

表の格子グラフモデルと編集アルゴリズム

土田賢省¹, 本橋友江², 夜久竹夫³, 山澤聡³, 吉住寿洋¹

¹東洋大学, ²関東学院大学, ³日本大学

概要—はじめに単一ページの表の 8 分格子グラフモデルに基づく行削除アルゴリズムを提案する。次に複数ページの表の 16 分格子グラフモデルとその上の壁移動アルゴリズムを提案する。

Keywords: 表, Octgrids, アルゴリズム, H3-Code, H5CODE.

Grid Graph Models for Tables and Their Editing Algorithms

Kensei Tsuchida¹, Tomoe Motohashi², Takeo Yaku³, Satoshi Yamazawa³, Toshihiro Yoshizumi¹

¹Toyo University, ²Kanto-Gakuin University, ³Nihon University

Abstract—Firstly, we propose line deletion algorithm to be based on octgrid graph model for a table of single page. Next, we propose hexadeci-grid graph model for tables of multi pages and wall movement algorithm on hexadeci-grid graph model.

Keywords: Table, Octgrids, Algorithms, H3-Code, H5CODE.

1 はじめに

表処理ソフトウェアは数多く存在する。これらのソフトウェアの編集操作ではユーザが予期しない動作を引き起こしたり、反応が遅かったりする。このため、効率的かつ信頼性のある表編集処理の実現が望まれている。そこで、本論文ではそのような編集作業に適した表のグラフモデルとそのグラフモデルに基づく効率的な表編集アルゴリズムを扱う。矩形分割のセルの操作について J.L. Bentley が 1975 年に 4 分木[1]を、また K. Kozminsky と E. Kinnen が 1985 年に双対グラフ[2]を、そして夜久らが 2002 年に表編集のためのアルゴリズムとして東壁移動[3]、2003 年に列の幅の変化・列の挿入・セルの結合のアルゴリズム[4]を発表している。そして呉羽らが 2007 年に Hexadeci-grids とそれに対応した複数枚の地形図のファイルフォーマット H8CODE を発表している。

本稿では、第 2 節で準備として 8 分格子グラフの定義[3,4]と H3-Code[6]を説明し、第 3 節では、単一ページの表編集アルゴリズムを提案する。第 4 節では、16 分格子グラフ[5]をもとに複数ページの表編集について 16 分格子グラフモデルと表編集アルゴリズムを提案する。最後に、第 5 節で、結果と今後の課題について触れる。

2 準備

2.1 8 分格子グラフ[3, 4]

図 1 において矩形分割とその矩形分割に対応する 8 分格子グラフの例を示す(各々のノードの出次数は、最大 8 である)。

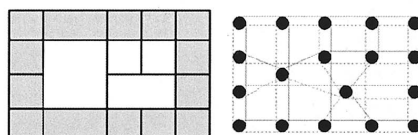


図 1: 矩形分割 D(左)と D に対応する 8 分格子グラフ(右)の例

2.2 8 分格子グラフのデータフォーマット[6]

8 分格子グラフを扱うためのファイル形式として、不均一型矩形分割コード H3-Code[6]が提案されている。

H3-Code は、8 分格子グラフに基づいた表形式文書表現するために使用される。図 2 は一つのセルのフィールドの例、そして図 3 は H3-Code のすべての構造になっている。

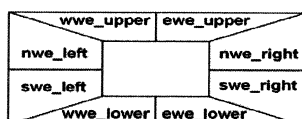


図 2: H3-Code の中のあるセルのフィールドの例

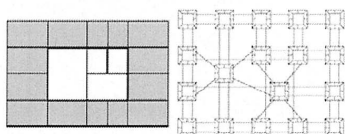


図 3: H3-CODE の全体の構造

3 単一ページの表編集

3.1 周辺セルを数えるアルゴリズム

北側周辺セルを数えるアルゴリズムである。

アルゴリズム CountColumns(G_D, v_x, m)

