

大規模フロアプランデータベースからの頻出部分パターンマイニング

川本 哲¹鈴木 祐介²正代 隆義³¹ 九州大学システム情報科学府情報理学専攻² 広島市立大学情報科学研究科³ 九州大学システム情報科学研究所情報理学部門

1 はじめに

フロアプランとは、モジュールと呼ばれる長方形の矩形ブロックを重複することなく、より大きな長方形上に配置した図形であり、その配置問題が VLSI レイアウトにおける重要な一ステップであることから、フロアプランに関して様々な側面から多くの研究がなされている [1, 2]. 特に、どのようにフロアプランを表現するかは、フロアプランの配置アルゴリズムの計算量に関わる基本的な課題のひとつである [3].

本論文では、フロアプランを表すのに適したデータ構造である BINDAG を新たに定義し、大規模フロアプランデータベースからの規模耐久性のある頻出部分パターンマイニング手法を提案する. BINDAG は、わずかな拡張で多次元フロアプランを表すこともできる. 本発表では、人工データからなるフロアプランデータベースからの頻出部分パターンマイニング実験の結果を報告し、提案手法の有効性を示す.

2 BINDAG

定義 1 2 方向無閉路有向グラフ (以下 BINDAG と略す) とは、3 組 (V, E_h, E_v) である. ここに、 V は頂点集合であり、 $E_h, E_v \subseteq V \times V$ はともに有向辺集合であり、すべての $u \in V$ について次を満たす. $|\{(u, v) \in E_h | v \in V\}| \leq 1$, $|\{(v, u) \in E_h | v \in V\}| \leq 1$, $|\{(u, v) \in E_v | v \in V\}| \leq 1$, $|\{(v, u) \in E_v | v \in V\}| \leq 1$.

フロアプランの罫線の交点を頂点、各罫線のセグメントのうち水平方向のものを E_h の元、垂直方向のものを E_v の元に有向辺の方向が右向きまたは下向きになるように割り当てることで、BINDAG を用いてフロアプランを表すことができる. 図 1 に、フロアプランを表す BINDAG の例を示す.

定義しうるすべての BINDAG がフロアプランを表しているわけではないことに注意されたい. 以降、フロアプランを表している BINDAG のみにについて論じる.

BINDAG $B = (V, E_h, E_v)$ の $v \in V$ について、 $(v, w) \in E_h$ を満たす w が存在するとき、 $R_B(v) = w$ とし、そのような w が存在しないとき、 $R_B(v) = \perp$ とする. また $(v, u) \in E_v$ を満たす u が存在するとき、 $D_B(v) = u$ とし、そのような u が存在しないとき、 $D_B(v) = \perp$ とする.

フロアプランを表す BINDAG $B = (V, E_h, E_v)$ において、フロアプランの左上の角に相当する頂点をその BINDAG の根と呼ぶ.

BINDAG 上で、根から深さ優先探索を行うことを考える. BINDAG の定義より、任意の頂点から出る有向辺の数は最大 2 本である. 深さ優先探索の際、次に探索する頂点の選択肢が 2 つある場合には必ず E_h に属する辺を先にたどるような探索方法を右下優先探索と呼ぶことにする. 任意の BINDAG について、右下優先探索で探索される頂点の順序は一意に定まる. また、任意の頂点は根から有向辺をたどることで到達できるので、右下優先探索で探索されない頂点は無い.

BINDAG $B = (V, E_h, E_v)$ において、 $v \in V$ の右下優先探索での順番を $N_B(v)$ と表記する. $N_B(v) = n$ であるとき、 $Vtx_B(n) = v$ と表記する. また、便宜上、 $N_B(\perp) = -1$ と定義する. たとえば、 $Vtx_B(7)$ の右に $Vtx_B(4)$ がある場合、 $N_B(R_B(Vtx_B(7))) = 4$ である.

列 $N_B(R_B(Vtx_B(1)))$, $\dots$, $N_B(R_B(Vtx_B(|V|)))$, $N_B(D_B(Vtx_B(1)))$, $\dots$, $N_B(D_B(Vtx_B(|V|)))$ を B の列表現と呼ぶ.

定義 2 2 つの BINDAG $B_1 = (V_1, E_{h1}, E_{v1})$ と $B_2 = (V_2, E_{h2}, E_{v2})$ について、それらが表すフロアプランが同型であるとき、2 つの BINDAG は同型であるとする. 2 つの BINDAG が同型であることは、すべての $u, v \in V_1$ について $(u, v) \in E_{h1} \Leftrightarrow (\psi(u), \psi(v)) \in E_{h2}$ かつ $(u, v) \in E_{v1} \Leftrightarrow (\psi(u), \psi(v)) \in E_{v2}$ であるような V_1 から V_2 への全単射 ψ が存在することと同値である.

このような ψ が存在するならば、任意の $v \in V_1$ について、 $N_{B_1}(v) = N_{B_2}(\psi(v))$ であるので、2 つの BINDAG が同型であることは $1 \leq i \leq |V_1|$ について、 $N_{B_1}(R_{B_1}(Vtx_{B_1}(i))) = N_{B_2}(R_{B_2}(Vtx_{B_2}(i)))$ かつ $N_{B_1}(D_{B_1}(Vtx_{B_1}(i))) = N_{B_2}(D_{B_2}(Vtx_{B_2}(i)))$ である.

