ローテーションの回数に基づく2分木間の距離

著者	清水 道夫
雑誌名	長野県短期大学紀要
巻	47
ページ	83-89
発行年	1992-12
URL	http://id.nii.ac.jp/1118/00000525/

長野県短期大学紀要　第47号　83-89頁　1992年12月
Journal of Nagano Prefectural College, No. 47, pp. 83-89, December 1992

ローテーションの回数に基づく2分木間の距離

清水　道夫

A Distance between Two Binary Trees based on the Number of the Rotations

Michio SHIMIZU

Abstract

A certain distance between two binary trees is defined. The distance is the minimal number of the rotations changing from tree A to tree B. If the distance between tree A and tree B is 1, we say that the two trees are connected. Binary trees are expressed by the codewords as the computer representations, and ranked by the lexicographic generation of codewords. An algorithm that makes the table of the connection using the rank is proposed. The distance is expressed by the minimal path length on the graph made from the table.

1．まえがき

木やグラフの近さや類似性に関する問題は，パターン認識や誤り訂正構文解析などに関連して検討されてきた（文献1）。距離とはこの近さや類似性を表す尺度で，いろいろな距離が提案され検討されている。たとえば，節の置換，挿入，脱落によるもの（文献2），部分木の交換操作によるもの（文献3）などである。

本報告では，最も基本的なデータ構造である2分木を取り上げ，そのローテーション（回転）の回数に基づく木の距離を考える。ローテーションは，木の変換操作の一つで，見出し（キー）の挿入や削除にともなう更新のための基本的技法である（文献4）。ここでは，2分木への見出しの挿入と削除は無視し，更新がローテーションのみによって行われると仮定する。同じ大きさの2分木 A と B があるとき，A から B へ変換するローテーションの最小回数を木の距離とする。この距離は，データ構造の更新における手間の一評価になると考えられる。なお，木 A と木 B の距離が1のとき，つまり一回のローテーションで変換されるとき，この2つの木は接続しているという。

2．では，コードワードとよばれる整数列を導入し，それが2分木に対応付けられることを示す。さらに，その辞書式順序に基づいて2分木のランク（番号）付けを行う。3．では，コードワードの分類表，および，ランクを要素とする接続表を紹介する。4．では，2分木の接続関係（接続表で示す）を，木の大きさ n について再帰的に求めるアルゴリズムを示す。5．では，2分木間の

〒380　長野市三輪8-49-7　長野県短期大学
Nagano Prefectural College, 49-7 Miwa 8-chome, Nagano 380, Japan.

距離がこの接続表から作成したグラフ上の最短経路問題となることを示す。

2．2分木の符号化

2分木は，根と2個の2分木（右部分木と左部分木という）から再帰的に構成される。この二次元的な2分木をコンピュータ上で能率よく扱うために，それを整数列として符号化する。2分木の符号化はいろいろ提案されている（文献5，6，7）が，ここでは，D. Zerling（文献5）によって導入されたコードワード（CW）を用いる。

CWは次の性質を満たす。長さLのCWをX_1, X_2, …, X_Lとおくと，X_1からX_b（$b=1, 2, \cdots, L$）までの総和はb以下である。たとえば，$L=1$から3のときすべてのCWをかくと，

$L=1$：0，1

$L=2$：00，01，02，10，11

$L=3$：000，001，002，003，010，011，012，020，021，100，101，102，110，111

となる。ただし，要素間のカンマは省略している。なお，これは辞書式順序（lexicographic order）に並んでおり，この順番を長さLのときのランクとよぶ。たとえば，$L=3$のとき010はランク5である。

CWの生成方法は，図1のバックトラッキング木を用いればわかりやすい。節内の見出しは分岐の数を表しており，見出しtの節から分岐された子の見出しを右から順に2，3，…，$t+1$とする。ただし，根の見出しを2とする。そして，分岐された枝には左から0，1，…，tとラベルをつける。図1の木で，根から枝をたどってラベルを並べると，$L=1\sim3$のCWが生成される。