入力

G_D : 矩形分割Dに対する8分格子グラフ。

v_x : G_D 中のノード。

出力

m : v_x の幅をあらわす周辺セルの個数。

手順

1. m を 1, v_m を v_x の西壁辺をたどった北側の周辺セル, v_a を v_x の東壁辺をたどった北側の周辺セルとする。
2. m に 1 を足し, v_m を v_{m-1} から北壁辺をたどりひとつ右側のノードとする。
3. $v_m \neq v_a$ を満たすなら 2. に戻る。

このアルゴリズムの計算時間は $O(\sqrt{n})$ と予想される。

3.2 セル分割アルゴリズム

セルを分割するアルゴリズムである。このアルゴリズムで周辺セルは増減しないものとする。

次の図で入出力の例を示す。

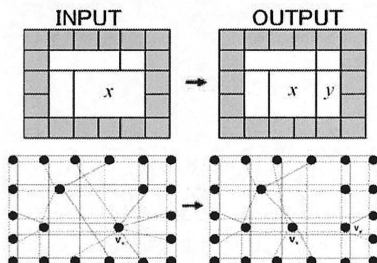


図 4: セルを分割した例 (入力で $m=3, n=2$)

アルゴリズム SplitCell(G_D, v_x, m, n, G_E)

入力

G_D : 矩形分割Dの8分格子グラフ。

v_x : 西壁辺と東壁辺に対応する北側の周辺セルが一致しないものとする。

m : v_x の北側周辺セルの個数 (CountColumns で得られた m)。

$n (n < m)$: n 単位分で 1 セル, 残りを 1 セル, 合計で 2 セルに分割される。

出力

G_E : v_x の西壁と東壁の間にある北側の周辺セルを n : $m-n$ の位置で分割して得られた 8 分格子グラフ。

手順

1. v_x にリンクされている東側ノードを v_a, v_b とし, v_y を加える。
2. v_x に東壁辺に接続された北側, 南側のノードをそれぞれ v_c, v_d とし, v_x と v_c の間の辺と v_x と v_d の間の辺を削除し, v_c と v_y, v_y と v_d の間にそれぞれ東壁辺を加える。
3. v_x と v_a の間の辺と v_x と v_b の間の辺を削除し, v_x と v_y の間に水平辺, v_y と v_a の間に北壁辺, v_y と v_b の間に南壁辺を加える。
4. v_x の北壁辺をたどった西側周辺ノード v_e と, v_x の西壁辺をたどった北側周辺ノード v_f を見つける。
5. v_f から北壁をたどり $n-1$ 東側に移動したノードを v_j , v_j にリンクされた一番西のノードを w_j とする。 v_j を東壁辺で一つずつ南に移動し, $nw(w_j) \geq nw(v_e)$ となる w_j を見つける。
6. v_x と v_{j-1} の間の東壁辺を削除, v_{j-1} と v_x, v_x と v_j の間に東壁辺を挿入する。
7. v_{k-1} を南壁辺によって v_{j-1} にリンクされた東側のノード, v_k を西壁辺によって v_{k-1} にリンクされた南側のノードとする。
8. v_y と v_{k-1} の間の西壁辺を削除, v_{k-1} と v_y, v_y と v_k のそれぞれの間に西壁辺を挿入する。

3.3 1行削除のアルゴリズム

指定したセルの北壁側 1 行を削除するアルゴリズムである。次の図で入出力の例を示す。

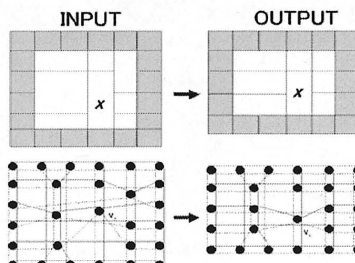


図 5: 指定したセルの北壁側 1 行を削除したもの

アルゴリズム DeleteRow-1(G_D, v_x, G_E)

入力

G_D : 矩形分割 D の 8 分格子グラフ.

v_x : 指定したセル x に対応する G_D の中のノード. ただし北側と南側の周辺ノードでないものとする.

出力

G_E : v_x の北壁側 1 行を削除して得られた 8 分格子グラフ.