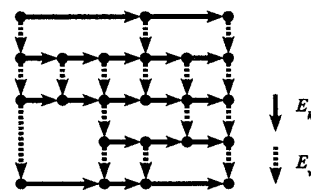


図 1: フロアプランを表す BINDAG の例

$Vtx_{B_2}(i))$ が成り立つことと同値である。よって、列表現が等しい2つの BINDAG は同型である。

3 提案するマイニング手法

フロアプランの中で罫線が矩形をなしている場合、その矩形の中もまたフロアプランとみなすことができる。このようなフロアプランを部分フロアプランと呼ぶ。

次のアルゴリズム 1 は、BINDAG $B = (V, E_h, E_v)$ で表されたフロアプラン F について、含まれるすべての部分フロアプランの四隅の角にあたる頂点を $O(|V|^2)$ 時間で出力する。

アルゴリズム 1 全部分フロアプランの抽出

```

For each  $v \in V$ 
   $L_1 := \emptyset$ ;  $t := v$ ;
  While not  $R_B(t) = \perp$  do
     $t := R_B(t)$ ;  $u := t$ ;
    While not  $D_B(u) = \perp$  do
       $u := D_B(u)$ ; Add  $u$  to  $L_1$ .  $U(u) := t$ ;
   $L_2 := \emptyset$ ;  $t := v$ ;
  While not  $D_B(t) = \perp$  do
     $t := D_B(t)$ ;  $u := t$ ;
    While not  $R_B(u) = \perp$  do
       $u := R_B(u)$ ; Add  $u$  to  $L_2$ .  $L(u) := t$ ;
  For each  $x \in L_1 \cap L_2$  Report  $(v, U(x), L(x), x)$ ;
```

四隅の角にあたる頂点が与えられれば、その部分フロアプランは部分フロアプランの頂点数に比例する時間で抽出できる。考慮するフロアプランの部屋数が定数で抑えられるとき、フロアプランの列表現はアルファベット数が定数で長さがフロアプランの頂点数に比例する文字列とみなすことができるので、線形時間で重複数を数えながら重複を除去することができる。よって、頂点数が高々 k のフロアプラン n 個について、それらに含まれるすべての部分フロアプランの重複の無い一覧と、それらの出現回数のリストは $O(k^2n + \ell)$ 時間で計算できる。ここで、 ℓ は含まれるすべての部分フロアプランの頂点数の総和である。

4 評価実験

3 節のアルゴリズムを実装し、このアルゴリズムがどの程度の規模のマイニングに適用しうるか評価した。実験用のデータとしては、人工的にランダム生成したフロアプランを用いた。実験では入力として部屋数が等しい定数個のフロアプランの集合を与え、それらのすべての部分フロアプランを計算させた。

なお、RAM の都合上、今回は発見した部分フロアプランをリストに追加していき、既にリストに存在する部分フロアプランと重複した場合はリストに追加せず、重複した回数を数えていくように実装している。

実験環境は 1GB の DDR2 SDRAM を搭載したデスクトップ PC (CPU は Intel® Core™2 Duo E6300) 上の Microsoft® Windows® XP Professional 上の Cygwin 環境で、C 言語で実装し、コンパイルには GCC 3.4.4 を用いた。対象とするフロアプランの数や部屋数を変えて実験を行った際の所要時間 (秒) は次のとおりである。

フロアプラン数 部屋数	64	128	256	512
50	0	0	1	5
200	2	5	17	55
800	17	53	170	564
3200	183	568	1830	メモリ不足

実験結果から、提案するアルゴリズムは十分な規模耐久性を持っているといえる。

5 おわりに

本論文では、大規模なフロアプランの集合からすべての部分フロアプランとその出現頻度を高速に計算するアルゴリズムを提案した。出現頻度に閾値を定めて出力するか否かを制御することで、このアルゴリズムを用いて頻出部分パターンマイニングを行うことができる。

BINDAG を拡張して $B = (V, E_1, E_2, \dots, E_n)$ のように有向辺集合をさらに増やすことで、 n 次元超立方体中に n 次元超立方体中を隙間無く詰め込んだものを表すことができる。このような拡張された BINDAG に対しても、ほぼ同一のアルゴリズムで同じ計算量で全部分パターンとそれらの出現頻度を計算できる。

参考文献

- [1] S. Nakano, "Enumerating Floorplans with n Rooms", IEICE Trans. Fundamentals E85-A(7), pp.1746–1750, 2002.
- [2] Z.C. Shen and C.C.N. Chu, "Bounds on the Number of Slicing, Mosaic, and General Floorplans", IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems 22(10), pp.1354–1361, 2003.
- [3] B. Yao, H. Chen, C.K. Cheng, and R. Graham, "Floorplan Representations: Complexity and Connections", ACM Trans. on Design Automation of Electronic Systems 8(1), pp.55–80, 2003.