次に，図2に2分木のローテーションを説明しておく。図2の（a）から（b）への変換を右ローテーション、（b）から（a）への変換を左ローテーションとよぶ。右ローテーションでは，節

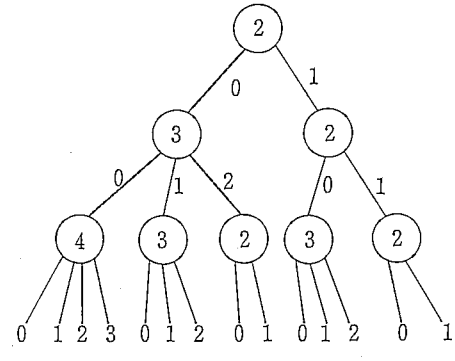

図1　バックトラッキング木
Fig. 1. A backtracking tree.

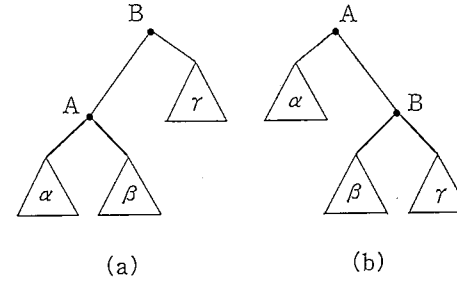

図2　ローテーション
Fig. 2. The rotation.

Aの右部分木βが節Bの左部分木になる。ここに，部分木α，β，γは空の場合もある。なお，（b）から（a）への変換を節Aに関する左ローテーションともいう。

2分木の大きさは節の数nで表す。大きさnの2分木の総数はカタラン数（Catalan number）${}_{2n}C_n/(n+1)$で表されることはよく知られているが，これは長さ$L=n-1$のCWの総数に一致する。以後，長さLのCWを，$X=x_{n-1}, x_{n-2}, \cdots, x_2, x_1$とかき，これを大きさ$n$にたいするCWとよぶ。またこの辞書式順序を$n$にたいするランクという。

ところで，2分木がCWにどのように対応しているかを示す。じつは，CWの各要素は2分木の左端路上の各節からみた右部分木に対応してい

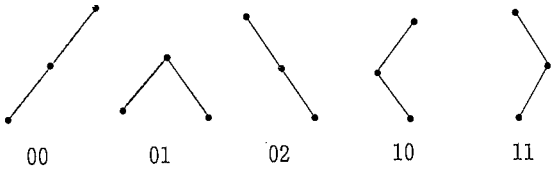

図3　$n=3$の2分木とCW
Fig. 3. Binary trees and their CW($n=3$)

る。x_1は根に関する左ローテーションによって，根の右部分木が空になるまでの回数を表している。つぎに，根の左子をpとし，pの右部分木をrとすると，x_2は根に関する左ローテーションが終了した時点での，pに関する左ローテーションによってrが空になるまでの回数を表している。以下同様であるが，CWは左端路だけからなる線形の2分木（0…0）に変換されるまでの，左端路上の各節（末端の節を除く）に関する左ローテーションの回数を表している。$n=3$の2分木とそのCWを図3に示す。

3．分類表と接続表

次節では，CW（2分木）の接続関係を再帰的に求めるアルゴリズムを提案するが，その準備として分類表と接続表の2種類の表を用意する。表1は$n=2\sim5$にたいする分類表で，これはnにたいするすべてのCWを左，右部分木の大きさにしたがって分類し，行と列がそれぞれ辞書式順序になるように並べたものである。左，右部分木の節の数をそれぞれu，vとすると，$(u, v)=(n-1, 0)$，$(n-2, 1)$，…，$(1, n-2)$，$(0, n-1)$のn通りに分けられる。分類表の一行分を横ブロックとよび，一列分を縦ブロックとよぶ。生成したCWを調べて，左，右部分木の大きさがつぎのように決められる。

アルゴリズム1

CWを右から走査して，1からjケタまでの数の総和がjでかつ$j+1$ケタ目が0または空のとき，このコードワードの縦ブロックを$(u, v)=(n-1-j, j)$，$j=0$，…，$n-1$とする。□