手順

1. v_x から北壁边によってリンクされた西側周辺ノードを v_0 とする.
2. v_0 から北壁边によってリンクされた全ての頂点に N とマークする.
3. v_0 と南壁边でリンクされたノードで N とマークされた内部ノードを削除する.
4. v_0 とリンクされた内部ノードの北壁边を直す.
5. v_0 とリンクされた内部ノードの南壁边を直す.
6. v_0 と v_0 にリンクされた東側周辺ノードの垂直边を直し, 2つのノードを水平边とともに削除する.
7. 削除した行の高さを直す.

計算時間は n 行 m 列の場合, $O(m)$ と予想される.

3.4 複数行削除アルゴリズム

指定したセルを含むすべての行を削除するアルゴリズムである. 次の図で入出力の例を示す.

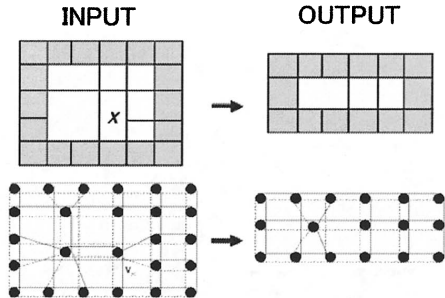


図 6: 指定したセルを含むすべての行を削除したもの

アルゴリズム DeleteRows(G_D, v_x, G_E)

入力

G_D : 矩形分割 D の 8 分格子グラフ.

v_x : G_D の中のノード, ただし北側と南側の周辺ノードでないものとする.

出力

G_E : v_x の高さの分だけ削除して得られた 8 分格子グラフ.

手順

1. v_x に南壁边によってリンクされた西側の周辺ノードを v_0 とする.
2. v_0 に隣接する南側のノードを v_h とする.
3. v_x に北壁边によってリンクされた西側の周辺ノードを v_i とする.
4. v_i に隣接する下側のノードを v_{i+1} とする.
5. DeleteRow-1(G_D, v_x, G_E).
6. i に 1 を足す.
7. $sw(v_i) < sw(v_h)$ なら Step 4 に戻る.

4 複数ページの表

4.1 16 分格子グラフモデル(cf [5])

多層型矩形分割を表現する方法として, 以下の図のような 16 分格子グラフ(hexadeci-grids)[5]がある.

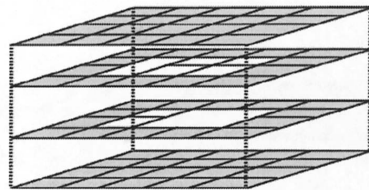


図 7: 多層型矩形分割の例

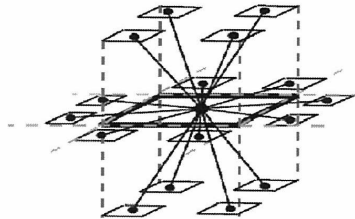


図 8: hexadeci-grids における一つのセルの周りのリンク ([5]より引用)

我々は多層型地形図の 16 分格子グラフモデル[5]を応用して, 複数ページの表を 16 分格子グラフで表現する.

4.2 モデルの実装; データフォーマット

ここでは, 我々は H3-Code を一般化して, 16 分格子グラフのデータフォーマット H5CODE を提案する. H5CODE は不均一型矩形分割コード(H8CODE[5])を変形したファイル形式である.

H5CODE はレコード (1 区切りのデータ) で表記し, データはレコード内のフィールド (データを書き込む最小単位) で表す.

表 1 は H5CODE のブロック構造を表しており, 第 1 ブロックから第 4 ブロックで構成される.

表 1: H5CODE のブロック構造

ブロック番号	ブロック名	説明
1	ヘッダー部	H5CODE の基本情報
2	リスト部	セルの情報
3	コンテンツ部	セル内部に記述するコンテンツの情報
4	レイヤー部	複数の矩形分割を管理するための情報

また図 9 で hexadeci-grids のリスト構造を示す[7].

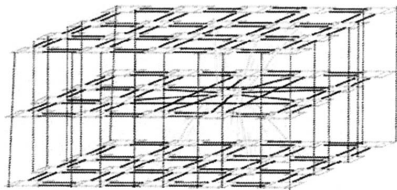


図 9: リスト構造 ([7]より引用)

4.3 多層型矩形分割に対応するアルゴリズム

Hexadeci-grids に対応したファイルフォーマット H8CODE で, CellUnification[7]などが提案されている.

