

Title	高さ平衡木にたいする総数の評価と最適構成に関する研究
Author(s)	清水, 道夫
Citation	大阪大学, 1986, 博士論文
Version Type	VoR
URL	https://hdl.handle.net/11094/823
rights	
Note	

Osaka University Knowledge Archive : OUKA

<https://ir.library.osaka-u.ac.jp/>

Osaka University

高さ平衡木にたいする総数の
評価と最適構成に関する研究

1986年5月

清水道夫

高さ平衡木にたいする総数の
評価と最適構成に関する研究

1986年5月

清水道夫

内 容 梗 概

本論文は、筆者が信州大学工学部(情報工学科)に在職中に行った「高さ平衡木にたいする総数の評価と最適構成」に関する研究のうち、主要な結果をまとめたものである。全体は7章から成り立っており、各章の見出しおよび内容梗概はつぎのとおりである。

第1章‘序論’では、関連する研究の背景、および筆者の行った研究の目的、概要を述べる。第2章‘基礎概念’では、数学記号と用語を定義し、木構造および高さ平衡木について説明する。

木の数学理論を考えることは、木構造の性質を明らかにし、認識を深めることを目的としている。また、実際への応用も含んでいる。本論文では、AVL木と2-3木に対し、大きく分けて2つの問題—数え上げ問題と最適構成問題—に関連した観点から考察する。数え上げ問題は、木構造におけるアルゴリズム解析の基礎をなすものであり、符号化にも応用されている。最適構成問題は数学的に興味深い問題であると共に木構造の能率に対して一つの目安を与える。また、データ構造の初期設定への応用が考えられる。

第3章と第4章は、それぞれAVL木と2-3木の数え上げに関するもので、その総数の具体的な表現を考える。第3章‘ある種のAVL木の総数の表現’では、高さ h で n 個の葉を持つAVL木のうち、葉の数が少ない場合は再帰数を用いて表現できること、葉の数が多き場合は2のべき乗の多項式で表現できることを示す。これは、発表論文(1)と(2)をまとめたものである。第4章‘2-3木の総数の表現’では、高さ h で n 個の葉を持つ2-3木の総数が Bruno の公式を用いて再帰的に表現できること、および、二項係数の積和によって表現できることを示す。また、後者の表現から葉の数が少ない場合の2-3木の総数に対する近

似式を導く。これは発表論文(3)とほぼ同じ内容である。

第5章と第6章は、最適構成問題に関するものである。2-3木に関する探索モデルを設定し、平均路長の期待値が最小となるような木の構成を考える。最適木の特徴、コストおよび構成アルゴリズムを示す。第5章‘比較コストに関する最適2-3木’では、見出しの比較回数に関する探索モデルを扱っている。これは発表論文(4)~(6)に関連しているが(4)を中心にまとめる。2-結果比較器を用いたときの最適木がフィボナッチ数を用いて表現できることを示す。第6章‘疎モデル2-3木の最適構成問題’は、発表論文(7)と(8)に関連しているが(8)を中心にまとめる。疎モデル2-3木の節内と節外の路長を一般化したVコストを設定し、それに対する最適木を考察する。

第7章‘結論’では、得られた結果や今後の研究課題について総括的に述べる。

関連発表論文

- (1) 清水, 中村: “ある種のAVL木の総数の表現”, 電子通信学会論文誌(A), J68-A, 4, pp.382-387 (1985).
- (2) 清水, 中村: “ある種のAVL木の総数に対する多項式表現”, 電子通信学会論文誌(A), J68-A, 10, pp.1128-1129 (1985).
- (3) 清水, 中村: “2-3木の総数の表現”, 電子通信学会論文誌(D), J67-D, 6, pp.715-721 (1984).
- (4) 清水, 岡本: “2-結果比較器を用いたときの比較コスト最小2-3木”, 電子通信学会論文誌(D), J63-D, 12, pp.987-993 (1980).
- (5) 岡本, 清水: “最適2-3兄弟木”, 電子通信学会論文誌(D), J64-D, 7, pp.577-584 (1981).
- (6) 清水: “対称比較コスト最小2-3木”, 電子通信学会全国大会, 1250, (1982).
- (7) 清水: “最適 $\alpha - \beta$ 2-3葉探索木”, 電子通信学会論文誌(D), J65-D, 3, pp.370-377 (1982).
- (8) 清水: “探索コスト最小2-3葉探索木”, 電子通信学会論文誌(D), J65-D, 4, pp.451-457 (1982).
(IAOR abs. Vol.26, No.1, 26418 (1983)).
- (9) M.Shimizu, G.Nakamura: “Expressions for the Number of 2-3 Trees with n Leaves,” SCC, 15, pp.84-91 (1985), Scripta Tech..

目 次

第1章	序 論	1
第2章	基礎概念	5
2.1	緒 言	5
2.2	数学的準備	5
2.3	木 構 造	8
2.4	高さ平衡木	9
2.4.1	AVL木	9
2.4.2	2-3木	10
2.5	結 言	11
第3章	ある種のAVL木の総数の表現	13
3.1	緒 言	13
3.2	母 関 数	13
3.3	漸 化 式	15
3.4	準フィボナッチ木にたいする表現	16
3.4.1	木の定義と漸化式	16
3.4.2	定 理	19
3.4.3	再帰数による表現	24
3.5	準完全2分木にたいする表現	28
3.5.1	木の定義と漸化式	28
3.5.2	2のべき乗の多項式による表現	30

3.6	結 言	31
第4章	2-3木の総数の表現	33
4.1	緒 言	33
4.2	母関数	34
4.3	高さ h の木の総数	34
4.4	Bruno の公式を用いた再帰的表現法	36
4.4.1	Bruno の公式	36
4.4.2	べき乗の関数の展開形	37
4.4.3	再帰的表現	39
4.5	二項係数による表現法	41
4.5.1	再帰表現	41
4.5.2	具体的表現	44
4.6	準完全2分木に対する総数の近似式	49
4.7	結 言	53
第5章	比較コストに関する最適2-3木	55
5.1	緒 言	55
5.2	コストの定義	56
5.3	フィボナッチ2分木	58
5.4	比較コスト最小2-3木	64
5.5	構成アルゴリズム	69
5.6	結 言	71

第6章	疎モデル2-3木の最適構成問題	73
6.1	緒言	73
6.2	定義	74
6.3	高さ h のコスト最小木	75
6.4	V の考察	80
6.5	最適木の構成法	85
6.5.1	構成アルゴリズム	85
6.5.2	構成例	88
6.6	結言	89
第7章	結論	91
謝辞	92
文献	93

第1章 序 論

計算機内に格納される情報の集合は、格納後に予想される分類、探索、更新などの操作が効率よく行われるように構造化される。これをデータ構造と呼ぶ。データ構造の概念は、計算機が多量の情報を扱うようになり、効率のよい処理が要求されるようになったことから生じた。データ構造のうちで最も一般的な概念は木構造であろう。これは主に2分木に写像されるものである。木構造における探索能率は線形リストに比べてすぐれているが、最悪の場合両者は一致する。これをさけるには、木構造の自由度に適当な制限をつければよい。しかし、常に完全なバランスを要求すると、バランス化のオーバーヘッドが大きすぎる。そこで、完全にバランスしている木に比べて平均探索能率は多少低下しても、再構成アルゴリズムを効率よく行う方法が考えられた。これが平衡木の方法である。1962年に Adel'son-Vel'skii と Landis によって提案されたAVL木⁽¹⁾はこの端緒をなすものであり、1970年に Hopcroft によって紹介された2-3木⁽⁶⁾とともに最もよく知られている。これらは木の路長(path length)に制限を加えるもので、高さ平衡木(height balanced tree)とよばれている。それ以外にもいくつかの平衡木が提案されている。たとえば、高さ平衡木である兄弟木(brother tree)⁽²⁹⁾や左右部分木の節の数に制限のある節平衡木(node balanced tree)⁽³³⁾などがある。しかし、よく知られている著書⁽³⁾⁻⁽⁶⁾に解説されているものはほとんどAVL木と2-3木に限られている。そこで、本論文では両者をまとめて高さ平衡木とよび、これにたいする考察を行う。

木構造における分類、探索、更新などの操作アルゴリズムの能率を考えると、最も重要な尺度は平均路長であろう。これは実行時間を直接反映しているからである。しかし、一般にこれの解析は困難とされている。それは、操作にともない

木構造が動的に変化するとき、どのような木がどのような頻度で出現するかということがわからなければならないからである。実際、高さ平衡木についての性能評価もコンピュータ・シミュレーションによるものが中心であった。⁽⁷⁾⁻⁽⁹⁾ 異なる n 個の見出しにたいし、 $n!$ 通りの置換が等確率で起こるとするとき、挿入アルゴリズムによって $n!$ 通りの木を構成すれば、大きさ n の木にたいする平均路長が計算される。しかし、一般に木の総数は見出し(節)の数にたいして指数関数的に増大するものであり、コンピュータ・シミュレーションによる評価は時間的にも容量的にもすぐに限界がくる。高さ平衡木の性能、特に平均路長に関する理論的考察は未解決な問題となっている。^{(5),(6)} このような背景から、本論文は高さ平衡木の平均路長にたいする数学的性質を明らかにしようと考え、これに関連した2つの問題—数え上げと最適構成問題—を考察する。

最初の問題は、位相的に異なるすべての木が等確率で出現すると仮定したときの平均路長を考えるものである。このとき平均路長を求める問題は、単にどの種類の木がいくつあるかという数え上げ問題となる。木の数え上げ問題は数学的モデルとして古くから研究が行われている。最初の理論は、A.Cayley が1875年に炭化水素の異性体の数を計算する方法として開発したものとされる。それから近年に至るまでの事情については Knuth(2,p.405) に詳しい。また、解析の方法としては、主に母関数による方法、反転公式による方法、Polya の方法などが知られている。⁽¹⁰⁾ 木構造とくに平衡木として提案された木は、その自由度が制限されており、解析を困難にしている。実際、AVL木の数え上げなどは組合せ問題におけるオープンプロブレムとなっている。⁽¹¹⁾ 本論文では、同じ大きさの高さ平衡木にたいし、最大路長つまり高さが等しい木の総数を考察する。主に、母関数による方法を用いて解析し、総数を再帰的または具体的に表現する。

木の数え上げ問題は、木構造の符号化に応用される。これは木という本来2次

元的な構造を1次元的に表現することであり、データ構造の圧縮などを考える問題である。たとえば、2分木はその総数がカタラン数(catalan number)⁽²⁾として容易に表されることが知られているが、この符号化については最近いくつかのアルゴリズムが提案されている。⁽¹²⁾ 高さ平衡木についても、その総数がわかれば符号化ができると考えられる。

2番目の問題は平均路長の期待値が最小となる木を考える問題である。路長の総和を探索コストとするとき、コスト最小木を構成する問題は一種の最適構成問題である。探索コストを見出し(節)の数でわったものが平均路長である。本研究は、見出しの重みが一律な場合の高さ平衡木について考察する。AVL木の場合は完全にバランスしている木が最適木である。2-3木では、次数の異なる2種類の節の重みに差があるとすれば、いくつかの最適構成問題が考えられる。コスト最小2-3木の特徴、コスト、構成法について考察する。

一般に、最適木の構成問題は数学的に興味深い問題である。その起源は1952年の Huffman のアルゴリズム⁽¹³⁾といわれているが、見出し探索木の最適構成問題については文献(18)に詳しく述べられている。最適構成問題の応用としては、データ構造の初期設定への応用が考えられる。データ集合へのアクセス頻度が低い場合には、初期状態が比較的長く保存されるから有効であろう。

以上述べたように、本研究は高さ平衡木の能率評価の一環として、またデータ構造の符号化、初期設定などをにらんだ基礎的な研究として、数え上げと最適構成問題を考えたものである。

第2章 基礎概念

2.1 緒言

本論文で用いられる基礎的な記号，用語を準備しておく。これらは文献(2)-(6)から抜粋したものである。2.2では数学記号および用語を定義する。これらは主に整数関数である。2.3では木に関する用語を定義する。2.4では，AVL木と2-3木を概説する。

2.2 数学的準備

ガウス記号，数え上げの基本的な関数，位数2の再帰数を定義する。

(I) 実数 x にたいし， $\lceil x \rceil$ は x より小さくない最小の整数を表し， $\lfloor x \rfloor$ は x より大きくない最大の整数を表す。

(II) p_{nk} は順列(permutation)を表す。これは， n 個の対象から k 個選んで一列に並べる方法の数である。

$$p_{nk} = n(n-1) \cdots (n-k+1). \quad (2.1)$$

$n!$ は階乗(factorial)を表す。これは n 個の対象の順列の数である。

$$n! = p_{nn} = n(n-1) \cdots (1). \quad (2.2)$$

これを近似的に表現するものにスターリングの公式がある。

$$n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n. \quad (2.3)$$

$\binom{n}{k}$ は二項係数(binomial coefficients)を表す。これは， n 個から k 個とる組合せの数である。階乗を用いて表現すると

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}, \quad \text{integer } n \geq \text{integer } k \geq 0 \quad (2.4)$$

である。これに関連して二項定理(binomial theorem)がある。

$$(x+y)^r = \sum_k \binom{r}{k} x^k y^{r-k}, \quad \text{integer } r \geq 0. \quad (2.5)$$

$\left[\begin{smallmatrix} n \\ k \end{smallmatrix} \right]$ は第1種のスターリング数を表している。これは2項係数から巾への変形に用いられる。

$$\begin{aligned} n! \binom{x}{n} &= x(x-1)\cdots(x-n+1) \\ &= \left[\begin{smallmatrix} n \\ n \end{smallmatrix} \right] x^n - \left[\begin{smallmatrix} n \\ n-1 \end{smallmatrix} \right] x^{n-1} + \cdots + (-1)^n \left[\begin{smallmatrix} n \\ 0 \end{smallmatrix} \right] \\ &= \sum_k (-1)^{n-k} \left[\begin{smallmatrix} n \\ k \end{smallmatrix} \right] x^k. \end{aligned} \quad (2.6)$$

表2.1に二項係数を、表2.2に第1種のスターリング数を示す。この表から、たとえば

$$\binom{x}{5} = \frac{1}{120}(x^5 - 10x^4 + 35x^3 - 50x^2 + 24x)$$

なることがわかる。

表 2.1 二項係数

r	$\binom{r}{0}$	$\binom{r}{1}$	$\binom{r}{2}$	$\binom{r}{3}$	$\binom{r}{4}$	$\binom{r}{5}$	$\binom{r}{6}$	$\binom{r}{7}$	$\binom{r}{8}$
0	1	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
2	1	2	1	0	0	0	0	0	0
3	1	3	3	1	0	0	0	0	0
4	1	4	6	4	1	0	0	0	0
5	1	5	10	10	5	1	0	0	0
6	1	6	15	20	15	6	1	0	0
7	1	7	21	35	35	21	7	1	0
8	1	8	28	56	70	56	28	8	1

表 2.2 第1種のスターリング数

n	$\begin{bmatrix} n \\ 0 \end{bmatrix}$	$\begin{bmatrix} n \\ 1 \end{bmatrix}$	$\begin{bmatrix} n \\ 2 \end{bmatrix}$	$\begin{bmatrix} n \\ 3 \end{bmatrix}$	$\begin{bmatrix} n \\ 4 \end{bmatrix}$	$\begin{bmatrix} n \\ 5 \end{bmatrix}$	$\begin{bmatrix} n \\ 6 \end{bmatrix}$	$\begin{bmatrix} n \\ 7 \end{bmatrix}$	$\begin{bmatrix} n \\ 8 \end{bmatrix}$
0	1	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0
2	0	1	1	0	0	0	0	0	0
3	0	2	3	1	0	0	0	0	0
4	0	6	11	6	1	0	0	0	0
5	0	24	50	35	10	1	0	0	0
6	0	120	274	225	85	15	1	0	0
7	0	720	1764	1624	735	175	21	1	0
8	0	5040	13068	13132	6769	1960	322	28	1

(Ⅲ) フィボナッチ数(Fibonacci number)は

$$F_1 = 1, \quad F_2 = 1, \quad F_{i+2} = F_{i+1} + F_i, \quad i \geq 1 \quad (2.7)$$

ルカ数(Lucas number)は

$$L_1 = 1, \quad L_2 = 3, \quad L_{i+2} = L_{i+1} + L_i, \quad i \geq 1 \quad (2.8)$$

の再帰式で定義されるもので、その具体的な値は

$$F_i = \frac{1}{\sqrt{5}} \left\{ \left(\frac{1+\sqrt{5}}{2} \right)^i - \left(\frac{1-\sqrt{5}}{2} \right)^i \right\} \cong \frac{\phi^i}{\sqrt{5}} \quad (2.9)$$

$$L_i = \left(\frac{1+\sqrt{5}}{2} \right)^i + \left(\frac{1-\sqrt{5}}{2} \right)^i \cong \phi^i \quad (2.10)$$

となる。ここに ϕ は黄金比(golden ratio)とよばれるもので

$$\phi = (1 + \sqrt{5}) / 2 = 1.618 \dots \quad (2.11)$$

である。また便法として $F_0 = 0$ も定義しておく。フィボナッチ数とルカ数との

間には

$$L_i = F_{i-1} + F_{i+1} \quad (2.12)$$

表 2.3 フィボナッチ数とルカ数

F_1	1	L_1	1
F_2	1	L_2	3
F_3	2	L_3	4
F_4	3	L_4	7
F_5	5	L_5	11
F_6	8	L_6	18
F_7	13	L_7	29
F_8	21	L_8	47
F_9	34	L_9	76
F_{10}	55	L_{10}	123
F_{11}	89	L_{11}	199
F_{12}	144	L_{12}	322
F_{13}	233	L_{13}	521
F_{14}	377	L_{14}	843
F_{15}	610	L_{15}	1364
F_{16}	987	L_{16}	2207
F_{17}	1597	L_{17}	3571
F_{18}	2584	L_{18}	5778
F_{19}	4181	L_{19}	9349
F_{20}	6765	L_{20}	15127

なる関係がある。表 2.3 にフィボナッチ数とルカ数を示す。フィボナッチ数やルカ数のように、数列をつくる各数とその直前にある 2 つの数の和で定義されるものを位数 2 の再帰数とよぶ。⁽¹⁵⁾

2.3 木構造

木(tree)は閉路のない有向グラフ(directed graph)で、節(node)と枝(edge)より成る。節に入ってくる枝を 1 本も持たない節が 1 個あり、これを根(root)とよぶ。根を除く節は共通部分を持たない集合に分割され、この集合は再び木となる。これを部分木(subtree)という。つまり、木は再帰的に定義される。節から出ていく枝を持たない節を外部節(external node)または葉(leaf)という。外部節でない節を内部節(internal node)という。根のレベル(level)を 0 とし、レベル d

の子(son)をレベル $d + 1$ とする。このようにして木のすべての節がレベル付けされたとき、最大のレベル + 1 がその木の高さ(height)である。内部節から出ていく枝が m 以下のとき、その木を m 分木(m -ary tree)とよぶ。特に、 $m = 2$ のときを 2 分木(binary tree)という。2 分木では、根の右子と左子を根とする部分木がそれぞれ右部分木と左部分木である。また、葉のレベルがすべて等しい 2 分木をとくに完全 2 分木(complete binary tree)という。内部節には一意な見出し(key)がおかれ、それが中間順(inorder)で配置されているとき、探索木(search tree)とよばれる。中間順は対称順(symmetric order)ともいわれ、左部分木、根、右部分木に含まれる見出しをそれぞれ a, r, b とするとき $a < r < b$ の関係が成り立つものである。根から節(見出し)までの枝の数が路長(path length)である。

2.4 高さ平衡木

AVL木と 2-3 木に関する説明を行う。これらの木は、与えられた見出しの探索、挿入、削除が、最悪の場合でも $O(\log N)$ 時間で行われるデータ構造である。また、探索と更新だけでなく、集合の分割(partition)や併合(merge)などの操作も効率よく行うことができる⁽³⁾

2.4.1 AVL木

AVL木は、平衡木(balanced tree)または HB(1)木ともよばれ、各節にたいする左、右部分木の高さの差がたかだか 1 であるような 2 分木である。高さ h の AVL木で節の数が最小のものはフィボナッチ木であり、最大のものは完全 2 分木である。図 2.1 に高さ 4 のフィボナッチ木を示す。○は節を表し、□は葉を表す。高さ h のフィボナッチ木の節の数は $F_{h+2} - 1$ であり、高さ h の完全 2 分木の節の数は $2^h - 1$ であるから、高さ h の AVL木の節の数 k は

$$F_{h+2} - 1 \leq k \leq 2^h - 1 \quad (2.13)$$

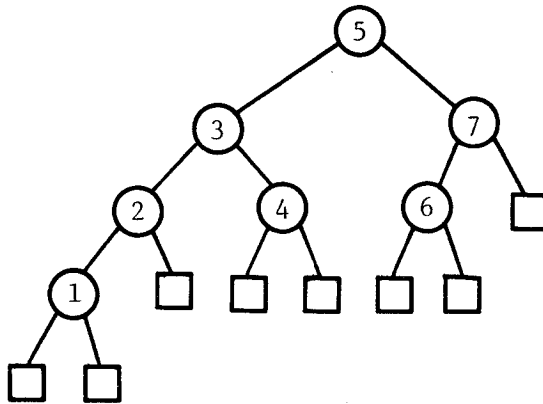


図 2.1 高さ4のフィボナッチ木の例

である。この式より、 k 個の節を持つ AVL 木の高さは

$$\log_2(k+1) \leq h \leq 1.44 \log_2(k+2) - 0.328 \quad (2.14)$$

であることが導かれる。

2.4.2 2-3木

2-3木は3分木の部分集合であり、葉がすべて同じレベルに置かれている木である。内部節は2個または3個の子を持つが、それぞれ2分節(binary node)、3分節(ternary node)とよばれる。2分節には1個の見出しがおかれ、3分節には2個の見出しがおかれる。見出しは中間順に配置されており、探索木として構成されている。図2.2に2-3木の例を示す。節内の数字は見出しを表している。2-3木は最小次数($m=3$)のB-木としても知られている。