表1　分類表（$n=2\sim5$）

$n=2$

(1, 0)	(0, 1)
0	1

$n=3$

(2, 0)	(1, 1)	(0, 2)
00	01	02
10		11

$n=4$

(3, 0)	(2, 1)	(1, 2)	(0, 3)
000	001	002	003
010		011	012
020			021
100	101		102
110			111

$n=5$

(4, 0)	(3, 1)	(2, 2)	(1, 3)	(0, 4)
0000	0001	0002	0003	0004
0010		0011	0012	0013
0020			0021	0022
0030				0031
0100	0101		0102	0103
0110			0111	0112
0120				0121
0200	0201			0202
0210				0211
1000	1001	1002		1003
1010		1011		1012
1020				1021
1100	1101			1102
1110				1111

このアルゴリズムによって左右の部分木の大きさが決定できることを示す。CWの右からjケタまでの数の総和がjであるから，対応する2分木の左端路上の節について，根に近い方の節から順にj回の左ローテーションを行ったとき，元の木の根Pは左端路上の$j+1$番目の節Pになる。それは，左ローテーションの定義により，節Pが根の右部分木のどの節よりも左にくるからである。また，このとき，CWの$j+1$ケタ目が0ま

表2 接続表（$n=2\sim4$）

$n=2$

R	CW	CONN.
1	0	2
2	1	1

$n=3$

R	CW	CONN.	
1	00	2	4
2	01	1	3
3	02	2	5
4	10	5	1
5	11	4	3

$n=4$

R	CW	CONN.		
1	000	2	5	10
2	001	1	3	11
3	002	2	4	6
4	003	3	7	12
5	010	6	1	8
6	011	5	7	3
7	012	6	4	9
8	020	9	5	13
9	021	8	7	14
10	100	11	1	13
11	101	10	12	2
12	102	11	4	14
13	110	14	8	10
14	111	13	9	12

たは空であるから，節Pの右部分木が存在しない。よって，元の木の根Pの右部分木に含まれる節の数はちょうどjになる。

一方，表2は$n=2\sim4$にたいする接続表である。生成したCWを辞書式順序にならべて番号（ランク）付けする。表2のCONN.の欄には既に各CWに接続するCWがランクで書き込まれているが，これの求め方について次節で説明する。なお，CONN.の中が左右2つに分けられているが，これについてもあとで述べる。

4．接続する2分木

分類表および接続表から接続関係を再帰的に求めるにはつぎのようにする。大きさが$2\sim n-1$の接続表と大きさが$2\sim n$にたいする分類表が用意されているとして，大きさnの接続表を求める。分類表において接続関係にあるのは，あとでみるように横ブロック内での接続と縦ブロック内での接続に限られるが，双方の接続数の和は2分木のローテーション可能な箇所に等しいからつねに$n-1$になる。説明をわかりやすくするために，具体例として$n=4$の場合を考えていく。

［I］横ブロック内での接続

分類表の横ブロックは，根に関するローテーションによる変化を示している。したがって，CWの末尾のケタのみが1づつ変化するから，隣合うものが接続している。横ブロックの両端の（$n-1$，0）と（0，$n-1$）のCWには，それぞれ（$n-2$，1）と（1，$n-2$）のCWが接続し，それ以外の（u，v）のCWには（$u+1$，$v-1$）と（$u-1$，$v+1$）の2個のCWが接続する。たとえば，$n=4$の分類表の横ブロック（000，001，002，003）については，001には000と002が接続し，002には001と003が接続する。この関係をランクで表したものが，$n=4$の接続表のCONN.に示されている。CONN.の左側の区画が横ブロック内での接続である。

［II］縦ブロック内での接続

（i）（$n-1$，0）のとき

このブロックの2分木は右部分木が存在しないから，左部分木について接続関係を調べればよい。実際，左部分木のCWはこのブロックのCWの末尾の0を除いたものである。したがって，大きさ$n-1$の接続表から接続関係がわかる。$n=4$の分類表（表1）の（3，0）をみると，000，010，020，100，110の5個のCWがあるが，末尾の0を除くと$n=3$のCWになる。そこで，$n=3$の接続表から接続関係を調べると，たとえば00は01と10に接続している。したがって，000は010と100に接続するから，これをランクで示すとランク1はランク5とランク10に接続することになる。

