1-2)/2;
         bl=r+1;
         bu=r-1;
         for(i=1;i<k+1;i++) {
            a=hoch(2,n1-2*i-1);
            bl=bl+a*4;
```

```
            br=7*s/4-2*a-1;
            bu=bu-a/2;
            for(j=1;j<a+1;j++) {
               v[br+j]=bl+a-v[a-1+j];
               v[br+j+s/4]=bl+a+(s/2-r)/2-v[a-1+j];
               }
            for(j=1;j<a/2+1;j++) {
               v[br+a+j]=bu-1+v[a/2-1+j];
               v[br+a+j+s/4]=bu-r/2+v[a/2-1+j];
               }
            }
         }
/*
```

Entsprechendes für den Fall, daß die Tiefe n1 ungerade ist.

```
*/
      else {
         s=hoch(2,n1);
         r=(s+4)/6;
         for(i=1;i<s/2+1;i++)
            v[s-1+i]=s/2+1-v[s/2-1+i]+r;
         k=(n1-1)/2;
         bl=r+1;
         bu=r-1;
         if(k>1) {
            for(i=1;i<k;i++) {
               a=hoch(2,n1-2*i-1);
               bl=bl+a*4;
               br=7*s/4-2*a-1;
               bu=bu-a/2;
               for(j=1;j<a+1;j++) {
                  v[br+j]=bl+a-v[a-1+j];
                  v[br+j+s/4]=bl+a+(s/2-r)/2-v[a-1+j];
                  }
               for(j=1;j<a/2+1;j++) {
                  v[br+a+j]=bu-1+v[a/2-1+j];
```

```
                  v[br+a+j+s/4]=bu-r/2+v[a/2-1+j];
                  }
               }
            }
         bl=bl+4;
         br=7*s/4-2;
         v[br]=bl;
         v[br+s/4]=bl+(s/2-r)/2;
         }
      }
   }
aus(n); /* Ausgabe der Blattwerte        */
}

/*
```

Hilfsfunktion zur Berechnung von bas^{expo}.

```
*/

long hoch(int bas,int expo)
{
if (expo==1)
   return(bas);
if (expo%2==0)
   hoch(bas*bas,expo/2);
else
   return(bas*hoch(bas*bas,(expo-1)/2));
}
```

```
/*

Hilfsfunktion zur Ausgabe der Blattwerte

*/

void aus(int n)
{
long i, k;
k=hoch(2,n+1);
printf("\nAusgabe der Tiefe %d\n\n", n);
for(i=k/2;i<k;i++)
   printf(" %8d ", v[i]);
}
```