2-3木の高さ h と葉の数 n にたいして

$$2^h \leq n \leq 3^h \quad (2.15)$$

が成り立つ。葉の数が $n = 2^h$ の木は内部節がすべて2分節からなる完全2分木であり、 $n = 3^h$ の木は、すべて3分節から成る完全3分木である。式(2.15)か

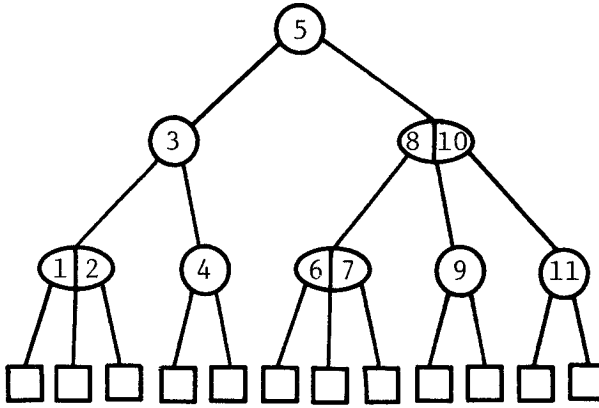


図 2.2 高さ4の2-3木の例

ら、 n 個の葉にたいする木の長さ h は

$$\lfloor \log_2 n \rfloor \geq h \geq \lceil \log_3 n \rceil \quad (2.16)$$

であることがわかる。

Yao⁽¹⁷⁾によると、ランダムな2-3木において N 個の見出しを格納するのに要する平均節数 $k(N)$ は、

$$0.70 N \leq k(N) \leq 0.79 N \quad (2.17)$$

である。

見出しがすべて葉におかれている2-3木をとくに疎モデルの2-3木または2-3葉木とよぶ。この内部節には見出しを探索するための分離値(query)がおかれる。疎モデル2-3木では、所要領域の小さい分離値を選ぶことによって、2-3木よりも領域利用率を上げることができる。第6章では、この疎モデル2-3木について考察している。

2.5 結 言

本論文では、数学的帰納法による証明、母関数による展開、O-記法による表

現をしばしば用いているが，これらについては文献(2)に詳しい．平衡探索木における見出しの挿入，削除に際して，再バランス化を行うアルゴリズムは，回転(rotation)の操作によって行われるが，これは文献(5)と(6)に説明されている．

第3章 ある種のAVL木の総数の表現

3.1 緒言

AVL木の数え上げは組合せ問題におけるオープンプロブレムとなっている⁽¹¹⁾が、ほとんど手がつけられていない。この章では、高さ h で k 個の節(n 個の葉)を持つAVL木の総数にたいする具体的表現を考察する。まず、高さ h のAVL木を表す母関数を導入し、そこから総数についての漸化式を導く。この漸化式は2つの項からなっているが、1つはフィボナッチ木に近い部分(これを準フィボナッチ木とよぶ)、もう1つは完全2分木に近い部分(これを準完全2分木とよぶ)の表現に対応している。それぞれの項を変換することにより、準フィボナッチ木の総数は位数2の再帰数を用いて表現されることを示し、準完全2分木の総数は2のべき乗の多項式で表現されることを示す。

3.2では、AVL木にたいする母関数を示すが、これは、Knuth(6,p.459)によって与えられているものである。3.3では母関数から漸化式を導く。3.4は準フィボナッチ木にたいする表現を考察する。3.4.1は木の定義および漸化式を示す。3.4.2は本章の最も重要な部分である。準フィボナッチ木の総数にたいする多項式表現の骨格を定理として与える。3.4.3では、その具体的な表現を考察する。3.5は準完全2分木にたいする表現である。3.5.1は木の定義および漸化式を示す。3.5.2では、準完全2分木の総数にたいする多項式表現の骨格を示し、3.5.3でその具体的な表現を考察する。

3.2 母関数

高さ h で k 個の節を持つAVL木の総数にたいする母関数を導入する。高さ h のAVL木の母関数を、Knuth⁽⁶⁾にしたがって、

$$B_h(z) = \sum_{k=F_{h+2}-1}^{2^h-1} b(h, k) z^k, \quad k \geq 0 \quad (3.1)$$

と定義する。係数 $b(h, k)$ は高さ h で k 個の節を持つ木の総数を表す。Knuth は、高さ h の AVL 木について、

$$\left. \begin{aligned} B_0(z) &= 1 \\ B_1(z) &= z \\ B_{h+1}(z) &= z B_h(z) \{ B_h(z) + 2 B_{h-1}(z) \} \end{aligned} \right\} \quad (3.2)$$

の関係を示し、これから、

$$\begin{aligned} B_2(z) &= 2 z^2 + z^3 \\ B_3(z) &= 4 z^4 + 6 z^5 + 4 z^6 + z^7 \\ B_4(z) &= 16 z^7 + 32 z^8 + 44 z^9 + \cdots + 8 z^{14} + z^{15} \end{aligned}$$

を与えている。そして一般的に、

$$\begin{aligned} B_h(z) &= 2^{F_{h+1}-1} z^{F_{h+2}-1} + 2^{F_{h+1}-2} L_{h-1} z^{F_{h+2}} \\ &\quad + \text{complicated terms} + 2^{h-1} z^{2^h-2} + z^{2^h-1} \end{aligned} \quad (3.3)$$

の表現を得ている。ここに、 L_i はルカ数である。この Knuth の結果から、調べられる範囲で $b(h, k)$ を求めると、

$$\begin{aligned} b(h, F_{h+2}-1) &= 2^{F_{h+1}-1} \\ b(h, F_{h+2}) &= 2^{F_{h+1}-2} L_{h-1} \\ b(h, 2^h-2) &= 2^{h-1} \\ b(h, 2^h-1) &= 1 \end{aligned} \quad (3.4)$$

となる。本章では、式(3.3)の complicated terms について若干の考察を行う。

3.3 漸化式

高さ h で n 個の葉を持つ AVL 木の総数にたいする漸化式を導く。AVL 木の節の数 k と葉の数 n との間には、 $n = k + 1$ の関係がある。そこで、 $b(h, k)$ のかわりに $c(h, n)$ を

$$c(h, n) = c(h, k + 1) = b(h, k) \quad (3.5)$$

と定義して以下の解析に用いる。これは、 n を変数としたほうが解析過程での表現が少し簡明になるためである。 $c(h, n)$ の母関数を

$$C_h(z) = \sum_{n=F_{h+2}}^{2^h} c(h, n) z^n = z B_h(z) \quad (3.6)$$

と定義すると、式(3.2)は

$$C_0(z) = z, \quad C_1(z) = z^2$$

$$C_h(z) = 2C_{h-2}(z)C_{h-1}(z) + C_{h-1}(z)C_{h-1}(z), \quad h \geq 2 \quad (3.7)$$

となる。これから、 $c(h, n)$ の漸化式が求まる。

$$c(h, n) = 2 \sum_{j=F_h}^{2^{h-2}} c(h-2, j) c(h-1, n-j)$$

$$+ \sum_{j=F_{h+1}}^{2^{h-1}} c(h-1, j) c(h-1, n-j), \quad h \geq 2 \quad (3.8)$$

但し、初期値は

$$c(0, j) = \begin{cases} 1, & j = 1 \\ 0, & j \neq 1 \end{cases} \quad c(1, j) = \begin{cases} 1, & j = 2 \\ 0, & j \neq 2 \end{cases} \quad (3.9)$$

である。式(3.8)が成り立つことは、次章4.5.1に示されている方法と同様にして証明される。式(3.8)と(3.9)を用いて $c(h, F_{h+2} + r)$, $h=1 \sim 6$, $r=0 \sim 4$ を求めると表3.1のようになる。

表 3.1 $c(h, F_{h+2}+r)$, $h=1\sim 6$, $r=0\sim 4$

$h \backslash r$	0	1	2	3	4
1	1				
2	2	1			
3	4	6	4	1	
4	16	32	44	60	70
5	128	448	864	1552	2720
6	4096	22528	67584	159744	334080

3. 4 準フィボナッチ木にたいする表現

3. 4. 1 木の定義と漸化式

AVL木の部分木である準フィボナッチ木を定義し、その総数に関する漸化式を導く。高さ h の AVL木のうちで、葉の数 n が

$$F_{h+2} \leq n < 2F_{h+1} \quad (3. 10)$$

を満たすものを特に準フィボナッチ木と呼ぶ。 $n = F_{h+2} + r$ とおくと、 $0 \leq r < F_{h-1}$ となる。表 3.1 では太線 β より下の部分が準フィボナッチ木に対する $c(h, F_{h+2} + r)$ を示している。次の補題 3.1 で準フィボナッチ木の総数 $c(h, F_{h+2} + r)$ に関する漸化式を与える。

【補題 3.1】 準フィボナッチ木の総数 $c(h, F_{h+2} + r)$, $0 \leq r < F_{h-1}$ について

$$c(h, F_{h+2} + r) = 2 \sum_{j=0}^r c(h-2, F_h + j) c(h-1, F_{h+1} + r - j) \quad (3. 11)$$

が成り立つ。

(証明) 式(3. 8)の右辺第 2 項では、変数 j の範囲が $F_{h+1} \leq j \leq 2^{h-1}$ であり、式(3. 10)より

$$n - j < 2 F_{h+1} - F_{h+1} = F_{h+1}$$

をみただから $c(h-1, n-j) = 0$ となる。つまり、第2項は0となる。

$$c(h, n) = 2 \sum_{j=F_h}^{2^{h-2}} c(h-2, j) c(h-1, n-j).$$

また、

$$n - j < F_{h+1} \text{ のとき } c(h-1, n-j) = 0$$

だから、 j の上限をかきかえて

$$c(h, n) = 2 \sum_{j=F_h}^{n-F_{h+1}} c(h-2, j) c(h-1, n-j).$$

$$n = F_{h+2} + r, \quad 0 \leq r < F_{h-1} \quad \text{とおくと,}$$

$$c(h, F_{h+2} + r) = 2 \sum_{j=F_h}^{F_{h+2} + r} c(h-2, j) c(h-1, F_{h+2} + r - j).$$

となり、 j を $j = j - F_h$ にかえると、式(3.11)が成り立つ。 (証明終)

たとえば、 $c(5, F_7 + 2)$ は、式(3.11)より

$$\begin{aligned} c(5, F_7 + 2) &= 2 \sum_{j=0}^2 c(3, F_5 + j) c(4, F_6 + 2 - j) \\ &= 2(4 \cdot 44 + 6 \cdot 32 + 4 \cdot 16) \\ &= 864 \end{aligned}$$

となり、表3.1の値と一致している。ここで、式(3.11)の右辺に代入できるのは鎖線 α よりも下の部分の数値であることに注意する。つまり、 α と β の間の数値が $c(h, F_{h+2} + r)$ の初期値となる。これらの初期値についての表現が得られなければ、式(3.11)は漸化式としては不完全であるが、本論文では主に式(3.11)の関係だけに着目して議論を進める。

ところで、準フィボナッチ木の総数を考察するには $c(h, n)$ を一定の2の

べき乗で割った整数を扱うほうが便利である。 $d(h, n)$ を

$$d(h, n) = \frac{c(h, n)}{2^{F_{h+3} - n - 1}} \quad (3.12)$$

で定義するとこれは常に整数である。たとえば、 $h=3$ のとき $C_3(z)$ を具体的に計算すると、式(3.7)より、

$$C_3(z) = 4z^5 + 6z^6 + 4z^7 + z^8$$

であるから、

$$\left. \begin{aligned} d(3, 5) &= c(3, 5) / 2^{F_6 - 5 - 1} = 4 / 2^2 = 1 \\ d(3, 6) &= c(3, 6) / 2^{F_6 - 6 - 1} = 6 / 2^1 = 3 \\ d(3, 7) &= c(3, 7) / 2^{F_6 - 7 - 1} = 4 / 2^0 = 4 \\ d(3, 8) &= c(3, 8) / 2^{F_6 - 8 - 1} = 1 / 2^{-1} = 2 \end{aligned} \right\}$$

となる。そこで以下では $c(h, n)$ のかわりに $d(h, n)$ を考察する。式(3.8)と(3.12)より $d(h, F_{h+2} + r)$, $h=1\sim 9$, $r=0\sim 5$ を求めると表3.2のようになる。表3.2の γ と δ は、それぞれ表3.1の α と β に対応している。

表 3.2 $d(h, F_{h+2} + r)$, $h=1\sim 9$, $r=0\sim 5$

$h \backslash r$	0	1	2	3	4	5
1	1					
2	1	1				
3	1	3	4	2		
4	1	4	11	30	70	112
5	1	7	27	97	340	1072
6	1	11	66	312	1305	5039
7	1	18	170	1168	6678	33812
8	1	29	434	4538	37667	265905
9	1	47	1126	18448	233681	2449663

【補題3.2】 準フィボナッチ木の総数 $d(h, F_{h+2} + r)$, $0 \leq r < F_{h-1}$ について

$$d(h, F_{h+2} + r) = \sum_{j=0}^r d(h-2, F_h + j) d(h-1, F_{h+1} + r - j) \quad (3.13)$$

が成り立つ.

(証明) 式(3.11)と(3.12)より

$$\begin{aligned} c(h, F_{h+2} + r) &= 2 \sum_{j=0}^r d(h-2, F_h + j) 2^{F_{h+1} - (F_h + j) - 1} \\ &\quad \cdot d(h-1, F_{h+1} + r - j) 2^{F_{h+2} - (F_{h+1} + r - j) - 1} \\ &= \sum_{j=0}^r d(h-2, F_h + j) d(h-1, F_{h+1} + r - j) \\ &\quad \cdot 2^{F_{h+3} - (F_{h+2} + r) - 1} \end{aligned}$$

となる. 両辺を $2^{F_{h+3} - (F_{h+2} + r) - 1}$ で割ると, 式(3.13)が成り立つ.

(証明終)

たとえば, $d(7, F_9 + 2)$ は式(3.13)より

$$\begin{aligned} d(7, F_9 + 2) &= \sum_{j=0}^2 d(5, F_7 + j) d(6, F_8 + 2 - j) \\ &= 1 \cdot 66 + 7 \cdot 11 + 27 \cdot 1 \\ &= 170 \end{aligned}$$

となり, 表3.2の値に一致していることがわかる.

3.4.2 定理

準フィボナッチ木の総数 $d(h, F_{h+2} + r)$ を具体的に表現するには, 次の定理3.1が重要である. これは, $d(h, F_{h+2} + r)$ の表現の骨格を与えるものであり, 準フィボナッチ木の総数に限らず, 漸化式(3.11)で表現されるものについて

てはすべて成り立つものである。ただし、 $d(h, F_{h+2}+r)$ の初期値(表3.2の γ と δ の間の数値)については、3.4.3 のおわりで触れることにして、それまでは $d(h, F_{h+2}+r)$ の表現が再帰的に式(3.11)を満たすことを証明する。つまり、 $h-2$ と $h-1$ で成り立つと仮定したとき、 h でも成り立つことを示す。

【定理3.1】 準フィボナッチ木の総数 $d(h, F_{h+2}+r)$, $0 \leq r < F_{h-1}$ は

$$d(h, F_{h+2}+r) = \frac{1}{r!} \sum_{j=0}^r \binom{r}{j} L_{h-1}^j \lambda_{h-1}(r-j) \quad (3.14)$$

と表現される。ここに、 L_{h-1} はルカ数であり、 $\lambda_{h-1}(i)$ は

$$\lambda_{h-1}(i) = \sum_{t=0}^i \binom{i}{t} \lambda_{h-3}(t) \lambda_{h-2}(i-t) \quad (3.15)$$

で定義される整数とする。

(証明) 式(3.14)が $h-1$ と $h-2$ で成り立つとしたとき、 h でも成り立つことを示す。式(3.13)と(3.14)より

$$\begin{aligned} & d(h, F_{h+2}+r) \\ &= \sum_{i=0}^r \left\{ \frac{1}{i!} \sum_{u=0}^i \binom{i}{u} L_{h-3}^u \lambda_{h-3}(i-u) \right\} \\ & \quad \cdot \left\{ \frac{1}{(r-i)!} \sum_{v=0}^{r-i} \binom{r-i}{v} L_{h-2}^v \lambda_{h-2}(r-i-v) \right\} \\ &= \frac{1}{r!} \sum_{i=0}^r \binom{r}{i} \sum_{u=0}^i \sum_{v=0}^{r-i} \binom{i}{u} L_{h-3}^u \lambda_{h-3}(i-u) \\ & \quad \cdot \binom{r-i}{v} L_{h-2}^v \lambda_{h-2}(r-i-v) \end{aligned}$$

となり、変数 u と v をそれぞれ

$$\begin{cases} s = u \\ j = u + v \end{cases}$$

に変換すると,

$$= \frac{1}{r!} \sum_{i=0}^r \binom{r}{i} \sum_{s=0}^i \sum_{j=s}^{r-i+s} \binom{i}{s} L_{h-3}^s \lambda_{h-3}(i-s) \\ \cdot \binom{r-i}{j-s} L_{h-2}^{j-s} \lambda_{h-2}(r-i-j+s)$$

となる. j と s の総和の領域は図 3.1 の I である. ここで, II と III の領域を考え, この領域での $\binom{i}{s} \binom{r-i}{j-s}$ をしらべる. 領域 II では,

$$j-s > r-i \geq 0 \text{ のため } \binom{r-i}{j-s} = 0$$

となり, 領域 III では,

$$s > i \geq 0 \text{ のため } \binom{i}{s} = 0$$

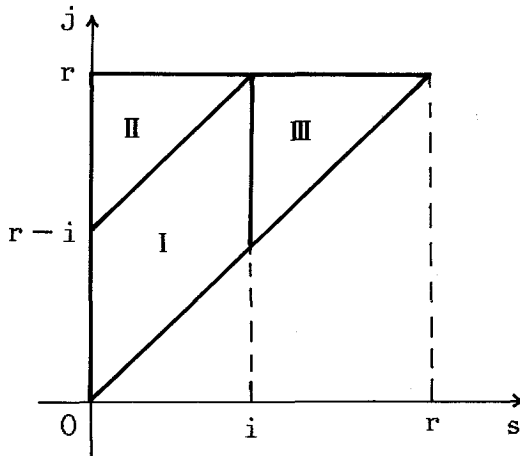


図 3.1 総和の領域

$$\sum_{i=0}^r \sum_{s=0}^i \sum_{j=s}^{r-i+s} \rightarrow \sum_{i=0}^r \sum_{s=0}^r \sum_{j=s}^r \rightarrow \sum_{i=0}^r \sum_{j=0}^r \sum_{s=0}^j$$

ともかける。これから

$$= \frac{1}{r!} \sum_{i=0}^r \binom{r}{i} \sum_{j=0}^r \sum_{s=0}^j \binom{i}{s} L_{h-3}^s \lambda_{h-3}(i-s) \\ \cdot \binom{r-i}{j-s} L_{h-2}^{j-s} \lambda_{h-2}(r-i-j+s)$$

となり、総和を入れかえると

$$= \frac{1}{r!} \sum_{j=0}^r \sum_{s=0}^j L_{h-3}^s L_{h-2}^{j-s} \sum_{i=0}^r \binom{r}{i} \binom{i}{s} \binom{r-i}{j-s} \lambda_{h-3}(i-s) \\ \cdot \lambda_{h-2}(r-i-j+s)$$

を得る。ここで、

$$\binom{r}{i} \binom{i}{s} \binom{r-i}{j-s} = \binom{r}{j} \binom{j}{s} \binom{r-j}{i-s}$$

に注意すると、

$$= \frac{1}{r!} \sum_{j=0}^r \binom{r}{j} \sum_{s=0}^j \binom{j}{s} L_{h-3}^s L_{h-2}^{j-s} \\ \cdot \sum_{i=0}^r \binom{r-j}{i-s} \lambda_{h-3}(i-s) \lambda_{h-2}(r-i-j+s)$$

となる。しかし、

$$i < s \text{ では } \binom{r-j}{i-s} = 0$$

となるから、

$$t = i - s$$

とおけば、

$$\begin{aligned} &= \frac{1}{r!} \sum_{j=0}^r \binom{r}{j} \sum_{s=0}^j \binom{j}{s} L_{h-3}^s L_{h-2}^{j-s} \\ &\quad \cdot \sum_{t=0}^{r-j} \binom{r-j}{t} \lambda_{h-3}(t) \lambda_{h-2}(r-j-t) \end{aligned}$$