（ii）（u，v），$n-2\geqq u\geqq 1$，$v=n-1-u$のとき

CWを左uケタと右vケタに分割すると，左uケタ分は縦ブロックの（u，0）に対応し，右vケタ分は（0，v）に対応している。つまり，

$(u,\ 0)$ は左部分木を表し、$(0,\ v)$ は右部分木を表す。ところで、$(u,\ v)$ 内での接続は、左、右部分木のどちらかが等しくて、他方が接続関係にある場合に限る。大きさ2～$n-1$の接続表からこの条件を満足しているかを調べ、$(u,\ v)$ 内のCWの接続関係を決定する。$n=4$の分類表の(2，1)には0 0 1と1 0 1の2つのCWがある。これを左2ケタと右1ケタに分けると、それぞれ0 0と1，1 0と1になる。そこで、$n=3$の接続表を調べると0 0と0 1が接続しているから、左部分木は接続している。また、右部分木は等しいから、0 0 1と1 0 1は接続している。つまり、ランク2はランク1 1に接続する。

(iii)$(0,\ n-1)$のとき

このブロックに含まれる2分木は左部分木が存在しないから、右部分木 T_{n-1}について接続関係を調べればよい。ところが、その対称形である$(n-1,\ 0)$のときと違って、このブロックのCWと T_{n-1}のCWが独立していないため、T_{n-1}のCWを簡単には取り出せない。そこで若干のくふうが必要になる。

この縦ブロックに含まれる2分木のCWを $X=x_{n-1},\ \cdots,\ x_2,\ x_1$とし、$T_{n-1}$のCWを $Y=y_{n-2},\ \cdots,\ y_2,\ y_1$とする。次のアルゴリズムによって、$X$を$Y$に変換する。ただし、大きさ$m$のすべてのCWの集合を$C_m$とする。

アルゴリズム2

ステップ1：$y_1=x_1-1$

ステップ2：$m=1$から$m=n-3$までつぎを繰り返す。

IF　$y_m=0$　AND　$y_{m-1},\ \cdots,\ y_1\in C_m$

THEN　$y_{m+1}=x_{m+1}-1$

ELSE　$y_{m+1}=x_{m+1}$　□

アルゴリズム2によってXがYに変換できることを示す。まず、左ローテーションの定義により、x_1は右端路上の枝の数を表していることに注意する。したがって、$y_1=x_1-1$となり、これがステップ1である。ステップ2では、T_{n-1}の左端路上の節はもはや左ローテーションの対象にならないから、その分の1を引いて調整する。これが$y_{m+1}=x_{m+1}-1$である。左ローテーションを何回か行って左端路上の$m+1$番目の節となるものが、既に左端路上の節となっている条件は、その節の右側にm個以上の節が存在しえないことである。つまり、大きさmの任意の2分木の左端路上の延長上の節となる場合である。これをCWで表すと、$y_m=0$，かつ，$y_{m-1},\ \cdots,\ y_1=a_m$となる。

表3に$(0,\ n-1)$，$(n=3\sim5)$のCWに対応するT_{n-1}のCWを示す。このT_{n-1}について、(i)のときと同様に大きさ$n-1$の接続表から接続関係を調べればよい。たとえば(0，3)に含まれる0 0 3は0 2を調べればよい。$n=3$の接続表から0 2は0 1と1 1に接続する。よって0 0 3は0 1 2と1 0 2に接続する。いいかえるとランク4はランク7とランク12に接続する。

表3　$(0,\ n)$とT_{n-1}の対応$(n=3\sim5)$

$n=3$

$(0,2)$	T_2
02	1
11	0

$n=4$

$(0,3)$	T_3
003	02
012	11
021	10
102	01
111	00

$n=5$

$(0,4)$	T_4
0004	003
0013	012
0022	021
0031	020
0103	102
0112	111
0121	110
0202	101
0211	100
1003	002
1012	011
1021	010
1102	001
1111	000