ここでは新たに MoveSouthWall4Layer を提案する. このアルゴリズムは次の図のように一つのノードを入力として使い, そのセルに関係する行列を含む層の値も変化する.

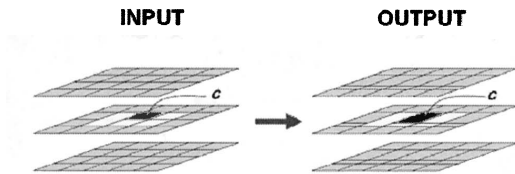


図 10: MoveSouthWall4Layer の入出力の例

アルゴリズム $\text{MoveSouthWall4Layer}(G_D, v, \delta, G_E)$

入力

G_D : 多層型矩形分割 D の 16 分格子グラフ

v_c : G_D 中のノード

$\delta \geq 0$: 移動値

$\Delta > 0$ は c の周辺層の南寄りに隣接する行のセルの高さで, それに対し $\Delta > \delta$ とする.

出力

G_E : D から南壁移動で得られた多層型矩形分割 E の 16 分格子グラフ.

手順

1. v_c の西側の周辺ノード v_a を見つける.

2. $sw(v_a)$ に δ 加える. v_a に南壁辺によってリンクされたノードを v_b とする.
3. $sw(v_b)$ に δ を加え, v_b は南壁辺で v_b にリンクされた東側ノードに変更する.
4. v_b が内部ノードなら step 3 に戻る.
5. $sw(v_b)$ に δ 加え, v_c から周辺ノード v_a まで移動.
6. v_d を v_a に西壁辺でリンクされた南側ノードとする.
7. $nw(v_d)$ に δ 加え, v_d を北壁辺で v_d にリンクされた東側ノードに変更する.
8. v_c が内部セルなら step 7 に戻る.
9. $nw(v_d)$ に δ を加える.
10. 各層で同様の操作を行う.

計算時間は n 行 m 列 l 層の場合 $O(ml)$ と予想される.

5 おわりに

本研究で単一ページの表編集として, セル分割アルゴリズム, 1 行削除アルゴリズム, そして行削除アルゴリズムを提案した. また, 複数ページの表編集でファイルフォーマット H5CODE と MoveSouthWall4Layer を提案した. 今後の課題として, 他の複数ページの表編集を提案することがある.

参考文献

- [1] J.L. Bentley, "Multidimensional binary searchtrees used for associative searching," Communications of the ACM, vol.18, no.9, pp.509-517, Sept. 1975.
- [2] K. Kozminsky and E. Kinnen, "Rectangular Duals of Planar Graphs", Networks 16, pp.145 - 157, 1985.
- [3] T. Motohashi, K. Tsuchida, and T. Yaku, "Attribute graphs for tables and their algorithms," Proc. Foundation of Software Engineering 2002, ed. K. Inoue, pp.183-186, Kindaigakusya, 2002.
- [4] T. Motohashi, K. Tsuchida, and T. Yaku, "Algorithms on Attribute Graphs for Tables", Proc. the 3rd Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications, pp.171-178, 2003
- [5] 赤木剛朗, 土田賢省, 野牧賢志, 本橋友江, 夜久竹夫, "16 分グラフによる多層型矩形分割の表現および表と地形図への応用", 日本応用数理学会, pp370-371, 2005.
- [6] Arita, T., Yaku, T. "H3-Code 2.3 Reference Manual.", <http://www.waap.gr.jp/waap-rr/waap-rr-06-001/index.html>, 2006.
- [7] Kureha, A., Kishira, S., Motohashi, T., Tsuchida, K., and Yaku, T. "hexadecimal Grid Graph Representation of Multilayer Rectangular Dissections and Its Applications.", 10th Society for Industrial and Applied Mathematics conf. Geometric Design & Comput, Abstract, Texas, USA, page 32, 2007.
- [8] 吉住寿洋, 塩野康徳, 山澤聡, 土田賢省, "8 分格子グラフにおけるグラフアルゴリズム", 日本応用数理学会 2008 年度年会, pp175-176, 2008.