ともかける。ところが、 $j \geq s$ のため、

$$t > r - j \text{ では } \binom{r-j}{t} = 0$$

となる。よって

$$\begin{aligned} &= \frac{1}{r!} \sum_{j=0}^r \binom{r}{j} \sum_{s=0}^j \binom{j}{s} L_{h-3}^s L_{h-2}^{j-s} \\ &\quad \cdot \sum_{t=0}^{r-j} \binom{r-j}{t} \lambda_{h-3}(t) \lambda_{h-2}(r-j-t) \\ &= \frac{1}{r!} \sum_{j=0}^r \binom{r}{j} (L_{h-3} + L_{h-2})^j \\ &\quad \cdot \sum_{t=0}^{r-j} \binom{r-j}{t} \lambda_{h-3}(t) \lambda_{h-2}(r-j-t) \\ &= \frac{1}{r!} \sum_{j=0}^r \binom{r}{j} L_{h-1}^j \sum_{t=0}^{r-j} \binom{r-j}{t} \lambda_{h-3}(t) \lambda_{h-2}(r-j-t) \end{aligned}$$

となる。

(証明終)

定理 3.1 より、 $d(h, F_{h+2} + r)$, $r = 0 \sim 5$ を示すと次のようになる。

$$\begin{aligned}
d(h, F_{h+2}) &= \lambda_{h-1}(0) \\
d(h, F_{h+2}+1) &= L_{h-1} \lambda_{h-1}(0) + \lambda_{h-1}(1) \\
d(h, F_{h+2}+2) &= \frac{1}{2} [L_{h-1}^2 \lambda_{h-1}(0) + 2L_{h-1} \lambda_{h-1}(1) + \lambda_{h-1}(2)] \\
d(h, F_{h+2}+3) &= \frac{1}{6} [L_{h-1}^3 \lambda_{h-1}(0) + 3L_{h-1}^2 \lambda_{h-1}(1) \\
&\quad + 3L_{h-1} \lambda_{h-1}(2) + \lambda_{h-1}(3)] \\
d(h, F_{h+2}+4) &= \frac{1}{24} [L_{h-1}^4 \lambda_{h-1}(0) + 4L_{h-1}^3 \lambda_{h-1}(1) \\
&\quad + 6L_{h-1}^2 \lambda_{h-1}(2) + 4L_{h-1} \lambda_{h-1}(3) + \lambda_{h-1}(4)] \\
d(h, F_{h+2}+5) &= \frac{1}{120} [L_{h-1}^5 \lambda_{h-1}(0) + 5L_{h-1}^4 \lambda_{h-1}(1) \\
&\quad + 10L_{h-1}^3 \lambda_{h-1}(2) + 10L_{h-1}^2 \lambda_{h-1}(3) \\
&\quad + 5L_{h-1} \lambda_{h-1}(4) + \lambda_{h-1}(5)]
\end{aligned}$$

(3. 16)

3. 4. 3 再帰数による表現

前段落の結果に基づき、 $d(h, F_{h+2}+r)$, $0 \leq r < F_{h-1}$ を具体的に表現する。その方法として式(3.15)の $\lambda_{h-1}(i)$ を位数2の再帰数で表現する。

まず、 $i=0, 1$ のときは、

$$\left. \begin{aligned}
\lambda_{h-1}(0) &= 1 \\
\lambda_{h-1}(1) &= 0
\end{aligned} \right\} \quad (3. 17)$$

となることを示す。h-2とh-1で成り立つとして、hのときを示せば十分である。

i = 0のとき、式(3.15)より

$$\lambda_{h-1}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \lambda_{h-3}(0) \lambda_{h-2}(0)$$

である。仮定により、 $\lambda_{h-3}(0) = 1$ 、 $\lambda_{h-2}(0) = 1$ であり、 $\begin{pmatrix} 0 \\ 0 \end{pmatrix} = 1$ だから右辺は1になる。

i = 1のとき、式(3.15)より

$$\lambda_{h-1}(1) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \lambda_{h-3}(0) \lambda_{h-2}(1) + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \lambda_{h-3}(1) \lambda_{h-2}(0)$$

である。仮定により $\lambda_{h-3}(1) = 0$ 、 $\lambda_{h-2}(1) = 0$ であるから右辺は0になる。

i \geq 2のとき、位数2の再帰数 $R_{h-1}(1) \sim R_{h-1}(i-1)$ を用いて $\lambda_{h-1}(i)$ を表現する。 $\lambda_{h-1}(2) \sim \lambda_{h-1}(6)$ は次のようになる。 $\lambda_{h-1}(i)$, $i \geq 7$ についても、再帰数の性質に注意して漸化式(3.15)を満たすようにくふうすれば、その表現が逐次に得られると考えられる。

$$\left. \begin{aligned} \lambda_{h-1}(2) &= R_{h-1}(1) \\ \lambda_{h-1}(3) &= 2 R_{h-1}(2) \\ \lambda_{h-1}(4) &= 3 R_{h-1}^2(1) + 6 R_{h-1}(3) \\ \lambda_{h-1}(5) &= 20 R_{h-1}(2) R_{h-1}(1) + 24 R_{h-1}(4) \\ \lambda_{h-1}(6) &= 15 R_{h-1}^3(1) + 90 R_{h-1}(1) R_{h-1}(3) \\ &\quad + 40 R_{h-1}^2(4) + 120 R_{h-1}(5) \end{aligned} \right\} \quad (3.18)$$

この右辺最終項 $R_{h-1}(i-1)$ に係数 $(i-1)!$ をつけた(具体的には 1, 2, 6,

2 4, 1 2 0)のは, 式(3.14)の右辺の係数の和が $r!$ になるように調整したためである.

式(3.18)の第1行が $h-2$ と $h-1$ で成り立つと仮定したとき, h で成り立つことを示す. $i = 2$ のとき式(3.15)より

$$\begin{aligned} \lambda_{h-1}(2) &= \binom{2}{0} \lambda_{h-3}(2) \lambda_{h-2}(0) + \binom{2}{1} \lambda_{h-3}(1) \\ &\quad \cdot \lambda_{h-2}(1) + \binom{2}{2} \lambda_{h-3}(0) \lambda_{h-2}(2) \end{aligned}$$

であり, 式(3.17)より

$$\lambda_{h-1}(2) = \binom{2}{0} \lambda_{h-3}(2) + \binom{2}{2} \lambda_{h-2}(2)$$

となる. 仮定により $\lambda_{h-3}(2) = R_{h-3}(1)$, $\lambda_{h-2}(2) = R_{h-2}(1)$ であり, $\binom{2}{0} = 1$, $\binom{2}{2} = 1$ だから,

$$\lambda_{h-1}(2) = R_{h-3}(1) + R_{h-2}(1) = R_{h-1}(1)$$

が成り立つ.

同様にして $i = 3 \sim 6$ についても逐次に証明される.

式(3.17)と(3.18)を式(3.16)に代入すると $d(h, F_{h+2}) \sim d(h, F_{h+2} + 5)$ は次のようになる.

$$d(h, F_{h+2}) = 1$$

$$d(h, F_{h+2} + 1) = L_{h-1}$$

$$d(h, F_{h+2} + 2) = \frac{1}{2} [L_{h-1}^2 + R_{h-1}(1)]$$

$$\begin{aligned}
d(h, F_{h+2}+3) &= \frac{1}{6} [L_{h-1}^3 + 3L_{h-1}R_{h-1}(1) + 2R_{h-1}(2)] \\
d(h, F_{h+2}+4) &= \frac{1}{24} [L_{h-1}^4 + 6L_{h-1}^2R_{h-1}(1) + 4L_{h-1} \\
&\quad \cdot 2R_{h-1}(2) + (3R_{h-1}^2(1) + 6R_{h-1}(3))] \\
d(h, F_{h+2}+5) &= \frac{1}{120} [L_{h-1}^5 + 10L_{h-1}^3R_{h-1}(1) \\
&\quad + 10L_{h-1}^2 \cdot 2R_{h-1}(2) + 5L_{h-1}(3R_{h-1}^2(1) \\
&\quad + 6R_{h-1}(3)) + (20R_{h-1}(2)R_{h-1}(1) \\
&\quad + 24R_{h-1}(4))]
\end{aligned}
\tag{3. 19}$$

ここで、式(3. 4)と(3.19)の最初の2項が対応することに注意する。

ところで、式(3.14)または(3.19)を用いて実際の値を求めるには、 $\lambda_{h-1}(i)$ または $R_{h-1}(i)$ の初期値を知る必要がある。しかし、初期値の系列に関する表現は得られなかったので、ここでは、 $d(h, F_{h+2}+r)$ の初期値の領域(表3. 2の γ と δ との間)において式(3.19)の表現が成り立つとしたときの $R_{h-1}(i)$ を初期値とし、それを $i=1\sim 5$ についてだけ示しておく。

$$\left.
\begin{aligned}
R_2(1) &= -1, & R_3(1) &= 6 \\
R_3(2) &= 22, & R_4(2) &= 67 \\
R_3(3) &= 6, & R_4(3) &= 77 \\
R_4(4) &= 147, & R_5(4) &= 621 \\
R_4(5) &= 125, & R_5(5) &= 2171
\end{aligned}
\right\}
\tag{3. 20}$$

念のため、式(3.20)の第1行が成り立つことを示す。 $i = 1$ のとき、式(3.19)の第3行において $h = 3, 4$ とするとそれぞれ

$$d(4, F_6 + 2) = \frac{1}{2} [L_2^2 + R_2(1)] = \frac{1}{2} [3^2 + (-1)] = 8/2 = 4$$

$$d(3, F_5 + 2) = \frac{1}{2} [L_3^2 + R_3(1)] = \frac{1}{2} [4^2 + 6] = 22/2 = 11$$

となり、表3.2の値と一致している。 $i = 2 \sim 5$ についても同様に示すことができる。

3.5 準完全2分木にたいする表現

3.5.1 木の定義と漸化式

高さ h のAVL木のうちで葉の数 n が

$$3 \cdot 2^{h-2} < n \leq 2^h \tag{3.21}$$

を満たすものを特に準完全2分木と呼ぶ。表3.3は、漸化式(3.8)によって $c(h, 2^h - i)$, $h = 1 \sim 7$, $i = 0 \sim 6$ を計算したもので、太線 α より下の部分が準完全2分木に対する $c(h, 2^h - i)$ を示している。比 $R(h)$ を

表 3.3 $c(h, 2^h - i)$

$h \backslash i$	0	1	2	3	4	5	6
1	1						
2	1	2					
3	1	4	6	4			
4	1	8	28	56	70	60	44
5	1	16	120	560	1820	4376	8104
6	1	32	496	4960	35960	201392	906640
7	1	64	2016	41664	635376	7624544	42871188

$$R(h) = \frac{\text{高さ } h \text{ の準完全 2 分木の葉数の範囲}}{\text{高さ } h \text{ の AVL 木の葉数の範囲}}$$

とおくと、式(2.13)と(3.21)より、

$$R(h) \cong \frac{2^h - 3 \cdot 2^{h-2}}{2^h - F_{h+2} + 1}$$

である。式(2.9)より、

$$= \frac{2^h - 3 \cdot 2^{h-2}}{2^h - \phi^{h+2} / \sqrt{5}}$$

となり、分子と分母を 2^h でわると

$$= \frac{1}{4 \left(1 - \frac{\phi^2}{\sqrt{5}} \left(\frac{\phi}{2} \right)^h \right)}$$

となる。 $\phi \cong 1.618$ だから

$$\cong \frac{1}{4(1 - 1.17 (0.809)^h)} > \frac{1}{4}$$

となる。このことから、高さ h の準完全 2 分木の葉数の範囲は、高さ h の AVL 木のうちでかなりの部分を占めているといえる。準完全 2 分木の総数に対する漸化式は、

$$f(h, r) = c(h, 2^h - r) = c(h, n) \quad (3.22)$$

とかくと、

$$f(h, r) = \sum_{j=0}^r f(h-1, j) f(h-1, r-j) \quad (3.23)$$

で表される。この式は、補題 3.1 と同様に、式(3.21)の条件下では式(3.8)の

第1項が0になることおよび第2項の総和の範囲が修正されることから導かれる。但し、式(3.23)の初期値は表3.3の太線 α のすぐ下の数値である。

3.5.2 2のべき乗の多項式による表現

次式は多項式表現の骨格を与えるものであり、漸化式(3.23)を変換することにより得られる。

$$f(h, r) = \frac{1}{r!} \sum_{j=0}^r \binom{r}{j} \lambda_{h-1}(j) (2^{h-1})^{r-j} \quad (3.24)$$

ここに、 $\lambda_{h-1}(j)$ は

$$\lambda_{h-1}(j) = \sum_{t=0}^j \binom{j}{t} \lambda_{h-2}(t) \lambda_{h-2}(j-t) \quad (3.25)$$

で表現される整数である。式(3.24)と(3.25)が再帰的に成り立つことは、定理3.1と同様に証明される。 $\lambda_{h-1}(j)$ の一般的表現は得られていないが、 $\lambda_{h-1}(0) \sim \lambda_{h-1}(6)$ は次のように予想される。

$$\left. \begin{aligned} \lambda_{h-1}(0) &= 1 \\ \lambda_{h-1}(1) &= 0 \\ \lambda_{h-1}(2) &= -(2^{h-1}) \\ \lambda_{h-1}(3) &= 2(2^{h-1}) \\ \lambda_{h-1}(4) &= 3(2^{h-1})^2 - 6(2^{h-1}) \\ \lambda_{h-1}(5) &= -20(2^{h-1})^2 + 84(2^{h-1}) \\ \lambda_{h-1}(6) &= -15(2^{h-1})^3 + 130(2^{h-1})^2 - 1560(2^{h-1}) \end{aligned} \right\} \quad (3.26)$$

これらを式(3.24)に代入すると $f(h, 0) \sim f(h, 6)$ の多項式表現が得られる。

$$f(h, 0) = 1$$

$$f(h, 1) = 2^{h-1}$$

$$f(h, 2) = \frac{1}{2} [(2^{h-1})^2 - (2^{h-1})]$$

$$f(h, 3) = \frac{1}{6} [(2^{h-1})^3 - 3(2^{h-1})^2 + 2(2^{h-1})]$$

$$f(h, 4) = \frac{1}{24} [(2^{h-1})^4 - 6(2^{h-1})^3 + 11(2^{h-1})^2 - 6(2^{h-1})]$$

$$f(h, 5) = \frac{1}{120} [(2^{h-1})^5 - 10(2^{h-1})^4 + 35(2^{h-1})^3 - 50(2^{h-1})^2 + 84(2^{h-1})]$$

$$f(h, 6) = \frac{1}{720} [(2^{h-1})^6 - 15(2^{h-1})^5 + 85(2^{h-1})^4 - 225(2^{h-1})^3 + 634(2^{h-1})^2 - 1560(2^{h-1})]$$

(3. 27)

実際、これらが表 3. 3 の太線 α より下の部分の数値を表していることは容易に確かめられる(式(3.22)に注意). 式(3.25)の $\lambda_{h-1}(j)$ も $f(h, r)$ と同様の議論によって 2 のべき乗の多項式に展開されることに注意すれば, $j \geq 7$ についても $\lambda_{h-1}(j)$ が逐次に求まると考えられる.

3. 6 結 言

高さ h で n 個の葉を持つ AVL 木の総数にたいする具体的表現が部分的に明らかになった. 式(3. 8)の両項が別々に考察できる部分について総数の表現を考察

したが、両項を一緒にした場合の数え上げつまり高さ h の AVL 木の完全な数え上げは今後の課題である。また、本方法を多分木の数え上げに拡張することも考えられる。

第4章 2-3木の総数の表現

4.1 緒言

2-3木の数え上げ問題については、Miller などの研究がある。⁽¹⁹⁾ かれらは、 n 個の葉を持つ2-3木の総数を a_n とすると、ある定数 u_1, u_2 が存在して

$$\frac{1}{n} \phi^n u_1 < a_n < \frac{1}{n} \phi^n u_2$$

が成り立つことを示した。ここで ϕ は黄金比である。また、Gupta などは2-3木の符号化に関する研究の中で、探索グラフを用いて a_n を $O(n^2)$ で数え上げるアルゴリズムを示している。⁽²²⁾

本章では、高さ h で n 個の葉を持つ2-3木の正確な総数を表現する。この総数がわかれば、それをすべての h について加えることにより、 n 個の葉を持つ2-3木の総数 a_n を求めることができる。Miller などの研究はこの a_n の漸近的性質を考察したものである。

次節以降の内容は、つぎのようである。4.2では、高さ h の2-3木を表現する母関数を与える。4.3では、高さ h の2-3木の総数を考察する。4.4では、Bruno の公式を用いて、 n 個の葉を持つ高さ h の2-3木の総数 $a(h, n)$ を再帰的に表現する。4.4.1では、複合関数の n 次導関数を表現する Bruno の公式を概説する。4.4.2では、Bruno の公式を用いて一般的なべき乗の関数の展開形を示す。4.4.3では、その特別な場合として $a(h, n)$ を表現する。4.5では、二項係数を用いて $a(h, n)$ を表現する。4.5.1では、二項係数を用いた $a(h, n)$ の再帰関数を示し、4.5.2で、そこから具体的表現を導く。4.6では、4.5.2の表現から、準完全2分木に対する $a(h, n)$ の近似式を求める。

4.2 母関数

高さ h の 2-3 木の母関数を,

$$F_h(z) = \sum_{n=2^h}^{3^h} a(h, n) z^n, \quad h \geq 1 \quad (4.1)$$

と定義する。係数 $a(h, n)$ は、高さ h で n 個の葉を持つ木の総数を表す。

$F_h(z)$ は、 $F_{h-1}(z)$ を用いて再帰的に表現できる。高さ h の 2-3 木が 2 分節を根に持つときは、左、右部分木がそれぞれ $F_{h-1}(z)$ で表現され、3 分節を根に持つときは、左、中、右部分木がそれぞれ $F_{h-1}(z)$ で表される。よって、2-3 木の母関数の性質から、

$$F_h(z) = \{F_{h-1}(z)\}^2 + \{F_{h-1}(z)\}^3 \quad (4.2)$$

となる。ただし、便宜的に高さ 0 の木は 1 個の葉を表すものとする。つまり

$F_0(z) = z$ である。これから、

$$F_1(z) = z^2 + z^3$$

$$\begin{aligned} F_2(z) &= (z^2 + z^3)^2 + (z^2 + z^3)^3 \\ &= z^4 + 2z^5 + 2z^6 + 3z^7 + 3z^8 + z^9 \end{aligned}$$

.....