5．最短経路問題

表2から$n=4$の接続グラフをかくと図4のようになる。接続グラフの節内の数は各2分木のランクを表している。ランクRの節PからランクR'の節Qに至る最短路長が2分木間の距離に相当し，この距離を求めることは，グラフ上のいわゆる最短経路問題となる。最短経路を求めるダイクストラのアルゴリズムを用いて，$n=3\sim6$にたいする距離の平均を求めると表4（opt）のようになる。

なお，表4（pri）に2つのCWの和の平均値も示しておく。これは，0…0を経由して変換するときのローテーションの平均回数を表している。CWの総数がカタラン数$b_n={}_{2n}C_n/(n+1)$で表され，要素の総和がj（$0\leqq j\leqq n-1$）のCWの総数が$a_{n,j}={}_{n+j}C_j\ (n-j)/(n+j)$であることに注意すると，平均値は$2\Sigma ja_{n,j}/b_n$より$2n(n-1)/(n+2)$となる。CWの要素の総数は$n-1$以下であるから，変換に要するローテーションの回数はたかだか$2(n-1)$になるが，nが大きくなるに従って平均値がこの値に近づくことがわかる。

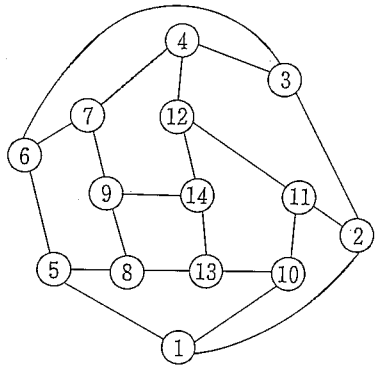

図4　$n=4$の接続グラフ
Fig. 4. A connected graph($n=4$)

表4　距離の平均値

n	opt	pri
3	1.2	2.4
4	2.0306	4
5	2.94331	5.71429
6	3.91988	7.5

6．むすび

本論文では，ローテーションの回数に基づく2分木間の距離を定義し，2分木の符号化を応用したアルゴリズムを提案した。コードワードと呼ばれる符号から2分木の接続関係が調べられることを示し，2分木をランクで表すことによって接続表や接続グラフを扱いやすいものにした。ただし，アルゴリズムの改良点として，次のことが残されている。それは，分類表の接続関係はどこの縦ブロックにおいてもその位置関係がおなじ，つまり，横ブロックどうしの接続になるらしいことである。たとえば，$n=5$の分類表（表1）の縦ブロック（4，0）内で，0 0 0 0と0 1 0 0が接続しているが，それを含む横ブロック内で同じ縦ブロックどうしがすべて接続する。0 0 0 1と0 1 0 1，0 0 0 3と0 1 0 2，0 0 0 4と0 1 0 3である。この証明はまだうまくいっていないが，このことを用いると2分木間の接続関係を求める再帰的なアルゴリズムはずっと簡潔になる。

文　献

1) 田中栄一：“構造をもつものの距離と類似度”，情報処理，31, 9, pp. 1270-1279 (1990)。
2) K.C. The Tree-to-Tree Correcting Problem', J. ACM, 26, pp. 422-433 (1979)。
3) K. Culik II and D. Wood：“A Note on Some Tree Similarity Measures”, Information Processing Letters, 15, pp. 39-42 (1982)。
4) D.E. Knuth: “The Art of Computer Programming III: Sorting and Searching”, Addison-Wesley, Reading MA. (1974)。
5) D. Zerling: “Generating Binary Trees Using

Rotations", J. ACM, 32, pp. 694-701 (1985)。

6) S. Zaks: "Lexicographic Generation of Ordered Trees", Theoretical Computer Science, 10, pp. 63-82 (1980)。

7) D. Rotem and Y.L. Varol: "Generation of Binary Trees from Ballot Sequences", J. ACM, 25, pp. 396-404 (1978)。