のようにして、任意の高さの木の総数が再帰的に計算される。実際、これらは図 4.1 に対応している。○は 2 分節、⊕は 3 分節、□は葉を表す。

4.3 高さ h の木の総数

2-3 木の母関数の式 (4.2) から容易に計算できるのは、高さ h の木の総数である。高さ h の木の総数は $F_h(1)$ で与えられるから、

$$\begin{cases} F_0(1) = 1 \\ F_h(1) = \{F_{h-1}(1)\}^2 + \{F_{h-1}(1)\}^3 \end{cases} \quad (4.3)$$

より、再帰的に計算される。 $h = 0 \sim 5$ については、表 4.1 のようになり、ほ

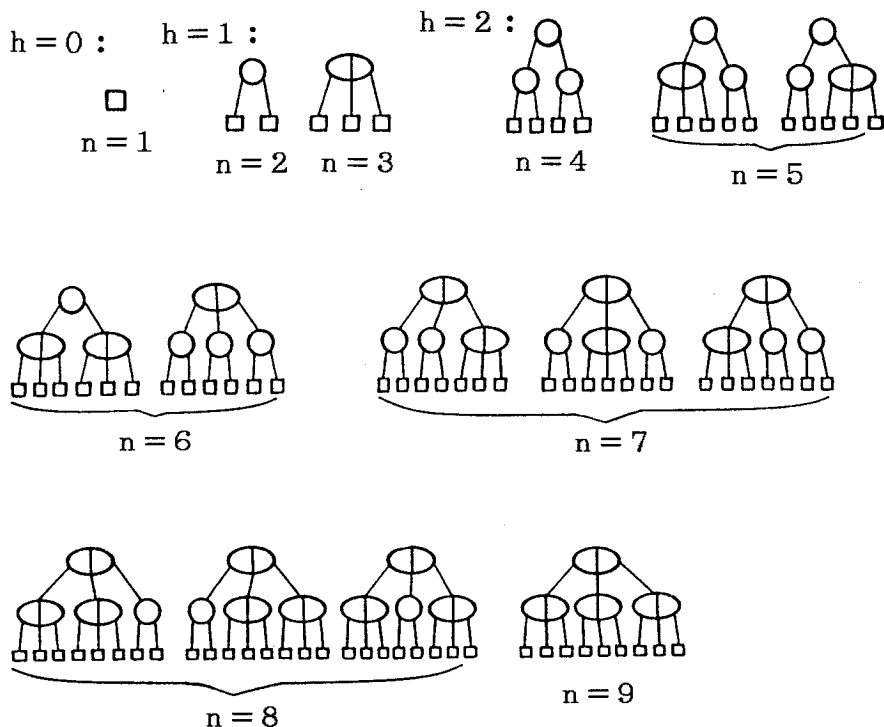


図 4.1 $h=0\sim 2$ にたいする2-3木

表 4.1 高さ h の木の総数

h	$F_h(1)$	β_h
0	1	2
1	2	1.5
2	12	1.083
3	1872	$1 + 5.3419 \dots \times 10^{-4}$
4	6563711232	$1 + 1.5235 \dots \times 10^{-10}$
5	$2.8278 \dots \times 10^{29}$	$1 + 3.5363 \dots \times 10^{-30}$

ほぼ指数関数的に増加する。なお、

$$\beta_h = \frac{F_{h+1}(1)}{\{F_h(1)\}^3} = 1 + \frac{1}{F_h(1)} \quad (4.4)$$

に着目すれば、 h が大きいところでは、ほぼ3乗倍ずつで激増することがわかる。表4.1の第3列には β_h も示した。

4.4 Bruno の公式を用いた再帰的表現法

4.4.1 Bruno の公式

高さ h で n 個の葉を持つ木の総数 $a(h, n)$ を、それよりも高さが1つだけ少ない $a(h-1, m)$ で表現する方法を提案する。これには、Bruno の公式⁽²¹⁾を用いるのが便利であるが、情報工学の分野ではあまり知られていない。そこで、これについて概説する。

まず、整数 n の分割(partition)について述べる。たとえば整数5は、

$$5 = 2 + 3$$

$$5 = 1 + 1 + 1 + 2$$

というように、いくつかの正の整数に分けられる。この分け方は7通りあるが、この数を5の分割数という。これを一般化して整数 n の分割を考えると、それに1が k_1 個、2が k_2 個、 \dots 、 n が k_n 個使われていれば、

$$n = k_1 + 2k_2 + \dots + nk_n \quad (4.5)$$

の関係があるはずである。そして

$$k = k_1 + k_2 + \dots + k_n \quad (4.6)$$

は、 n を何個の整数に分けたかという数を表す。この分け方の総数を $P(n)$ とし、 $P(n)$ 個の分け方の集合を $\pi(n)$ と書く。

つぎに、複合関数(composite function)の導関数について考察する。 $Y(z)$ を

関数 $f(z)$ と $g(z)$ の複合関数とし、

$$Y(z) = f[g(z)] \tag{4.7}$$

と表す。 $Y(z)$, $f(z)$, $g(z)$ の n 次導関数を、それぞれ、

$$Y_n = \left(\frac{d}{dz}\right)^n Y(z), \quad f_n = \left(\frac{d}{du}\right)^n f(u) \Big|_{u=g(z)}, \quad g_n = \left(\frac{d}{dz}\right)^n g(z)$$

とかくとき、 Y_n は

$$Y_n = \sum_{\pi(n)} \frac{n! f_k}{k_1! k_2! \dots k_n!} \left(\frac{g_1}{1!}\right)^{k_1} \left(\frac{g_2}{2!}\right)^{k_2} \dots \left(\frac{g_n}{n!}\right)^{k_n} \tag{4.8}$$

で表される⁽²¹⁾。ここに、総和の下での $\pi(n)$ は、 n の分割の集合を意味する。式 (4.8) は Bruno の公式とよばれる。表 4.2 に $Y(z)$ の導関数 $Y_1 \sim Y_5$ を示す。

4.4.2 ベキ乗の関数の展開形

式(4.2)の右辺が2乗と3乗の項からなることに着目し、一般的なベキ乗の関数を考える。まえと同様にして、

$$Y(z) = f[g(z)] \tag{4.9}$$

と表し、特に、

表 4.2 $Y(z)$ の導関数 $Y_1 \sim Y_5$

$Y_1 = f_1 g_1$
$Y_2 = f_1 g_2 + f_2 g_1^2$
$Y_3 = f_1 g_3 + f_2 (3 g_2 g_1) + f_3 g_1^3$
$Y_4 = f_1 g_4 + f_2 (4 g_3 g_1 + 3 g_2^2)$ $\quad + f_3 (6 g_2 g_1^2) + f_4 g_1^4$
$Y_5 = f_1 g_5 + f_2 (5 g_4 g_1 + 10 g_3 g_2)$ $\quad + f_3 (10 g_3 g_1^2 + 15 g_2^2 g_1)$ $\quad + f_4 (10 g_2 g_1^3) + f_5 g_1^5$

$$f(z) = z^r \quad (4.10)$$

$$g(z) = 1 + b_1 z + b_2 z^2 + \dots \quad (4.11)$$

の場合を考える。すると、

$$\begin{aligned} Y(z) &= (1 + b_1 z + b_2 z^2 + \dots)^r \\ &= 1 + e_1(r) z + e_2(r) z^2 + \dots \end{aligned} \quad (4.12)$$

と表され、 $e_1(r)$ 、 $e_2(r)$ 、 \dots は、 b_1 、 b_2 、 \dots で表現されるはずである。 $e_n(r)$ は、

$$e_n(r) = \frac{1}{n!} \left(\frac{d}{dz} \right)^n Y(z) \Big|_{z=0} \quad (4.13)$$

であるから、

$$g_k \Big|_{z=0} = k! b_k, \quad (1 \leq k) \quad (4.14)$$

$$f_k \Big|_{z=0} \equiv \begin{cases} (r)_k = r(r-1)\dots(r-k+1), & (1 \leq k \leq r) \\ 0, & (k \geq r+1) \end{cases} \quad (4.15)$$

の関係に注意しながら、Bruno の公式を用いると、式(4.8)より、

$$e_n(r) = \frac{1}{n!} \sum_{\pi(n)} \frac{n! (r)_k}{k_1! \dots k_n!} \left(\frac{b_1}{1!} \right)^{k_1} \dots \left(\frac{b_n}{n!} \right)^{k_n} \quad (4.16)$$

$$= \sum_{\pi(n)} \binom{r}{k} \frac{k!}{k_1! \dots k_n!} b_1^{k_1} \dots b_n^{k_n} \quad (4.17)$$

となる。ここに、 k は式(4.6)で定義したものであり、

$$\binom{r}{k} = 0, \quad (k < 0, k > r)$$

とおいている。 $e_1(r) \sim e_5(r)$ は表4.2より次のようになる。

$$1! e_1(r) = (r)_1 d_1 \quad]$$

$$\begin{aligned}
2! e_2(r) &= (r)_1 d_2 + (r)_2 d_1^2 \\
3! e_3(r) &= (r)_1 d_3 + (r)_2 (3 d_2 d_1) + (r)_3 d_1^3 \\
4! e_4(r) &= (r)_1 d_4 + (r)_2 (4 d_3 d_1 + 3 d_2^2) \\
&\quad + (r)_3 (6 d_2 d_1^2) + (r)_4 d_1^4 \\
5! e_5(r) &= (r)_1 d_5 + (r)_2 (5 d_4 d_1 + 10 d_3 d_2) \\
&\quad + (r)_3 (10 d_3 d_1^2 + 15 d_2^2 d_1) \\
&\quad + (r)_4 (10 d_2 d_1^3) + (r)_5 d_1^5
\end{aligned}$$

ただし、表示を簡明にするために $d_k = k! b_k$ とおいた。

4. 4. 3 再帰的表現

前段落の結果を用いて $a(h, n)$ についての再帰的表現を示す。母関数の式

(4. 1)は、

$$\alpha_h = 3^h - 2^h$$

とおけば、式(4. 2)から、

$$\begin{aligned}
F_h(z) &= z^{2^h} \sum_{i=0}^{\alpha_h} a(h, 2^h + i) z^i \\
&= \left\{ z^{2^{h-1}} \sum_{i=0}^{\alpha_{h-1}} a(h-1, 2^{h-1} + i) z^i \right\}^2 \\
&\quad + \left\{ z^{2^{h-1}} \sum_{i=0}^{\alpha_{h-1}} a(h-1, 2^{h-1} + i) z^i \right\}^3 \quad (4. 18)
\end{aligned}$$

となるが、これを

$$\begin{aligned}
&= z^{2^h} \sum_{i=0}^{2\alpha_{h-1}} B(h, i) z^i + z^{3 \cdot 2^{h-1}} \sum_{i=0}^{3\alpha_{h-1}} C(h, i) z^i \\
&= z^{2^h} \sum_{i=0}^{2\alpha_{h-1}} B(h, i) z^i + z^{2^h} \sum_{i=2^{h-1}}^{3\alpha_{h-1} + 2^{h-1}} C(h, i - 2^{h-1}) z^i
\end{aligned}$$

とおく。ここで、

$$\begin{aligned} 3\alpha_{h-1} + 2^{h-1} &= 3(3^{h-1} - 2^{h-1}) + 2^{h-1} \\ &= 3^h - 2^h = \alpha_h \end{aligned}$$

$$\begin{aligned} 2\alpha_{h-1} + 3^{h-1} &= 2(3^{h-1} - 2^{h-1}) + 3^{h-1} \\ &= 3^h - 2^h = \alpha_h \end{aligned}$$

に注意して、

$$B(h, i) \equiv 0, \quad 2\alpha_{h-1} < i \leq \alpha_h$$

$$C(h, i) \equiv 0, \quad i < 0$$

と定義すれば、

$$\begin{aligned} z^{2^h} \sum_{i=0}^{\alpha_h} a(h, 2^h + i) z^i \\ = z^{2^h} \sum_{i=0}^{\alpha_h} \{B(h, i) + C(h, i - 2^{h-1})\} z^i \end{aligned} \quad (4.19)$$

となる。これから、

$$a(h, 2^h + i) = B(h, i) + C(h, i - 2^{h-1}) \quad (4.20)$$

となる。この式の、 $B(h, i)$ と $C(h, i - 2^{h-1})$ を $a(h-1, 2^{h-1} + i)$ で表現するのに Bruno の公式を用いれば、 $a(h, 2^h + i)$ に対する再帰式が得られる。式(4.18)が $F_{h-1}(z)$ の2乗と3乗の和として表されるから、式(4.12)の b_k を

$$b_k = a(h-1, 2^{h-1} + k)$$

とおいたときの2乗と3乗の関数を考えればよい。このとき、 $B(h, i)$ と $C(h, i)$ は、それぞれ $e_i(2)$ と $e_i(3)$ に対応するから、式(4.17)より、

$$\begin{aligned} B(h, i) &= \sum_{\pi(i)} \binom{2}{k} \frac{k!}{k_1! \cdots k_i!} \{a(h-1, 2^{h-1} + 1)\}^{k_1} \\ &\quad \cdots \{a(h-1, 2^{h-1} + i)\}^{k_i} \end{aligned} \quad (4.21)$$

$$C(h, i) = \sum_{\pi(i)} \binom{3}{k} \frac{k!}{k_1! \cdots k_i!} \{a(h-1, 2^{h-1} + 1)\}^{k_1} \cdots \{a(h-1, 2^{h-1} + i)\}^{k_i} \quad (4.22)$$

となる。

なお、2-3木の拡張であるB-木の場合も、式(4.16)を利用すれば、まったく同じ再帰式が得られる。

4.5 二項係数による表現法

4.5.1 再帰表現

高さ h で n 個の葉を持つ木の総数 $a(h, n)$ について、前節で述べた再帰表現とは別の再帰表現を示す。

最初に着目することは、再帰式(4.2)の代わりに、

$$F_h(z) = F_{h-1}(z^2 + z^3) \quad (4.23)$$

も成り立つことである。これは再帰式の最初の部分が、

$$\begin{aligned} F_0(z) &= z \\ F_1(z) &= z^2 + z^3 \\ F_2(z) &= (z^2 + z^3)^2 + (z^2 + z^3)^3 \\ F_3(z) &= \{ (z^2 + z^3)^2 + (z^2 + z^3)^3 \}^2 \\ &\quad + \{ (z^2 + z^3)^2 + (z^2 + z^3)^3 \}^3 \\ &\dots \end{aligned}$$

となることから推察されるように、 $F_{h-1}(z)$ の変数 z に $z^2 + z^3$ を代入しても、 $F_h(z)$ が得られるからである。

式(4.18)は式(4.23)から、

$$F_h(z) = \sum_{i=0}^{\alpha_h} a(h, 2^h + i) z^{2^h + i} \quad (4.24)$$

$$= \sum_{j=0}^{\alpha_{h-1}} a(h-1, 2^{h-1}+j) (z^2+z^3)^{2^{h-1}+j} \quad (4.25)$$

となる。これに二項定理(式(2.5))を用いると、

$$\begin{aligned} F_h(z) &= \sum_{j=0}^{\alpha_{h-1}} a(h-1, 2^{h-1}+j) \sum_{k=0}^{2^{h-1}+j} \binom{2^{h-1}+j}{k} z^{2^h+2j+k} \\ &= \sum_{j=0}^{\alpha_{h-1}} \sum_{k=0}^{2^{h-1}+j} a(h-1, 2^{h-1}+j) \binom{2^{h-1}+j}{k} z^{2^h+2j+k} \end{aligned}$$

となる。いま、簡便のために、

$$f(j, k) = a(h-1, 2^{h-1}+j) \binom{2^{h-1}+j}{k} z^{2^h+2j+k}$$

とおくと、

$$F_h(z) = \sum_{j=0}^{\alpha_{h-1}} \sum_{k=0}^{2^{h-1}+j} f(j, k)$$

である。この総和の領域は図4.2のIのようになるが、ここで技巧的にIIとIIIの領域を考え、

$$S_I = \sum_{j=0}^{\alpha_{h-1}} \sum_{k=0}^{2^{h-1}+j} f(j, k)$$

$$S_{II} = \sum_{j=0}^{\alpha_{h-1}} \sum_{k=2^{h-1}+j+1}^{(2^{h-1}+3\alpha_{h-1})-2j} f(j, k)$$

$$S_{III} = \sum_{j=0}^{\alpha_{h-1}} \sum_{k=-2j}^{-1} f(j, k)$$

とおくと、領域IIでは、

$$2^{h-1}+j < k \text{ より, } \binom{2^{h-1}+j}{k} = 0$$

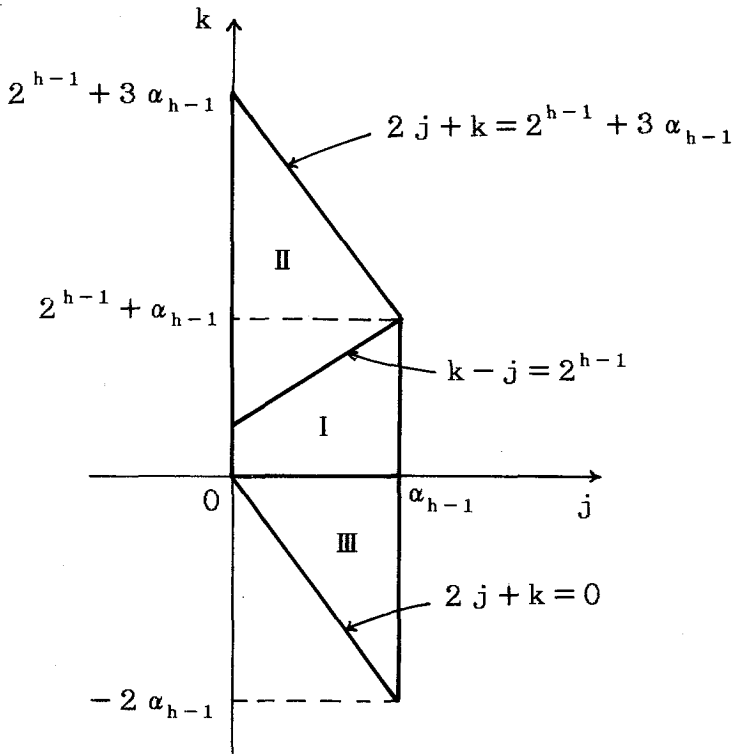


図 4.2 総和の領域

だから、 $f(j, k) = 0$ である。また、領域Ⅲでは、

$$k < 0, \quad 2^{h-1} + j > 0 \text{ より, } \binom{2^{h-1} + j}{k} = 0$$

だから、 $f(j, k) = 0$ である。したがって、

$$F_h(z) = S_I = S_I + S_{II} + S_{III}$$

$$= \sum_{j=0}^{\alpha_{h-1}} \sum_{k=-2j}^{(2^{h-1} + 3\alpha_{h-1}) - 2j} f(j, k)$$

ともかける。ここで、変数 j と k を、

$$\begin{cases} j = j \\ m = 2j + k \end{cases}$$

に変換すると,

$$2^{h-1} + 3\alpha_{h-1} = \alpha_h$$

に注意して,

$$S_I + S_{II} + S_{III} = \sum_{j=0}^{\alpha_{h-1}} \sum_{m=0}^{\alpha_h} f(j, m-2j)$$

となる. すると, 総和はそのまま入れ替えられて,

$$S_I + S_{II} + S_{III} = \sum_{m=0}^{\alpha_h} \sum_{j=0}^{\alpha_{h-1}} f(j, m-2j)$$

となる. この式を具体的にかくと,

$$F_h(z) = \sum_{m=0}^{\alpha_h} \sum_{j=0}^{\alpha_{h-1}} a(h-1, 2^{h-1}+j) \binom{2^{h-1}+j}{m-2j} z^{2^h+m} \quad (4.26)$$

である. この式と式(4.24)の z^{2^h+i} の係数が等しいとおくと,

$$a(h, 2^h+i) = \sum_{j=0}^{\alpha_{h-1}} a(h-1, 2^{h-1}+j) \binom{2^{h-1}+j}{i-2j} \quad (4.27)$$

$$0 \leq i \leq \alpha_h$$

が成り立つ. これは, $a(h, 2^h+i)$ の h についての再帰式であり, 式(4.20)の別の表現となる.

4.5.2 具体的表現

前段落の結果から, $a(h, 2^h+i)$ を二項係数の積和で表現する.

【定理4.1】 $h \geq 1$ のとき, $a(h, 2^h+i)$ は,

$$a(h, 2^h+i) = \sum_{S_{h-1}=0}^{\alpha_{h-1}} \cdots \sum_{S_0=0}^{\alpha_0} \binom{2^0+S_0}{S_1-2S_0} \cdots \binom{2^{h-1}+S_{h-1}}{i-2S_{h-1}} \quad (4.28)$$

$$0 \leq i \leq \alpha_h$$

のように表される. ただし, $a(0, 1) = 1$ である.

(証明) $h=1$ のとき成り立つことを示す. 式(4.28)で $h=1$ とおくと,

$$a(1, 2+i) = \sum_{S_0=0}^{\alpha_0} \binom{2^0+S_0}{i-2S_0} = \sum_{S_0=0}^0 \binom{1+S_0}{i-2S_0} = \binom{1}{i}$$

となる. これから,

$$a(1, 2) = \binom{1}{0} = 1, \quad a(1, 3) = \binom{1}{1} = 1$$

$$a(1, 2+j) = \binom{1}{j} = 0, \quad j \geq 2$$

となり,

$$F_1(z) = z^2 + z^3 = a(1, 2)z^2 + a(1, 3)z^3$$

と一致する.

つぎに, $h=k-1$ のとき成り立つと仮定して, $h=k$ のときに成り立つことを示す. 帰納法の仮定より,

$$a(k-1, 2^{k-1}+i) = \sum_{S_{k-2}=0}^{\alpha_{k-2}} \cdots \sum_{S_0=0}^{\alpha_0} \binom{2^0+S_0}{S_1-2S_0} \cdots \binom{2^{k-2}+S_{k-2}}{i-2S_{k-2}}$$

$$0 \leq i \leq \alpha_{k-1}$$

が成り立つ. そこで, これを式(4.27)に代入すると,

$$a(k, 2^k+i) = \sum_{j=0}^{\alpha_{k-1}} a(k-1, 2^{k-1}+j) \binom{2^{k-1}+j}{i-2j}$$

$$= \sum_{j=0}^{\alpha_{k-1}} \sum_{S_{k-2}=0}^{\alpha_{k-2}} \cdots \sum_{S_0=0}^{\alpha_0} \binom{2^0+S_0}{S_1-2S_0}$$

$$\cdots \binom{2^{k-2}+S_{k-2}}{j-2S_{k-2}} \binom{2^{k-1}+j}{i-2j}$$

$$0 \leq i \leq \alpha_k$$

となる.

(証明終)

表 4.3 二項係数による $a(h, 2^k + i)$, $i = 0 \sim 7$ の表現

$$a(h, 2^k) = 1$$

$$a(h, 2^k + 1) = \binom{2^{k-1}}{1}$$

$$a(h, 2^k + 2) = \binom{2^{k-1}}{2} + \binom{2^{k-2}}{1}$$

$$a(h, 2^k + 3) = \binom{2^{k-1}}{3} + \binom{2^{k-2}}{1} \binom{2^{k-1} + 1}{1}$$

$$a(h, 2^k + 4) = \binom{2^{k-1}}{4} + \binom{2^{k-2}}{1} \binom{2^{k-1} + 1}{2} + \binom{2^{k-2}}{2} + \binom{2^{k-3}}{1}$$

$$a(h, 2^k + 5) = \binom{2^{k-1}}{5} + \binom{2^{k-2}}{1} \binom{2^{k-1} + 1}{3} + \binom{2^{k-2}}{2} \binom{2^{k-1} + 2}{1} \\ + \binom{2^{k-3}}{1} \binom{2^{k-1} + 2}{1}$$

$$a(h, 2^k + 6) = \binom{2^{k-1}}{6} + \binom{2^{k-2}}{1} \binom{2^{k-1} + 1}{4} + \binom{2^{k-2}}{2} \binom{2^{k-1} + 2}{2} \\ + \binom{2^{k-2}}{3} + \binom{2^{k-3}}{1} \binom{2^{k-1} + 2}{2} + \binom{2^{k-3}}{1} \binom{2^{k-2} + 1}{1}$$

$$a(h, 2^k + 7) = \binom{2^{k-1}}{7} + \binom{2^{k-2}}{1} \binom{2^{k-1} + 1}{5} + \binom{2^{k-2}}{2} \binom{2^{k-1} + 2}{3} \\ + \binom{2^{k-2}}{3} \binom{2^{k-1} + 3}{1} + \binom{2^{k-3}}{1} \binom{2^{k-1} + 2}{3} \\ + \binom{2^{k-3}}{1} \binom{2^{k-2} + 1}{1} \binom{2^{k-1} + 3}{1}$$

式(4.28)から、 $i = 0, 1, \dots, 7$ に対する $a(h, 2^h + i)$ は表4.3のようになる。特に、 $2^h > i$ のとき、

$$\lambda_h = 2^{h-1}$$

として、 $a(h, 2^h + i)$ を λ_h の多項式に展開すると、表4.4のようになる。また、 $h = 1 \sim 8$ について計算すると、高さ h の木の総数として表4.5の数値が得られる。

表 4.4 $a(h, 2^h + i)$, $i = 0 \sim 7$ の式($2^h > i$)

$$a(h, 2^h) = 1$$

$$a(h, 2^h + 1) = \lambda_h$$

$$a(h, 2^h + 2) = \frac{\lambda_h^2}{2!}$$

$$a(h, 2^h + 3) = \frac{\lambda_h^3}{3!} + \frac{5}{6} \lambda_h$$

$$a(h, 2^h + 4) = \frac{\lambda_h^4}{4!} + \frac{5}{6} \lambda_h^2 - \frac{\lambda_h}{4}$$

$$a(h, 2^h + 5) = \frac{\lambda_h^5}{5!} + \frac{5}{12} \lambda_h^3 - \frac{\lambda_h^2}{4} + \frac{\lambda_h}{5}$$

$$a(h, 2^h + 6) = \frac{\lambda_h^6}{6!} + \frac{5}{36} \lambda_h^4 - \frac{\lambda_h^3}{8} + \frac{197}{360} \lambda_h^2 + \frac{\lambda_h}{4}$$

$$a(h, 2^h + 7) = \frac{\lambda_h^7}{7!} + \frac{5}{144} \lambda_h^5 - \frac{\lambda_h^4}{24} + \frac{1127}{2520} \lambda_h^3 + \frac{\lambda_h^2}{24} + \frac{39}{28} \lambda_h$$

表 4.5 木の総数 $a(h, 2^h+i)$

$h \backslash i$	0	1	2	3	4	5	6	7
1	1	1	0	0	0	0	0	0
2	1	2	2	3	1	0	0	0
3	1	4	8	14	32	43	63	63
4	1	8	32	92	222	472	906	1626
5	1	16	128	696	2940	10384	32036	88804
6	1	32	512	5488	44536	293024	1633416	7953544
7	1	64	3048	43744	702448	9056064	97743376	4531110928
8	1	128	8192	349632	11198432	287200896	6145427488	112879219744

4.6 準完全2分木に対する総数の近似式

ここでは、前節の結果を用いて、準完全2分木に対する $a(h, n)$ の近似式を導く。準完全2分木とは、葉の数 n が $r_h \ll 2^h$ なる r_h に対して $2^h \leq n \leq 2^h + r_h$ であるような高さ h の木である。表4.4の多項式から推察すると、 $2^h \gg r_h$ なる r_h に対し、 $0 \leq i \leq r_h$ のとき、

$$a(h, 2^h + i) = \begin{cases} \frac{\lambda_h^i}{i!} & : i = 0, 1, 2 \\ \frac{\lambda_h^i}{i!} + \frac{5}{6} \frac{\lambda_h^{i-2}}{(i-3)!} + O(\lambda_h^{i-3}) & : 3 \leq i \leq r_h \end{cases} \quad (4.29)$$

が予想される。そこで、実際に式(4.29)が成り立つことを帰納法により証明する。

まず、 $h=1$ のときは $\lambda_1=1$ となるから、式(4.29)は、

$$a(1, 2+i) = \frac{1^i}{i!}$$

となる。これから、

$$a(1, 2) = \frac{1^0}{0!} = 1, \quad a(1, 3) = \frac{1^1}{1!} = 1$$

となり、

$$F_1(z) = z^2 + z^3 = a(1, 2)z^2 + a(1, 3)z^3$$

と一致する。

つぎに、 $h=k-1$ のとき成り立つとして、 $h=k$ のとき成り立つことを示す。

帰納法の仮定より、 $2^{k-1} \gg r_{k-1}$ なる r_{k-1} に対して

$$a(k-1, 2^{k-1} + i) = \begin{cases} \frac{\{2^{(k-1)-1}\}^i}{i!} & : i = 0, 1, 2 \\ \frac{\{2^{(k-1)-1}\}^i}{i!} + \frac{5}{6} \frac{\{2^{(k-1)-1}\}^{i-2}}{(i-3)!} \\ + O(\{2^{(k-1)-1}\}^{i-3}) & : 3 \leq i \leq r_{k-1} \end{cases}$$

$$= \begin{cases} \frac{(2^{k-1})^i}{2^i \cdot i!} & : i = 0, 1, 2 \\ \frac{(2^{k-1})^i}{2^i \cdot i!} + \frac{5}{6} \frac{(2^{k-1})^{i-2}}{2^{i-2} \cdot (i-3)!} + O\left(\frac{(2^{k-1})^{i-3}}{2^{i-3}}\right) \\ & : 3 \leq i \leq r_{k-1} \end{cases}$$

が成り立つ。そこで、 $\lambda_k = 2^{k-1}$ として式(4.27)に代入する。 $r_{k-1} < j \leq a_{k-1}$ のとき、

$$a(k-1, 2^{k-1} + j) = 0$$

として、 $2^k \gg r_k \geq i \geq 0$ のとき、

$$\begin{aligned} a(k, 2^k + i) &= \sum_{j=0}^{r_{k-1}} a(k-1, 2^{k-1} + j) \binom{2^{k-1} + j}{i-2j} \\ &= \sum_{j=0}^2 \frac{\lambda_k^j}{2^j \cdot j!} \binom{\lambda_k + j}{i-2j} + \sum_{j=3}^{r_{k-1}} \left\{ \frac{\lambda_k^j}{2^j \cdot j!} \right. \\ &\quad \left. + \frac{5}{6} \frac{\lambda_k^{j-2}}{2^{j-2} \cdot (j-3)!} + O\left(\frac{\lambda_k^{j-3}}{2^{j-3}}\right) \right\} \binom{\lambda_k + j}{i-2j} \\ &= \sum_{j=0}^{r_{k-1}} \frac{\lambda_k^j}{2^j \cdot j!} \binom{\lambda_k + j}{i-2j} \end{aligned}$$

$$\begin{aligned}
& + \frac{5}{6} \sum_{j=3}^{r_k-1} \frac{\lambda_k^{j-2}}{2^{j-2} \cdot (j-3)!} \binom{\lambda_k+j}{i-2j} + \sum_{j=3}^{r_k-1} O\left(\left(\frac{\lambda_k}{2}\right)^{j-3}\right) \binom{\lambda_k+j}{i-2j} \\
& = \text{I} + \text{II} + \text{III}
\end{aligned}$$

となる。ここで、 $\lambda_k \gg j$ を利用すると、

$$\begin{aligned}
\frac{\lambda_k^j}{2^j \cdot j!} \binom{\lambda_k+j}{i-2j} &= \frac{\lambda_k^j}{2^j \cdot j!} \{ \lambda_k^{i-2j} + O(\lambda_k^{i-2j-1}) \} \\
&= O(\lambda_k^{i-j})
\end{aligned}$$

となるから、

$$\text{I} = O(\lambda_k^i)$$

$$\text{II} = O(\lambda_k^{i-5})$$

$$\text{III} = O(\lambda_k^{i-6})$$

であることがわかる。よって、式(4.29)を求めるには、Iの展開だけを考えればよい。そこで、

$$\text{I} = \binom{\lambda_k}{i} + \frac{\lambda_k}{2} \binom{\lambda_k+1}{i-2} + \frac{\lambda_k^2}{8} \binom{\lambda_k+2}{i-4} + O(\lambda_k^{i-3})$$

としてこの右辺を展開する。第1種の Stirling 数(式(2.6))を用いると第1項～3項はそれぞれ次のようになる。

$$\begin{aligned}
\binom{\lambda_k}{i} &= \frac{1}{i!} \left\{ \left[\begin{matrix} i \\ i \end{matrix} \right] \lambda_k^i - \left[\begin{matrix} i \\ i-1 \end{matrix} \right] \lambda_k^{i-1} + \left[\begin{matrix} i \\ i-2 \end{matrix} \right] \lambda_k^{i-2} + O(\lambda_k^{i-3}) \right\} \\
&= \frac{\lambda_k^i}{i!} - \frac{\lambda_k^{i-1}}{2 \cdot (i-2)!} + \left\{ \frac{i(i-1)}{8 \cdot (i-2)!} - \frac{2i-1}{12 \cdot (i-2)!} \right\} \\
&\quad \cdot \lambda_k^{i-2} + O(\lambda_k^{i-3})
\end{aligned}$$

$$\begin{aligned} \frac{\lambda_k}{2} \binom{\lambda_k+1}{i-2} &= \frac{\lambda_k}{2 \cdot (i-2)!} \left\{ \begin{bmatrix} i-2 \\ i-2 \end{bmatrix} \lambda_k^{i-2} - \begin{bmatrix} i-2 \\ i-3 \end{bmatrix} \lambda_k^{i-3} \right. \\ &\quad \left. + O(\lambda_k^{i-4}) \right\} \\ &= \frac{\lambda_k^{i-1}}{2 \cdot (i-2)!} - \frac{1}{2 \cdot (i-2)!} \left\{ \frac{(i-3)(i-4)}{2} \right. \\ &\quad \left. - 1 \right\} \lambda_k^{i-2} + O(\lambda_k^{i-3}) \end{aligned}$$

$$\begin{aligned} \frac{\lambda_k^2}{8} \binom{\lambda_k+2}{i-4} &= \frac{\lambda_k^2}{8 \cdot (i-4)!} \left\{ \begin{bmatrix} i-4 \\ i-4 \end{bmatrix} \lambda_k^{i-4} + O(\lambda_k^{i-5}) \right\} \\ &= \frac{\lambda_k^{i-2}}{8 \cdot (i-4)!} + O(\lambda_k^{i-3}) \end{aligned}$$

よって,

$$\begin{aligned} a(k, 2^k + i) &= I + II + III \\ &= \frac{\lambda_k^i}{i!} + \left\{ -\frac{1}{2 \cdot (i-2)!} + \frac{1}{2 \cdot (i-2)!} \right\} \lambda_k^{i-1} \\ &\quad + \left\{ \frac{i(i-1)}{8 \cdot (i-2)!} - \frac{2i-1}{12 \cdot (i-2)!} - \frac{(i-4)(i-3)-2}{4 \cdot (i-2)!} \right. \\ &\quad \left. + \frac{1}{8 \cdot (i-4)!} \right\} \lambda_k^{i-2} + O(\lambda_k^{i-3}) \\ &= \frac{\lambda_k^i}{i!} + \frac{5}{6} \frac{\lambda_k^{i-2}}{(i-3)!} + O(\lambda_k^{i-3}) \end{aligned}$$

となる.

(証明終)

4.7 結 言

高さ h で n 個の葉を持つ 2-3 木の正確な総数を表現する方法について考察した。Bruno の公式を用いた再帰的な表現(式(4.20))と、二項係数による再帰的および具体的な表現(式(4.27)と(4.28))とを明らかにした。また、準完全 2 分木に対する $a(h, n)$ の近似式(4.29)を求めた。

本節に関連して、2-3 木の拡張である B-木の数え上げ問題がある。 m 次の B-木を表現する母関数は、

$$F_h(z) = \{ F_{h-1}(z) \}^{\lceil \frac{m+1}{2} \rceil} + \dots + \{ F_{h-1}(z) \}^m$$

で表されるから、高さ h で n 個の葉を持つ木の総数は、2-3 木の場合と同様に Bruno の公式を用いて再帰的に表現できる。これから式(4.28)に対応するような具体的な式を求めることが研究問題として残されている。

第5章 比較コストに関する最適2-3木

5.1 緒言

2-3木の最適構成問題については Rosenberg などの研究がある。^{(19),(23),(24)} 各種の性能評価の尺度にたいして、最適木を特徴づけ、その構成法を示している。文献(19)は平均訪問節数(node-visit-cost)に関するものである。これは、節内の探索時間を無視した能率評価であるが、このときの最適木は領域利用率がかなり悪いので、文献(24)では領域利用率が最大の木を考えている。そしてその木が平均訪問節数においても最適に近いことを示している。文献(23)は、見出しの比較回数に関するコスト(key-comparison-cost)を扱ったもので、節訪問時間は無視している。このときの比較は、1度に3とおりの比較結果($<$, $=$, $>$)が得られる3-結果比較器(3-outcome-comparator)を用いると仮定しているが、今後の研究課題として「1度に2とおりの比較結果(\leq , $>$)しか得られない場合、つまり、2-結果比較器(2-outcome-comparator)を用いた場合の考察」を与えている。本節ではこの問題を扱う。実際に、計算機では1度に2とおりの比較結果しか得られないので、2-結果比較器を用いた場合の見出しの比較回数の考察も意味があると考えられる。

5.2では、2-結果比較器を用いたときの探索アルゴリズムを示し、比較コストを定義する。5.3では、その探索アルゴリズムによる比較コストを調べるために、2-3木の2分木変換(binanzation)を考え、2分木(binary tree)が比較コスト最小となる場合を特徴づける。5.4では、比較コスト最小2分木の特徴に基づいて、比較コスト最小2-3木を特徴づけ、5.5で、その構成アルゴリズムを示す。

5.2 コストの定義

2-結果比較器を用いたときの探索アルゴリズムを定め、比較コストを定義する。まず、2-3木に対する記法の定義を行う。

【定義5.1】

- (1) 2-3木の節の集合を N とし、 N_0, N_2, N_3 をそれぞれ葉、2分節、3分節の集合とする。
- (2) 議論を簡単にするために、2-3木のすべての節に対して、次のように定義されるアルファベット $\{\lambda, \mu, \rho, l, r\}$ 上の記号列を対応づける。
 - (i) 根を Λ とする。
 - (ii) 記号列 x に対応する2分節の左子節(left son)を $x l$ とし、右子節(right son)を $x r$ とする。記号列 x に対応する3分節の左子節を $x \lambda$ 、中子節(middle son)を $x \mu$ 、右子節を $x \rho$ とする。
- (3) 3分節 x に対して、 $x[1]$ は小さい方の見出しを表し、 $x[2]$ は大きい方の見出しを表す。

7個の葉を持つ2-3木の例を図5.1に示し、2-3木の探索における見出しの比較を図5.2に示す。図5.2(a)は、文献(23)で仮定した3-結果比較器を用いた場合であり、(b)は、本節で扱う2-結果比較器を用いた場合である。

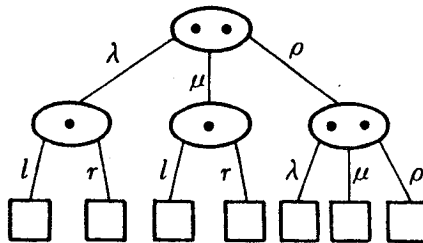
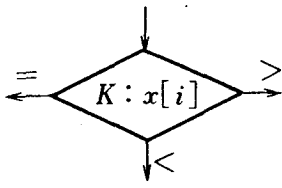
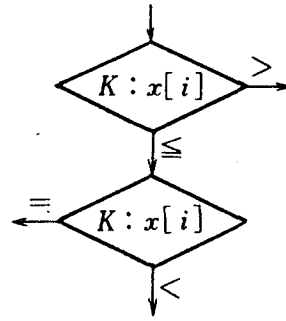


図 5.1 2-3木の例



(a) 3-結果比較器



(b) 2-結果比較器

図 5.2 見出しの比較($i = 1, 2$)

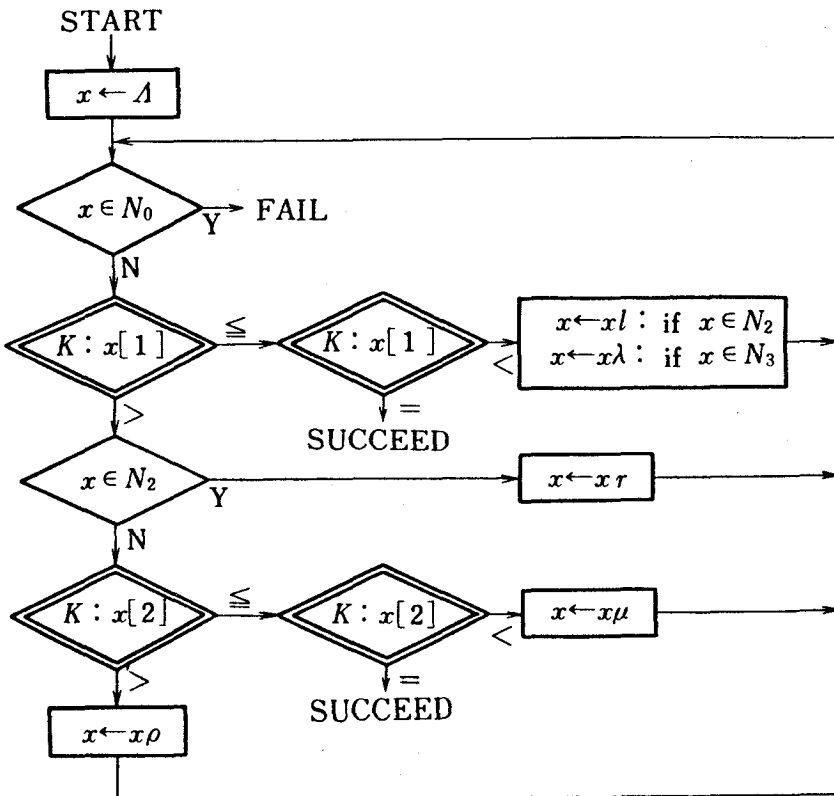


図 5.3 探索アルゴリズム A^*

2-結果比較器を用いたときの2-3木の探索アルゴリズム A^* を図5.3に示す。

図の4箇所にある2重のひし形が見出し同士の比較を表す。

アルゴリズム A^* を用いて2-3木の探索を行うときの比較コストを定義するために、関数 place を次のように定義する。

【定義5.2】 関数 place は、次の(a), (b)で定義される。

(a) $\text{place}(A[1]) = 2$

(b) $x \in N_2 + N_3$ に対して

$x \in N_2$ のとき

$$\text{place}(x \ell [1]) = \text{place}(x[1]) + 2$$

$$\text{place}(x r [1]) = \text{place}(x[1]) + 1$$

$x \in N_3$ のとき

$$\text{place}(x \lambda [1]) = \text{place}(x[1]) + 2$$

$$\text{place}(x [2]) = \text{place}(x[1]) + 1$$

$$\text{place}(x \mu [1]) = \text{place}(x[2]) + 2$$

$$\text{place}(x \rho [1]) = \text{place}(x[2]) + 1$$

$\text{place}(x[i])$ は、アルゴリズム A^* を用いて、見出し $x[i]$ をアクセスするのに必要な比較回数を表す。

【定義5.3】 節の集合が、 $N = N_0 + N_2 + N_3$ の2-3木 T の比較コストは、次式で表される。

$$C(T) = \sum_{x \in N_2 + N_3} \text{place}(x[1]) + \sum_{x \in N_3} \text{place}(x[2])$$

5.3 フィボナッチ2分木

前節で定めた探索アルゴリズムで2-3木を探索したときの比較コストを調べるために、2-3木の2分木変換(binartization)を考え、2分木が比較コスト最

小となる場合を特徴づける。

【定義 5.4】⁽²³⁾ 2-3木の2分木変換 β は、図5.4で示される。3分節は2個の2分節に分けられ、大きい方の見出しを含む節は、小さい方の見出しを含む節の右子節となる。

【補題 5.1】⁽²³⁾ すべての2-3木 T に対して変換 β は定義され、 $\beta(T)$ は T と同じ比較コストを持つ。

任意の2分木に対して、関数 $place$ で各節のコストを定義した場合の比較コスト最小2分木については、文献(25)に示されている。次の補題5.2と補題5.3は、文献(25)による。

【補題 5.2】⁽²⁵⁾ 任意の2分木において、任意の内部節のコスト i は次式で示される。今、根からその節までの路上の右枝(right edge)の個数を K_r 、左枝(left edge)の個数を K_l とすると、

$$i = 2 + K_r + 2K_l.$$

このとき、同じコスト i を持つ節の個数は、たかだか F_{i-1} 個である。ここで、 F_i はフィボナッチ数である。

2分木の各節のコストを図5.5に示す。補題5.2より、次の補題5.3が成り立つ。

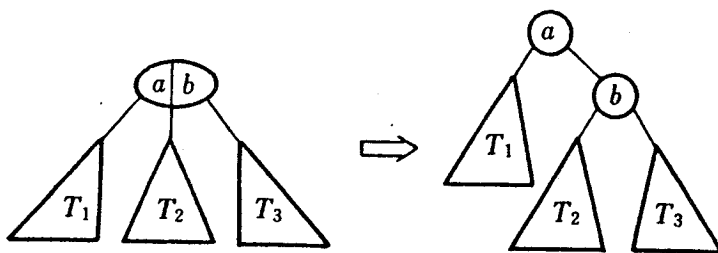


図 5.4 2分木変換の説明

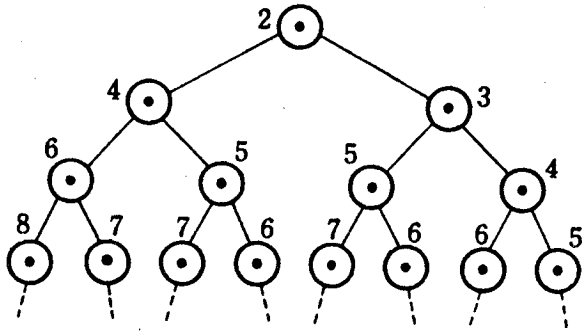


図 5.5 2分木の各節のコスト

【補題5.3】⁽²⁵⁾ n 個の葉を持つ比較コスト最小2分木 T_n は次の性質を持つ。

今, $F_i < n \leq F_{i+1} (i \geq 2)$ とすると,

- (1) T_n は, コスト $j (j = 2, \dots, i - 1)$ の節を, F_{j-1} 個, コスト i の節を $(n - F_i)$ 個持つ。
- (2) T_n の左部分木(left subtree)は, コスト $j (j = 4, \dots, i - 1)$ の節を F_{j-3} 個, コスト i の節を, $0 \leq n_l \leq F_{i-3}$ 個含む。 T_n の右部分木(right subtree)は, コスト $j (j = 3, \dots, i - 1)$ の節を F_{j-2} 個, コスト i の節を $0 \leq n_r \leq F_{i-2}$ 個含む。ここで, $n - F_i = n_l + n_r$ である。
- (3) T_n のコスト $C(T_n)$ は, 次式で示される。

$$C(T_n) = \sum_{j=2}^{i-1} F_{j-1} \cdot j + i(n - F_i)$$

以下で, 比較コスト最小2分木を特徴づけるために, フィボナッチ2分木を定義する。これは, 2章で述べたフィボナッチ木を, 任意の n の場合に拡張したものである。

【定義5.5】 n 個の葉を持つフィボナッチ2分木 T_n^F を, 次のように定義し, そのクラスを \mathcal{F}_n とする。

(1) $n = 1, 2$ のとき, T_1^F, T_2^F を図 5.6 (a) で示す.

(2) $F_i < n \leq F_{i+1}$ のとき, T_n^F の左, 右部分木をそれぞれ T_{n_1}, T_{n_2} とする (図 5.6 (b)). ここで, $n = n_1 + n_2$ なる n_1, n_2 に対して,

$$F_{i-2} \leq n_1 \leq F_{i-1}, F_{i-1} \leq n_2 \leq F_i$$

$$T_{n_1} \in \mathcal{T}_{n_1}, T_{n_2} \in \mathcal{T}_{n_2}$$

である.

【補題 5.4】 n 個の葉を持つ 2 分木 T_n は, $T_n \in \mathcal{T}_n$ なるとき, 同じ数の葉を持つ 2 分木のうちで比較コスト最小であり, 又, そのときに限る.

(証明) 十分性: n についての帰納法により, $T_n \in \mathcal{T}_n$ なる $T_n (F_i < n \leq F_{i+1})$ のコストが,

$$C(T_n) = \sum_{j=2}^{i-1} F_{j-1} \cdot j + i(n - F_i) \quad (5.1)$$

となることを示す.

(i) $n \leq F_4$ のとき明らかに成り立つ.

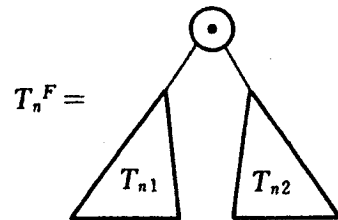
(ii) $n \leq F_i$ のとき成り立つと仮定して, $F_i < n \leq F_{i+1}$ について示す.

T_n の左部分木を T_{n_1} , 右部分木を T_{n_2} とすると, フィボナッチ 2 分木の定義より, $n = n_1 + n_2, F_{i-2} \leq n_1 \leq F_{i-1}, F_{i-1} \leq n_2 \leq F_i$ である. 帰納法の仮定お

$$T_1^F = \square, \quad \mathcal{T}_1 = \{ \square \} \quad (\square \text{ は 1 個の葉を表す})$$

$$T_2^F = \begin{array}{c} \circ \\ / \quad \backslash \\ \square \quad \square \end{array}, \quad \mathcal{T}_2 = \left\{ \begin{array}{c} \circ \\ / \quad \backslash \\ \square \quad \square \end{array} \right\}$$

(a)



(b)

図 5.6 フィボナッチ 2 分木

よび2分木におけるコストの定義より,

$n_1 = F_{i-2}$ のとき,

$$\begin{aligned} C(T_{n_1}) &= \sum_{j=2}^{i-4} F_{j-1} \cdot (j+2) + (i-1)(n_1 - F_{i-3}) \\ &= \sum_{j=2}^{i-3} F_{j-1} \cdot (j+2) \end{aligned} \quad (5.2)$$

$F_{i-2} < n_1 \leq F_{i-1}$ のとき,

$$C(T_{n_1}) = \sum_{j=2}^{i-3} F_{j-1} \cdot (j+2) + i(n_1 - F_{i-2}) \quad (5.3)$$

$n_2 = F_{i-1}$ のとき,

$$\begin{aligned} C(T_{n_2}) &= \sum_{j=2}^{i-3} F_{j-1} \cdot (j+1) + (i-1)(n_2 - F_{i-2}) \\ &= \sum_{j=2}^{i-2} F_{j-1} \cdot (j+1) \end{aligned} \quad (5.4)$$

$F_{i-1} < n_2 \leq F_i$ のとき,

$$C(T_{n_2}) = \sum_{j=2}^{i-2} F_{j-1} \cdot (j+1) + i(n_2 - F_{i-1}) \quad (5.5)$$

である.

$n_1 = F_{i-2}$ のとき, $F_{i-1} < n_2 \leq F_i$ となり, T_n のコストは, 式(5.2)と

(5.5)より,

$$\begin{aligned} C(T_n) &= 2 + \sum_{j=2}^{i-3} F_{j-1} \cdot (j+2) + \sum_{j=2}^{i-2} F_{j-1} \cdot (j+1) + i(n_2 - F_{i-1}) \\ &= 2 + 3F_1 + \sum_{j=2}^{i-3} F_{j+1} \cdot (j+2) + i(n_2 - F_{i-1}) \end{aligned}$$

ここで、 $n_2 - F_{i-1} = n_2 + n_1 - (F_{i-1} + F_{i-2})$ より、

$$= 2 + 3F_1 + \sum_{j=2}^{i-3} F_{j+1} \cdot (j+2) + i(n - F_i)$$

となり、 $F_1 = F_2 = 1$ に注意すると、式(5.1)が得られる。同様に、 $n_2 = F_{i-1}$ のときは、式(5.3)、(5.4)より式(5.1)が得られ、 $F_{i-2} < n_1 \leq F_{i-1}$ 、 $F_{i-1} < n_2 \leq F_i$ のときは、式(5.3)、(5.5)より式(5.1)が得られる。よって補題5.3より比較コスト最小である。

必要性： T_n が比較コスト最小のとき、 $T_n \in \mathcal{T}_n$ なることを、 n についての帰納法により示す。

(i) $n \leq F_4$ のとき明らかに成り立っている。

(ii) $n \leq F_i$ のとき成り立つと仮定する。 $F_i < n \leq F_{i+1}$ のとき、 T_n の左部分木を T_{n_1} 、右部分木を T_{n_2} とし、 T_{n_1} のすべての節のコストから2を引いたものを T'_{n_1} 、 T_{n_2} のすべての節のコストから1を引いたものを T'_{n_2} とする。このとき、 T_n がコスト最小であることより、 T'_{n_1} 、 T'_{n_2} は、コスト最小である。

一方、 T_n がコスト最小であることより、補題5.3を用いれば、 T_{n_1} の葉の数 n_1 は、

$$F_{i-2} \leq n_1 = \sum_{j=4}^{i-1} F_{j-3} + n_2 + 1 \leq F_{i-1}$$

T_{n_2} の葉の数 n_2 は、

$$F_{i-1} \leq n_2 = \sum_{j=3}^{i-1} F_{j-2} + n_1 + 1 \leq F_i$$

となることが示される。ここで、 $n - F_i = n_2 + n_1$ である。

従って、帰納法の仮定より、 $T'_{n_1} \in \mathcal{T}_{n_1}$ 、 $T'_{n_2} \in \mathcal{T}_{n_2}$ であるから、 $T_n \in \mathcal{T}_n$ である。 (証明終)

5.4 比較コスト最小2-3木

2分木が比較コスト最小となる場合の条件より、2-結果比較器を用いたときの比較コスト最小2-3木を特徴づけ、そのコストを示す。

【定理5.1】 n 個の葉を持つ2-3木 T_n は、次の性質Mを満たすとき、同じ数の葉を持つ2-3木のうちで比較コスト最小であり、又、そのときに限る。

(性質M) T_n の任意の内部節を根とする部分木は、次の条件式を満足する。今、部分木の葉の数を N_l とし、その部分木の2分左、2分右、3分左、3分中、3分右部分木の葉の数を、それぞれ $N_l, N_r, N_\lambda, N_\mu, N_\rho$ とすると、 $F_i < N_l \leq F_{i+1}$ のとき、

部分木の根が2分節であれば、

$$(F_{i-2} \leq N_l \leq F_{i-1}) \text{ 且つ } (F_{i-1} \leq N_r \leq F_i)$$

部分木の根が3分節であれば、

$$(F_{i-2} \leq N_\lambda \leq F_{i-1}) \text{ 且つ } (F_{i-3} \leq N_\mu \leq F_{i-2}) \text{ 且つ } (F_{i-2} \leq N_\rho \leq F_{i-1})$$

である。

12個の葉を持つ比較コスト最小2-3木の例を図5.7に示す。節内の数は、各見出しのコストを示す。

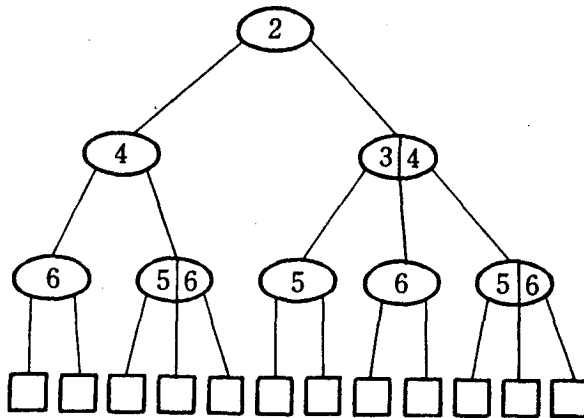


図 5.7 比較コスト最小2-3木の例

定理 5.1 を証明するのに、次の補題 5.5 と 5.6 が必要である。

【補題 5.5】 n 個の葉を持つ 2-3 木 T_n は、性質 M を満たすとき、 $\beta(T_n) \in \mathcal{T}_n$ であり、又、そのときに限る。

(証明) n についての帰納法により証明する。

[I] $n \leq F_4$ のとき、明らかに成り立つ(図 5.8)。

[II] $n \leq F_i$ のとき成り立つと仮定し、 $F_i < n \leq F_{i+1}$ の場合について示す。

(i) T_n の根が 2 分節のとき

十分性: T_n が性質 M を満たすことより、左、右部分木の葉の数をそれぞれ n_1 , n_2 とすると $n = n_1 + n_2$, $F_{i-2} \leq n_1 \leq F_{i-1}$, $F_{i-1} \leq n_2 \leq F_i$ である。ここで、 T_n の左、右部分木 T_{n_1} と T_{n_2} は性質 M を満足しているから、帰納法の仮定より、 $\beta(T_{n_1}) \in \mathcal{T}_{n_1}$, $\beta(T_{n_2}) \in \mathcal{T}_{n_2}$ である。故に、定義 5.5 より、 $\beta(T_n) \in \mathcal{T}_n$ である。

必要性: $\beta(T_n) \in \mathcal{T}_n$ のとき、定義 5.5 より、 $\beta(T_n)$ の左、右部分木をそれぞれ T_{n_1}, T_{n_2} とすると、 $T_{n_1} \in \mathcal{T}_{n_1}$, $T_{n_2} \in \mathcal{T}_{n_2}$ である。ここで、 $n = n_1 + n_2$, $F_{i-2} \leq n_1 \leq F_{i-1}$, $F_{i-1} \leq n_2 \leq F_i$ である。このとき、帰納法の仮定よ

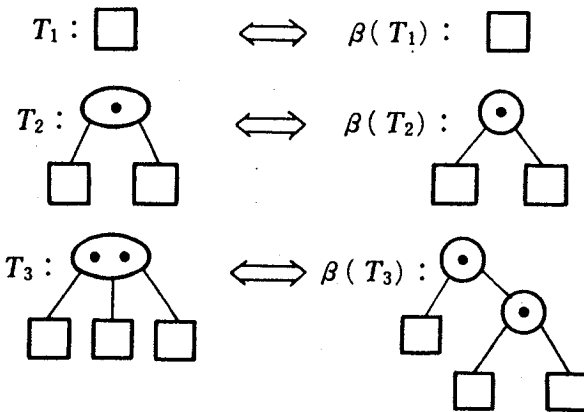


図 5.8 補題 5.5 の説明 [I]

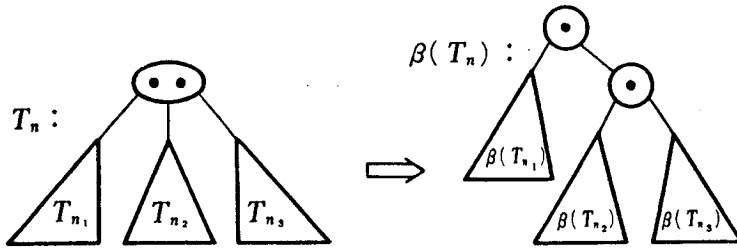


図 5.9 補題 5.5 の説明【II】

り, $\beta^{-1}(T_{n_1})$ と $\beta^{-1}(T_{n_2})$ は, 性質 M を満足する. 故に, T_n は性質 M を満足する. ここで, β^{-1} は, 2分木変換 β の逆写像であり, $\beta(\beta^{-1}(T)) = T$ とする.

(ii) T_n の根が 3 分節のとき

十分性: T_n が性質 M を満足することより, 左, 中, 右部分木の葉の数を, それぞれ n_1, n_2, n_3 とすると, $n = n_1 + n_2 + n_3$, $F_{i-2} \leq n_1 \leq F_{i-1}$, $F_{i-3} \leq n_2 \leq F_{i-2}$, $F_{i-2} \leq n_3 \leq F_{i-1}$ である. ここで, T_n の左, 中, 右部分木 $T_{n_1}, T_{n_2}, T_{n_3}$ は, 性質 M を満足しているから, 帰納法の仮定より, $\beta(T_{n_1}) \in \mathcal{T}_{n_1}$, $\beta(T_{n_2}) \in \mathcal{T}_{n_2}$, $\beta(T_{n_3}) \in \mathcal{T}_{n_3}$ である(図 5.9). このとき, $F_{i-1} \leq n_2 + n_3 \leq F_i$ であり, 且つ, $\beta(T_{n_2}), \beta(T_{n_3})$ をそれぞれ, 左, 右部分木とする $\beta(T_n)$ の部分木が, $\mathcal{T}_{n_2+n_3}$ に含まれることに注意すると, 定義 5.5 より $\beta(T_n) \in \mathcal{T}_n$ である.

必要性: $\beta(T_n)$ を図 5.9 のように分解して, T_n の根が 2 分節のときの必要性と同様に証明される. (証明終)

【補題 5.6】 任意の n に対して, $\beta(T_n) \in \mathcal{T}_n$ を満たす 2-3 木が存在する.

(証明) T_n を帰納的に構成する.

【I】 $n \leq F_4$ のとき T_1, T_2, T_3 を図 5.10(a) で示す.

[II] $F_i < n = F_i + r \leq F_{i+1}$ のとき

(1) i が偶数のとき

根を2分節とし、左、右部分木をそれぞれ T_{n_1} , T_{n_2} とする(図5.10(b)).

ここで、

(i) $1 \leq r \leq F_{i-2} + 1$ のとき $n_1 = F_{i-2} + 1$, $n_2 = F_{i-1} + r - 1$

(ii) $F_{i-2} + 2 \leq r \leq F_{i-1}$ のとき $n_1 = r$, $n_2 = F_i$

とする。

(2) i が奇数のとき

根を3分節とし、左、中、右部分木を、それぞれ T_{n_1} , T_{n_2} , T_{n_3} とする

(図5.10(c)). ここで、

(i) $1 \leq r \leq F_{i-3} + 1$ のとき $n_1 = F_{i-2}$, $n_2 = F_{i-3} + 1$,

$$n_3 = F_{i-2} + r - 1$$

(ii) $F_{i-3} + 2 \leq r \leq F_{i-2}$ のとき $n_1 = F_{i-2}$, $n_2 = r$, $n_3 = F_{i-1}$

(iii) $F_{i-2} + 1 \leq r \leq F_{i-1}$ のとき $n_1 = r$, $n_2 = F_{i-2}$, $n_3 = F_{i-1}$

とする。

上記の構成法において、 $F_{2i} < n \leq F_{2i+2}$ なる n に対する T_n の高さは i であることが、以下のように示される。

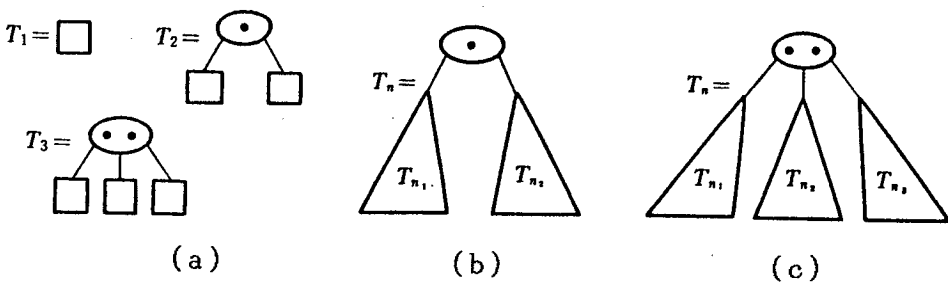


図 5.10 2-3木の構成

(1) $i = 2$ のとき, $F_4 < n < F_6$ なる T_n の高さは $h = 2$ となり, 成り立っている.

(2) $F_{2i-2} < n \leq F_{2i}$ なる n に対して, $h = i - 1$ が成り立っていると仮定する.
 $F_{2i} < n \leq F_{2i-2}$ なる n に対して, $h = i$ が成り立つことを示す.

(i) $F_{2i} < n \leq F_{2i+1}$ のとき, T_n は, T_{n_1} , T_{n_2} をそれぞれ左, 右部分木とすると, 根を 2 分節として持ち, n_1 , n_2 は, それぞれ $F_{2i-2} < n_1 \leq F_{2i-1}$, $F_{2i-1} < n_2 \leq F_{2i}$ であるから, 帰納法の仮定より, 高さ $h = (i - 1) + 1 = i$ である.

(ii) $F_{2i+1} < n \leq F_{2i+2}$ のとき, T_n は, T_{n_1} , T_{n_2} , T_{n_3} をそれぞれ左, 中, 右部分木とすると, 根を 3 分節として持ち, n_1 , n_2 , n_3 は, それぞれ $F_{2i-1} < n_1 \leq F_{2i}$, $F_{2i-2} < n_2 \leq F_{2i-1}$, $F_{2i-1} < n_3 \leq F_{2i}$ であるから, 帰納法の仮定より, 高さ $h = (i - 1) + 1 = i$ である.

故に, $F_{2i} < n \leq F_{2i+2}$ なる n に対する T_n の高さ h は, すべて $h = i$ となることが示された. これより, T_n の根から葉までのすべての路長が等しくなることが帰納的に示されるから, T_n は 2-3 木となる.

又, 構成法より, $\beta(T_n) \in \mathcal{T}_n$ なることも明らかである. (証明終)

(定理 5.1 の証明)

十分性: T_n が性質 M を満たすとき, 補題 5.5 より $\beta(T_n) \in \mathcal{T}_n$ であり, 補題 5.4 よりコスト最小である.

必要性: T_n が性質 M を満たしていないと仮定する. すると, 補題 5.5 より $\beta(T_n) \notin \mathcal{T}_n$ となり, 又, 補題 5.6 より任意の n に対して $\beta(T_n) \in \mathcal{T}_n$ である 2-3 木が存在することより, T_n はコスト最小でありえない. 故に, 比較コスト最小のときは, 性質 M を満たす. (証明終)

2-結果比較器を用いたときの比較コスト最小2-3木のコストを次の定理で与える。

【定理5.2】 n 個の葉を持つ2-3木の最小比較コスト $C_{\min}(n)$ は、 $F_i < n \leq F_{i+1}$ のとき、

$$C_{\min}(n) = \sum_{j=2}^{i-1} F_{j-1} \cdot j + i(n - F_i)$$

で表される。

5.5 構成アルゴリズム

昇順に並んだ K 個の見出しから、 n 個 ($n = K + 1$) の葉を持つ比較コスト最小2-3木 T_n を構成するアルゴリズムを示す。アルゴリズムは、P1, P2 の2段階より成る。

P1. 構成手続き BUILD によって2-3木 T_n の形を決める。

Procedure BUILD (n):

入力: n

出力: T_n

begin

$F_i < n \leq F_{i+1}$ なる i に対して

if $i \leq 2$ **then** $T_n \leftarrow \square$

else

if $i = \text{偶数}$ **then**

begin

T_n の根 \leftarrow 2分節;

if $n - F_i \leq F_{i-2} + 1$ **then**

begin

T_n の左部分木 \leftarrow BUILD ($F_{i-2} + 1$);

T_n の右部分木 \leftarrow BUILD ($n - F_{i-2} - 1$)

```

    end
    else
    begin
     $T_n$  の左部分木 ← BUILD ( $n - F_i$ );
     $T_n$  の右部分木 ← BUILD ( $F_i$ )
    end
end
else
begin
 $T_n$  の根 ← 3 分節 ;
if  $n - F_i \leq F_{i-3} + 1$  then
begin
 $T_n$  の左部分木 ← BUILD ( $F_{i-2}$ ) ;
 $T_n$  の中部分木 ← BUILD ( $F_{i-3} + 1$ ) ;
 $T_n$  の右部分木 ← BUILD ( $n - F_{i-1} - 1$ )
end
else
if  $n - F_i \leq F_{i-2}$  then
begin
 $T_n$  の左部分木 ← BUILD ( $F_{i-2}$ ) ;
 $T_n$  の中部分木 ← BUILD ( $n - F_i$ ) ;
 $T_n$  の右部分木 ← BUILD ( $F_{i-1}$ )
end
else
begin
 $T_n$  の左部分木 ← BUILD ( $n - F_i$ ) ;
 $T_n$  の中部分木 ← BUILD ( $F_{i-2}$ ) ;
 $T_n$  の右部分木 ← BUILD ( $F_{i-1}$ )
end
end
end
end

```

P2. T_n の節を，中間順 FILLORDER でたどり，順番に見出しを置いていく．ここで，FILLORDER は，次のように再帰的に記述されるものである．

- 1 FILLORDER で左部分木をたどる
 - 2 根をたどり，見出しを置く
 - 3 FILLORDER で中部分木をたどる
 - 4 根をたどり，見出しを置く
 - 5 FILLORDER で右部分木をたどる．
- } 根が3分節の場合

5.6 結 言

2-結果比較器を用いて2-3木の探索を行ったとき，比較コストが最小となるような2-3木を特徴づけ，そのコストを求めた．これらは，フィボナッチ数を用いて表現できることが示された．本節のモデルは，3分節内では常に小さい方の見出しから比較すると仮定したものであるが，3分節内の2個の見出しの比較順をランダムに決めるとしたときの最適木およびコストについては発表論文(6)で考察した．

比較コスト最小2-3木の性能を評価するには，ランダムな2-3木におけるコストとの比較，又は，比較コスト最大2-3木のコストとの比較が考えられる．これらは，今後の研究課題として残されている．

第6章 疎モデル2-3木の最適構成問題

6.1 緒言

本章では、疎モデル2-3木の平均路長に関する最適構成問題を考察する。疎モデルの2-3木は2-3葉木(2-3 leaf search tree)ともよばれる。一般に、多分木(multiway tree)の探索コストは節訪問コストと節内探索コストとの和で表される。⁽²⁶⁾ 2-3葉木では、節内探索コストは分離値の比較コストに相当する。分離値の比較コストを0としたときは、明らかに高さが最小の木がコスト最小である。また、節訪問コストを0にしたときのコスト最小木は、文献(23)の比較コスト最小2-3木に等しい。そこで、節訪問コストと比較コストの両方を考慮した場合を考える。

発表論文(7)では、2-3葉木が節間に重み β の枝を持ち、3分節内の分離値間に重み α の枝を持つようなモデルを考察した。これは、節訪問コストと比較コストの両方を考慮したものであるが、任意の比 α/β と葉の数 n が与えられたときの最適木の高さを特徴づけることができなかった。そこで、探索モデルを簡略化し、単に2分節の重みを1、3分節の重みを V ($V > 1$)として、荷重路長(weighted path length)の総和を探索コストとする。これは3分節内の分離値の比較順序を無視したもので、実際の探索を正確に表現してはいないが、 $11/7 < V < 8/5$ を除く V の値にたいして、任意の数の葉を持つ最適木を特徴づけることができる。

6.2ではコストを定義し、6.3では、高さ h で n 個の葉を持つ木のうちでコストが最小の木を特徴づけ、そのコストを示す。6.4では、同じ葉の数にたいして、各高さの最小コストを比較することにより、 V の値と最適木の高さとの関係を調べる。6.5では、最適木の構成アルゴリズムを与え、構成例を示す。

6.2 定義

2-3葉木におけるコストの定義を行う。2-3葉木では見出しは左から右に昇順に並んでおり、見出しの数は葉の数に等しい。内部節におかれる分離値および葉におかれる見出しの値はすべて異なるものとする。図6.1に7個の見出し1, 2, ..., 7を持つ2-3葉木の例を示す。節内の点は分離値を表す。分離値の配置は次のようである。任意の節に含まれる分離値は、左部分木(右部分木)に含まれるすべての分離値と見出しよりも大きい(小さい)。また、任意の3分節内の小さい(大きい)方の分離値は、中部分木に含まれるすべての分離値と見出しよりも小さい(大きい)。

2-3葉木における探索アルゴリズムは2-3木の場合と同様である。見出し探索において、節を1回訪問するコストを β 、分離値を1回比較するコストを α とすると、2分節の探索コストは $\alpha + \beta$ であり、3分節の探索コストは、探索する見出しが3分節内の小さい方の分離値よりも小さいとき $\alpha + \beta$ であり、大きいとき $\alpha + 2\beta$ である。ここで、探索モデルを簡略化して、3分節内の分離値の比較順序を無視し、すべての路上の3分節内の比較コストは一定であるとする。つまり、路上の3分節の重みはすべて同じであるが、2分節の重みよりは大きいものとする。2分節と3分節の重みの比だけを考えればよいから、2分節の重みを1

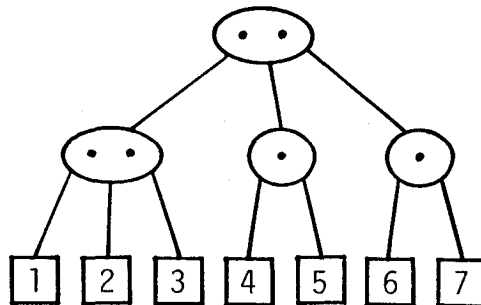


図 6.1 2-3葉木の例

とし、3分節の重みを $V(V > 1)$ とする。このようなモデルにおける探索コストを V -コストとよぶ。

【定義6.1】 n 個の見出しを持つ2-3葉木 T_n の V -コスト $C(T_n)$ を次式で表す。

$$\begin{aligned} C(T_n) &= \sum_{i=1}^n \omega_i = \sum_{i=1}^n (a_i + b_i V) \\ &= A + BV \end{aligned} \tag{6.1}$$

ここで、 ω_i は左から i 番目の見出しの荷重路長を表し、 a_i, b_i はそれぞれ路上の2分節、3分節の個数を表す。

n 個の葉を持つ2-3葉木 T_n の高さが h であるとする

$$a_i + b_i = h \quad (1 \leq i \leq n) \tag{6.2}$$

であり、

$$A + B = nh \tag{6.3}$$

である。式(6.1)と(6.2)より、 T_n の V -コストは

$$C(T_n) = nh + (V - 1)B \tag{6.4}$$

で表される。 n と V が与えられたとき、 V -コストが最小の木を最適木とよぶ。

図6.1の木では、各見出しの荷重路長はそれぞれ $w_1 = w_2 = w_3 = 2V, w_4 = w_5 = w_6 = w_7 = 1 + V$ だから、 V -コストは $C(T_7) = 4 + 10V$ である。以後、単に2-3葉木を木、 V -コストをコストともよぶ。

6.3 高さ h のコスト最小木

高さ h で n 個の葉を持つ木のうちで V -コスト最小の木を特徴づけ、そのコストを示す。それは以下に示すように、高さを固定したときは V の値にかかわらずその高さでのコスト最小木が特徴づけられるからである。 n 個の葉を持つ木の

ストは式(6.4)で表されるが、この式から高さ h が固定されたときは路上の3分節数の総和 B が最小の木がコスト最小であるといえる。よって、この B を V 係数とよび、 V 係数が最小の木を特徴づける。

2-3木のある部分集合をフラット木と名づけ、その木が高さを固定したときのコスト最小木となることを示す。以下では、見出しや分離値の配置はコストつまり V 係数に無関係であるので、木の形だけに着目して議論する。

【定義6.2】 任意の2路上の3分節数の差が、たかだか1であるような2-3葉木をフラット木とよぶ。

まずフラット木の性質を調べる。

【補題6.1】 高さ i の木 T の任意の路上における3分節の数が $j-1$ か j ($j=1, \dots, i$) であれば(2分節の数は $i-j+1$ か $i-j$)、木 T の葉の数 n は

$$2^{i-j+1} 3^{j-1} < n \leq 2^{i-j} 3^j$$

である。

(証明) 高さ h についての帰納法により証明する。

(I) $h=1$ のとき

路上の3分節の数は $j=1$ で、葉の数は $n=3$ だから成り立つ。

(II) $h=i-1$ で成り立つとして、 $h=i$ のときを示す。

(i) 高さ i の木が2分節を根に持つとき、路上の3分節の数が $j-1$ か j であれば、左、右部分木においても $j-1$ か j である。左、右部分木の葉の数をそれぞれ n_1, n_2 とすると、帰納法の仮定より

$$2^{(i-1)-j+1} 3^{j-1} < n_1 \leq 2^{(i-1)-j} 3^j$$

$$2^{(i-1)-j+1} 3^{j-1} < n_2 \leq 2^{(i-1)-j} 3^j$$

である。よって

$$2(2^{(i-1)-j+1} 3^{j-1}) < n = n_1 + n_2 \leq 2(2^{(i-1)-j} 3^j)$$

より，成り立つ．

(ii) 高さ i の木が 3 分節を根に持つとき，路上の 3 分節の数が $j-1$ か j であれば，左，中，右部分木においては $j-2$ か $j-1$ である．左，中，右部分木の葉の数をそれぞれ n_1, n_2, n_3 とすると，帰納法の仮定より

$$2^{(i-1)-(j-1)+1} 3^{(j-1)-1} < n_1 \leq 2^{(i-1)-(j-1)} 3^{(j-1)}$$

$$2^{(i-1)-(j-1)+1} 3^{(j-1)-1} < n_2 \leq 2^{(i-1)-(j-1)} 3^{(j-1)}$$

$$2^{(i-1)-(j-1)+1} 3^{(j-1)-1} < n_3 \leq 2^{(i-1)-(j-1)} 3^{(j-1)}$$

である．よって

$$3(2^{(i-1)-(j-1)+1} 3^{(j-1)-1}) < n \leq 3(2^{(i-1)-(j-1)} 3^{(j-1)})$$

より成り立つ．

(証明終)

【補題 6.2】 高さ i の木 T の路上の 3 分節数が $j-1$ か j のとき，つまり T の葉の数が $2^{i-j+1} 3^{j-1} < n \leq 2^{i-j} 3^j$ ， $j=1, \dots, i$ のとき， T の V 係数は

$$B = n(j+2) - 2^{i-j+1} 3^j \quad (6.5)$$

である．

(証明) 高さ h についての帰納法により証明する．

(I) $h=1$ のとき

$$\text{根が 2 分節のとき: } B = 2(2) - 2^2 = 0$$

$$\text{根が 3 分節のとき: } B = 3(1+2) - 2 \cdot 3 = 9 - 6 = 3$$

により，明らかに成り立つ．

(II) $h < i$ のとき成り立つとして $h=i$ のときを示す．

(i) 高さ i の木 T が 2 分節を根に持つとき，左，右部分木の葉の数をそれぞれ n_1, n_2 とする ($n = n_1 + n_2$)． T の路上の 3 分節数が $j-1$ または j のとき，左，右部分木においても路上の 3 分節数は $j-1$ または j である．よって帰納法の仮定により，

$$\begin{aligned}
B &= \{n_1(j+2) - 2^{(i-1)-j+1} 3^j\} + \{n_2(j+2) - 2^{(i-1)-j+1} 3^j\} \\
&= (n_1+n_2)(j+2) - 2(2^{(i-1)-j+1} 3^j) \\
&= n(j+2) - 2^{i-j+1} 3^j
\end{aligned}$$

となり、成り立つ。

(ii) 高さ i の木 T が 3 分節を根に持つとき、左、中、右部分木の葉の数をそれぞれ n_1, n_2, n_3 とする ($n = n_1 + n_2 + n_3$)。 T の路上の 3 分節数が $j-1$ または j のとき、左、中、右部分木の路上の 3 分節数は $j-2$ または $j-1$ である。よって帰納法の仮定により、

$$\begin{aligned}
B &= \{n_1(j+1) - 2^{(i-1)-(j-1)+1} 3^{(j-1)}\} + \{n_2(j+1) - \\
&\quad 2^{(i-1)-(j-1)+1} 3^{(j-1)}\} + \{n_3(j+1) - 2^{(i-1)-(j-1)+1} 3^{(j-1)}\} \\
&\quad + (n_1+n_2+n_3) \\
&= (n_1+n_2+n_3)(j+2) - 3(2^{i-j+1} 3^{j-1}) \\
&= n(j+2) - 2^{i-j+1} 3^j
\end{aligned}$$

となり、成り立つ。

(証明終)

次に、高さ h のコスト最小木はフラット木に含まれることを示す。

【補題 6.3】 高さ h で n 個の葉を持つ木 T がフラット木でなければ、 T は同じ高さで同じ葉の数を持つ木のうちでコスト最小ではない。

(証明) 木 T はコスト最小であるがフラット木ではないとする。つまり、 T には路上の 3 分節数の差が d ($d \geq 2$) であるような 2 路が少なくとも 1 組存在する。今、 3 分節数の差が $d = 2$ であるような路の組が 1 組だけあるとし、この路を a, b とする。 a の 3 分節数は b のそれよりも 2 だけ少ないとする。さて、路 a と b との路上にはそれぞれ 2 分節と 3 分節が同じレベルに存在するはずである。このうち、最も大きいレベルつまり最も葉に近いレベルの 2 分節と 3 分節をそれぞれ β, τ とする (図 6.2)。節 β と τ のレベルを i とする。 β を根とする部分木 A

と、 τ を根とする部分木Bの葉の数を考えてみる。路aのレベルiから葉までの路上の3分節数は、路bのそれよりも1だけ少ない。aとbを除くすべての路については、路上の3分節数は等しいはずである(そうでなければ仮定に反する)。その3分節数をjとすると、aとbの3分節数はそれぞれ(j-1とj)または(jとj+1)のどちらかである。よって部分木AとBの葉の数は、補題6.1と同様の議論により、それぞれ $2^{i-j} 3^j - 1$, $2^{i-j} 3^j$ または $2^{i-j} 3^j$, $2^{i-j} 3^{j+1}$ のどちらかであることがわかる。ところで、部分木AとBを交換する、つまり τ を β の親の子とし β を τ の親の子とすると、V係数Bは1減る。よってTはコスト最小ではありえない。つまり仮定に反する。3分節数の差が $d=2$ の組が2組以上ある場合も同様の議論により示される。 $d \geq 3$ についても同様である。 (証明終)

ここで、フラット木が高さhのコスト最小木であること、および、そのコストを示す。

【補題6.4】 高さiの木Tがフラット木であれば、Tは高さiで同じ数の葉を持つ木のうちでコスト最小であり、葉の数が $2^{i-j+1} 3^{j-1} < n \leq 2^{i-j} 3^j$, $j=1, \dots, i$ のとき、そのコストは

$$c(T) = ni + (V-1) \{ n(j+2) - 2^{i-j+1} 3^j \} \quad (6.6)$$

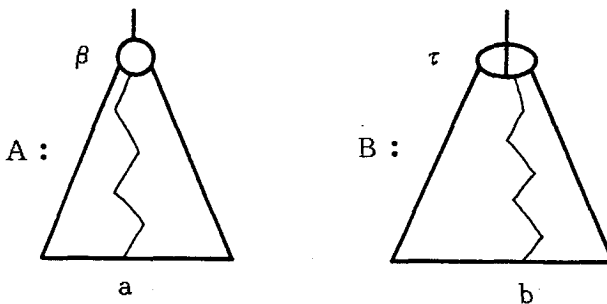


図 6.2 部分木AとB

である。

(証明) 補題6.3より, 高さ i のコスト最小木はフラット木に含まれる。ところが, 補題6.2より, フラット木の V 係数はすべて等しい。よってフラット木はコスト最小である。式(6.4)に式(6.5)を代入すると式(6.6)のようになる。

(証明終)

6.4 V の考察

最適木の高さ h と V の値との関係を調べる。 n 個の葉を持つ最適木は, V の値により異なるが, いずれも前節で示したフラット木に含まれる。そこで, V の値にたいして $\lceil \log_3 n \rceil \leq h \leq \lfloor \log_2 n \rfloor$ のうちの高さのフラット木が最適木になるかを調べる。それには, 2つの異なる高さにたいする式(6.6)が等しい, とおいた V を変数とする方程式の根を考察すればよい。その結果, すべての根が一定の狭い範囲内に存在することを示す。

【定理6.1】 同じ葉の数を持つ高さ h , $h+1$ ($l=1, 2, \dots$) のフラット木のコストをそれぞれ C , C_l とする。式(6.1)より

$$C = A + BV, \quad C_l = A_l + B_l V$$

とおく。このとき $C = C_l$ なる V を V_0 とする。つまり

$$V_0 = - (A - A_l) / (B - B_l) \tag{6.7}$$

とおくと,

$$11/7 \leq V_0 \leq 1.6 \tag{6.8}$$

が成り立つ。

定理6.1を証明するのに次の補題6.5と6.6が必要である。補題6.5では, 高さ h の木にたいする葉の数 $2^{h-1}3^i$ と, 高さ h' の木にたいする葉の数 $2^{h'-1}3^j$ が交番することを示す。例えば, 高さ3の木にたいする葉の数は 2^3 ,

$2^2 \cdot 3$, $2 \cdot 3^2$, 3^3 であり, 高さ4の場合は 2^4 , $2^3 \cdot 3$, $2^2 \cdot 3^2$, $2 \cdot 3^3$, 3^4 である. これより, $2^2 \cdot 3 < 2^4 < 2 \cdot 3^2 < 2^3 \cdot 3 < 3^3 < 2^2 \cdot 3^2$ が成り立つことがわかる. ところで, 高さ h と $h+1$ の木が存在する葉の数 n の範囲は $2^{h+1} \leq n \leq 3^h$ である.

【補題6.5】 高さ h の木にたいする葉の数 $2^{h-i} 3^i$ と, 高さ $h+1$ の木にたいする葉の数 $2^{h+1-j} 3^j$ について, $j = i - \lceil \alpha i \rceil$, ($i = 1, 2, \dots$) のとき次の不等式が成り立つ.

$$2^{h-i+1} 3^{i-1} < 2^{h+1-j} 3^j < 2^{h-i} 3^i \quad (6.9)$$

ここで, $\alpha = 1 / (\log_2 3 - 1) \cong 1.7095$ であり, $\lceil \alpha i \rceil \leq i \leq h$, $0 \leq j \leq h - \lceil \alpha i \rceil$ である.

(証明) $\lceil \alpha i \rceil = \alpha i + \theta$ ($0 < \theta < 1$) とおく. $j = i - \alpha i - \theta$ なる j について

$$2^{-i+1} 3^{i-1} < 2^{i-j} 3^j < 2^{-i} 3^i \quad (6.10)$$

が成り立つことを示せばよい.

$$\begin{aligned} 2^{-i+1} 3^{i-1} - 2^{i-j} 3^j \\ = 2^{-i+1} 3^{i-1} - 2^{i-i+\alpha i+\theta} 3^{i-i-\alpha i-\theta} \end{aligned}$$

ここで, $(2/3)^{\alpha i} = 2^{-i}$ より $2^i (2/3)^{\alpha i} = 1$ だから

$$\begin{aligned} &= 2^{-i} 3^i \{ (2/3) - (2/3)^\theta \} \\ &< 0 \quad (0 < \theta < 1 \text{ より}) \end{aligned}$$

同様にして

$$\begin{aligned} 2^{i-j} 3^j - 2^{-i} 3^i \\ = 2^{-i} 3^i \{ (2/3)^\theta - 1 \} \\ < 0 \quad (0 < \theta < 1 \text{ より}) \end{aligned}$$

以上より式(6.10)が成り立つ.

(証明終)

補題6.6では、微小正関数を与える。

【補題6.6】 $0 < \theta < 1$ のとき、関数

$$f(\theta) = 2 + \theta - 2(3/2)^\theta \quad (6.11)$$

と

$$g(\theta) = 3 - \theta - 3(2/3)^\theta \quad (6.12)$$

はともに非負で 0.051 を越えない。図6.3に $f(\theta)$ と $g(\theta)$ を示す。

(定理6.1の証明) まず、 l が一定であるとして、 V_0 は $n = 2^{h-i} 3^i$ 、

($\lceil \alpha l \rceil \leq i \leq h$) のとき最大値をとり、 $n = 2^{h-i-j} 3^j$ ($0 \leq j \leq h - \lceil \alpha l \rceil$) のとき最小値をとることを示す。

補題6.5より、 $\lceil \alpha l \rceil \leq i \leq h$ 、 $0 \leq j \leq h - \lceil \alpha l \rceil$ なる i, j について、 $j = i - \lceil \alpha l \rceil$ のとき $2^{h-i+1} 3^{i-1} < 2^{h-i-j} 3^j < 2^{h-i} 3^i$ が成り立つから、 $2^{h-i+1} 3^{i-1} < n \leq 2^{h-i} 3^i$ なる n について示す。

式(6.6)より $2^{h-i+1} 3^{i-1} < n \leq 2^{h-i} 3^i$ のとき

$$c = nh + (V-1) \{ n(i+2) - 2^{h-i+1} 3^i \} \quad (6.13)$$

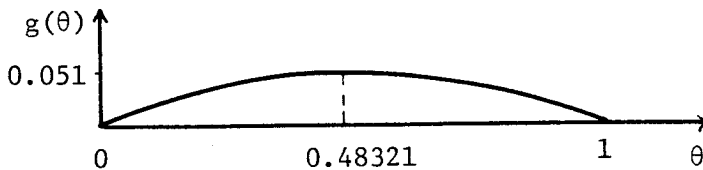
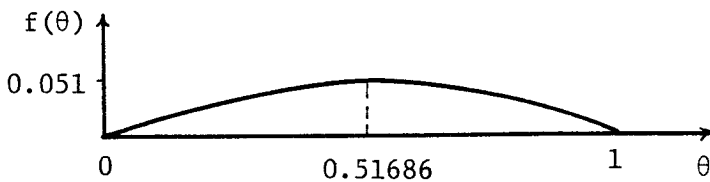


図 6.3 関数 $f(\theta)$ と $g(\theta)$

$2^{h+l-j+1} 3^{j-1} < n \leq 2^{h+l-j} 3^j$ のとき

$$c_l = n(h+l) + (V-1)\{n(j+2) - 2^{h+l-j+1} 3^j\} \quad (6.14)$$

である。

$2^{h+i+1} 3^{i-1} < n \leq 2^{h+i-j} 3^j$ のとき式(6.13)と(6.14)より

$$V_0 = 1 + \frac{l}{(i-j) - 2^{h+i}\{2^{-i} 3^i - 2^{l-j} 3^j\}/n} \quad (6.15)$$

となる。式(6.15)を $j = i - \alpha l - \theta$ とおくと

$$V_0 = 1 + \frac{l}{(\alpha l + \theta) - 2^{h+i+1} 3^i \{1 - (2/3)^\theta\}/n} \quad (6.16)$$

となる。

式(6.16)は、 $1 - (2/3)^\theta > 0$ であることと、分母が正であることから、 $n = 2^{h+i+1} 3^{i-1} + 1$ のとき最大値

$$V_0 = 1 + \frac{l}{\alpha l + \theta - 3 + 3(2/3)^\theta + \frac{3\{1 - (2/3)^\theta\}}{2^{h+i+1} 3^{i-1} + 1}} \quad (6.17)$$

をとり、 $n = 2^{h+l-j} 3^j$ のとき最小値

$$V_0 = 1 + \frac{l}{\alpha l + \theta + 2 - 2(1.5)^\theta} \quad (6.18)$$

をとる。

$2^{h+l-j} 3^j < n \leq 2^{h-i} 3^i$ のとき、補題6.5より式(6.15)を $j = i - \alpha l - \theta + 1$ とおいて

$$V_0 = 1 + \frac{l}{(\alpha l + \theta - 1) - 2^{h+i+1} 3^i \{1 - (3/2)(2/3)^\theta\}/n} \quad (6.19)$$

ここで、 $1 - (3/2)(2/3)^\theta < 0$ であり、分母は正だから $n = 2^{h-i} 3^i$ のとき

最大値

$$V_0 = 1 + \frac{l}{\alpha l + \theta - 3 + 3 (2/3)^\theta} \quad (6.20)$$

をとり, $n = 2^{h+l-j} 3^j + 1$ のとき最小値

$$V_0 = 1 + \frac{l}{\alpha l + \theta + 2 - 2(1.5)^\theta - (1 - (2/3)(3/2)^\theta) \left(1 + \frac{2}{2^{h+l-j} 3^j + 1}\right)} \quad (6.21)$$

をとる.

以上より $2^{h+l} 3^{l-1} < n \leq 2^{h+l} 3^l$ なる n について次のことがいえる. 式(6.17), (6.20)と $1 - (2/3)^\theta > 0$ より $n = 2^{h+l} 3^l$ のとき最大値をとり, 式(6.18), (6.21)と, $1 - (2/3)(3/2)^\theta > 0$ より, $2^{h+l-j} 3^j$ のとき最小値をとる.

更に, 式(6.18)は, $l = 1$ のとき最小値をとり, 式(6.20)は $l = 1$ のとき最大値をとることを示す.

式(6.18)は, $l = 1, 2, \dots$ にたいして

$$V_0 = 1 + \frac{l}{\alpha l + \theta + 2 - 2(1.5)^\theta} \geq \frac{11}{7} \quad (6.22)$$

を満たすことを示す.

$l = 1$ のとき $V_0 = 11/7$ となるから, $l \geq 2$ のとき成り立つことを示す. 両辺から1を引いて両辺に左辺の分母を掛け, 差をとると

$$\begin{aligned} & l - (4/7) \{ \alpha l + \theta + 2 - 2(1.5)^\theta \} \\ &= l \{ 1 - (4/7) \alpha \} - (4/7) \{ 2 + \theta - 2(1.5)^\theta \} \end{aligned}$$

ここで, $f(\theta) = 2 + \theta - 2(1.5)^\theta$ とおくと, 補題6.6より

$$> l \{ 1 - (4/7) \alpha \} - (4/7)(0.051)$$

$\alpha < 1.71, 4/7 < 0.572$ だから,

$$> 0.02188 l - 0.029172$$

$$> 0 \quad (l \geq 2)$$

となる.

式(6.20)は, $l = 1, 2, \dots$ にたいして

$$V_0 = 1 + \frac{l}{\alpha l + \theta - 3 + 3 (2/3)^l} \leq 1.6 \quad (6.23)$$

が成り立つことを示す.

$l = 1$ のとき $V_0 = 1.6$ となるから, $l \geq 2$ の場合を示す. 下限の場合と同様にして差をとると

$$\begin{aligned} & l - 0.6 \{ \alpha l + \theta - 3 + 3 (2/3)^l \} \\ & = l (1 - 0.6 \alpha) + 0.6 \{ 3 - \theta - 3 (2/3)^l \} \end{aligned}$$

ここで, $g(\theta) = 3 - \theta - 3 (2/3)^l$ とおくと, 補題6.6より

$$< l (1 - 0.6 \alpha) + 0.6 (0.051)$$

$\alpha > 1.7$ だから,

$$< -0.02 l + 0.0306$$

$$< 0 \quad (l \geq 2)$$

となる.

(証明終)

【注6.1】 $l \rightarrow \infty$ のとき V_0 は次の値に近づく.

$$V_0 \rightarrow \log_2 3 \cong 1.585 \quad (6.24)$$

6.5 最適木の構成法

6.5.1 構成アルゴリズム

n 個の見出しと V が与えられたとき, 最適木を構成するアルゴリズムを示す.

手続き **CONS** (h):

$2^{h-i+1} 3^{i-1} < n \leq 2^{h-i} 3^i$ ($0 \leq i \leq h$) のとき, レベル $0 \sim i-2$ がすべて 3分節, レベル $i-1 \sim h-1$ がすべて 2分節から成る木をトップダウンに構成し, レベル $h-1$ の 2分節のうち, $(n - 2^{h-i+1} 3^{i-1})$ 個を 3分節におきかえる.

メインプログラム:

```
begin
  if  $1 < V \leq 11/7$  then CONS ( $\lceil \log_3 n \rceil$ )
  else
    if  $V \geq 1.6$  then CONS ( $\lfloor \log_2 n \rfloor$ )
    else
      begin
         $C_{mh} := \infty$ ;
        for  $h := \lceil \log_3 n \rceil$  step 1 until  $\lfloor \log_2 n \rfloor$  do
          begin
             $i := \lceil (\log_2 n - h) / (\log_2 3 - 1) \rceil$ ;
             $C_h := (V - 1) \{ n(i + 1) - 2^{h+1} (1.5)^i \}$ 
               $+ n \cdot h$ ;
            if  $C_h < C_{mh}$  then
              begin
                 $C_{mh} := C_h$ ;  $mh := h$ 
              end
            end;
          end
        CONS ( $mh$ )
      end
    end
  end
```

この構成アルゴリズムによって最適木が構成できることを示す。手続き **CONS** (h) によって構成された木 T が, 高さ h の木のうちでコスト最小であり, n 個の葉を持つことを示す。構成法から T がフラット木であることは明らか

であるから、補題6.4によりコスト最小である。木Tがn個の葉を持つことを示す。CONSによってトップダウンに木を構成したとき、レベルh-1がすべて2分節であれば $2^{h-1} 3^{i-1}$ 個の葉を持つ。レベルh-1の2分節のうち、 $(n - 2^{h-1} 3^{i-1})$ 個を3分節におきかえると葉の数は $2^{h-1} 3^{i-1} + (3-2)(n - 2^{h-1} 3^{i-1}) = n$ となる。

メインプログラムは、Vの値にたいする最適木の高さを決め、手続きCONS(h)を呼ぶものである。前節の定理6.1では、高さの異なる2つのフラット木(高さhとh+1)のコストCとC'

$$C = A + BV, \quad C' = A' + B'V$$

にたいして

$$V_0 = -(A - A') / (B - B')$$

なる V_0 がすべて $11/7 \sim 1.6$ の間に存在することが示された。このことから、コストの差 $C - C' = (A - A') + (B - B')V$ を考えると、 $A < A'$ 、 $B > B'$ であるから(高さが高い木ほど2分節が多く3分節が少ない),

$$1 < V < 11/7 \leq V_0 \quad \text{のとき} \quad C < C'$$

$$V > 1.6 \geq V_0 \quad \text{のとき} \quad C > C'$$

といえる。いいかえると、前者では高さが低いフラット木ほどコストが小さいこと、後者では高さが高いフラット木ほどコストが小さいことを意味する。つまり、最適木の高さhは

$$1 < V \leq 11/7 \quad \text{のとき} \quad h = \lceil \log_3 n \rceil$$

$$V \geq 1.6 \quad \text{のとき} \quad h = \lfloor \log_3 n \rfloor$$

となる。

微小区間 $11/7 < V < 1.6$ では、最適木の高さがわからないから、 $\lceil \log_3 n \rceil \leq h \leq \lfloor \log_3 n \rfloor$ なるすべての高さについて式(6.6)のコストを計算し、そのうちで

最小値をとる高さを最適木の高さとする。

構成アルゴリズムにより木の形を構成した後、葉に見出しを配置する。アルゴリズムの計算時間は $O(n)$ である。

6.5.2 構成例

構成例として $V=1.574$ ，見出しが $1, 2, \dots, 17$ の場合を示す。 $11/7 < 1.574 < 1.6$ であり， $\lceil \log_3 17 \rceil = 3$ ， $\lfloor \log_2 17 \rfloor = 4$ だから， $h = 3, 4$ である。

$h = 3$ のとき

$$i = \lceil (\log_2 17 - 3) / (\log_2 3 - 1) \rceil = 2$$

$$C_3 = 17 \cdot 3 + 0.574 \{ 17 \cdot 4 - 2^4 (1.5)^2 \} = 69.369.$$

$h = 4$ のとき

$$i = \lceil (\log_2 17 - 4) / (\log_2 3 - 1) \rceil = 1$$

$$C_4 = 17 \cdot 4 + 0.574 \{ 17 \cdot 3 - 2^3 (1.5) \} = 69.722.$$

よって $C_3 < C_4$ だから， $CONS(3)$ を呼び、葉に見出しを配置すると図6.4のようになる。

構成例の2番めとして $V=1.5$ のときを考える。これは文献(6, p.367)の次数路長(degree path length)に等しい。つまり，2分節と3分節の子の数の比を表している。このとき， $1.5 < 11/7$ だから最適木の高さは $h = \lceil \log_3 n \rceil$ となる。

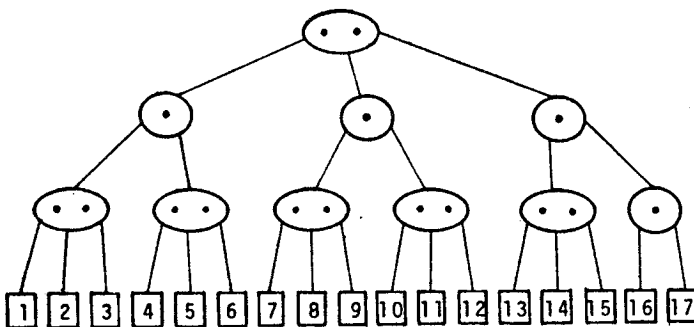


図 6.4 最適木の例

6.6 結 言

2-3木の最適構成問題の一つとして、疎モデルの場合を考察した。本探索モデルでは、路上の3分節の重みはすべて等しいと仮定し、2分節と3分節の重みの違いだけを考慮しているが、このコストは結果的に文献(27)のコストに一致する。文献(27)では、3分木について最適構成問題を扱っているが、このときのコストは文献(6, p.367)の次数路長と外部路長(external path length)の和で表されるコストを一般化したものである。葉のレベルがすべて等しい3分木が2-3木であるから、本モデルは3分木の自由度に制限のある場合を扱ったものともいえる。

第7章 結 論

高さ平衡木の総数にたいする再帰的性質および具体的表現が明らかになった。再帰的表現を考察することは計算機科学において重要である。それはコンピュータで具体的な値を求めるときの計算効率に関係しているからである。ここで用いた手法とくに漸化式の変換法は、木をはじめとする他の組合せ問題の数え上げにも適用できると考えられる。2-3木を前提としたいいくつかの探索モデルにたいして、路長に関する最適木の特徴、コスト、および構成法が明らかになった。これらはデータ構造の初期設定のモデルを考える際の参考になると考えられる。以上の研究は、木構造の性質を理解し、認識を深める上でも役立つと考えられる。

高さ平衡木の総数を近似的に表現することは今後の課題である。これには解析学的考察が必要とされる。また、高さ平衡木の数え上げにたいして、組合せ論における反転公式や Polya の方法が適用できるか調べるのも興味深い。高さ平衡木の最適構成に関しては、見出しの重みが一律でない場合の考察が残されている。この場合の最適木は動的計画法(dynamic programming)によって構成できるが、効率のよい構成アルゴリズムを考えることが研究の対象となろう。

謝 辞

本論文をまとめるにあたり，懇切なる御指導を賜った大阪大学工学部 手塚慶一教授に謹んで感謝の意を表す。また，有益な御教示，御助言を頂いた中西義郎教授，西田俊夫教授に厚く御礼申し上げる。

常日頃，研究活動全般にわたり多大な御指導，御鞭撻を頂き，本論文の第3章と第4章の数え上げ問題について御指導を頂いた信州大学工学部 中村義作教授に深く感謝申し上げます。また，最適木構成問題研究のきっかけを与えられ，第5章の内容について御教示，御討論を頂いた同研究室 岡本正行助教授に心から感謝する。

最後に，お世話になった大阪大学，信州大学の方々，ワープロ入力を手伝ってくれた妻名保美に感謝する。

文 献

- (1) Adel'son-Vel'skii, G.M. and Landis, E.M. : An Algorithm for the Organization of Information. Dokl. Akad. Nauk SSSR, Vol.146, pp.263-266 (1962).
- (2) Knuth, D.E. : The Art of Computer Programming 1 : Fundamental Algorithms, Addison-Wesley (1971).
- (3) Aho, A.V., Hopcroft, J.E. and Ullman, J.D. : The Design and Analysis of Computer Algorithms. Addison-Wesley (1974).
- (4) Aho, A.V., Hopcroft, J.E. and Ullman, J.D. : Data Structures and Algorithms, Addison-Wesley (1983).
- (5) N.ビルト, 片山訳 : アルゴリズム+データ構造=プログラム, 日本コンピュータ協会 (1979).
- (6) Knuth, D.E. : The Art of Computer Programming 3 : Sorting and Searching, Addison-Wesley (1973).
- (7) Bear, J.L. and Schwab, B. : A Comparison of Tree-Balancing Algorithms, C.ACM, Vol.20, No.5, pp.322-330 (1977).
- (8) Karlton, P.L., Fuller, S.H., Scroggs, R.E. and Koehler, E.B. : Performance of Height-Balanced Trees, C.ACM, Vol.19, No.1, pp.23-28 (1976).
- (9) Zaki, A. and Baer, J.L. : A Comparison of Query Costs in AVL and 2-3 Trees, Technical Report #78-02-01, University of Washington (1979).
- (10) C.ベルジュ, 野崎訳 : 組合せ論の基礎, サイエンス社 (1973).

- (11) Chvatal, V., Klarner, D.A. and Knuth, D.E. : Selected Combinatorial Research Problems, STAN-CS-72-292, Stanford University (1972).
- (12) Zaks, S. : Lexicographic Generation of Ordered Trees, Theoretical Computer Science, Vol.10, pp.63-82 (1980).
- (13) Huffman, D.A. : A Method for the Construction of Minimum Redundancy Codes, Proc. IRE, Vol.40, pp.1098-1101 (1952).
- (14) Knuth, D.E. : Optimum Binary Search Trees, Acta Informatica, Vol.1, No.1, pp.14-25 (1971).
- (15) Dickson, L.E. : History of the Theory of Numbers, Vol.1, XVII, chelsea (1971).
- (16) ホール, 岩堀訳 : 組合せ理論, 吉岡書店 (1974).
- (17) Yao, A.C-C. : On Random 2-3 Trees, Acta Informatica, Vol.9, pp.159-170 (1978).
- (18) 弓場, 星 : 木構造を用いた見出し探索の技法, 情報処理, Vol.21, No.1, pp.28-49 (1980).
- (19) Miller, R.E., Pippenger, N., Rosenberg, A.L. and Snyder, L. : Optimal 2,3-Trees, SIAM J. Comput., Vol.8, No.1, pp.42-59 (1979).
- (20) Odryzko, A.M. : Periodic Oscillations of Coefficients of Power Series that Satisfy Functional Equations, Advances in Mathematics, Vol.44, No.2, pp.180-205 (1982).
- (21) Riordan, J. : An Introduction to Combinatorial Analysis, John Wiley & Sons, Inc. (1958).

- (22) Gupta, U., Lee, D.T. and Wong, C.K. : Ranking and Unranking of 2-3 Trees, SIAM J. Comput., Vol.11, No.3, pp.582-590 (1982).
- (23) Rosenberg, A.L. and Snyder, L. : Minimal-Comparison 2, 3-Trees, SIAM J. Comput., Vol.7, No.4, pp.465-480 (1978).
- (24) Rosenberg, A.L. and Snyder, L. : Time and Space-Optimality in B-trees, ACM Transactions on Database Systems, Vol.6, No.1, pp.174-183 (1981).
- (25) Hoshi, M. and Yuba, T. : Searching in Preorder Trees and Its Evaluation, The Trans. of IECE of Japan, Vol.E61, No.1, pp.8-14 (1978).
- (26) Vaishnavi, V.K., Kriegel, H.P. and Wood, D. : Optimum Multiway Search Trees, Acta Informatica, Vol.14, pp.119-133 (1980).
- (27) Bagchi, A. and Roy, J.K. : On V-Optimal Trees, SIAM J. Comput., Vol.8, No.4, pp.524-541 (1979).
- (28) Choy, D.M. and Wong, C.K. : Optimal α - β Trees with Capacity Constraint, Acta Informatica, Vol.10, pp.276-296 (1978).
- (29) Ottmann, T., Parker, D.S. and Rosenberg, A.L. : Minimal-Cost Brother Trees, SIAM J. Comput., Vol.13, No.1, pp.197-217 (1984).
- (30) Zaki, A. : Insertion and Search in 2-3 Trees versus HB(1) Trees, Doctor of Philosophy, University of Washington (1977).
- (31) Otter, R. : The Number of Trees, Annals of Mathematics, Vol.49, No.3, pp.583-599 (1948).

- (32) Brown, M.R. : A Partial Analysis of Random Height-Balanced Trees,
SIAM J. Comput, Vol.8, No.1, pp.33-41 (1979).
- (33) Nievergelt, J. and Reingold, E.M. : Binary Search Trees of Bounded
Balance, SIAM J. Comput., Vol.2, No.1, pp.33-43 (1973).
- (34) Luccio, F. and Pagli, L. : On the Height of Height-Balanced Trees,
IEEE Trans. on Computers, Vol.c-25, No.1, pp.87-90 (1976).