L字形描画のコンパクトな符号
Compact Codes for L-floorplans

Md. Rezaul Karim[1], Shin-ichi Nakano[2] and Md. Saidur Rahman[3]

[1] Dept. of Computer Science and Engineering, University of Dhaka, Dhaka-1000, Bangladesh
rkarim@univdhaka.edu
[2] Dept. of Computer Science, Gunma University, Japan
nakano@cs.gunma-u.ac.jp
[3] Dept. of Computer Science and Engineering, Bangladesh University of Engineering and Technology(BUET), Dhaka-1000, Bangladesh
saidurrahman@cse.buet.ac.bd

Abstract. A floorplan is a partition of a rectangle into a set of rectilinear polygons. An L-floorplan is a floorplan where each rectilinear polygon is either a rectangle or an L-shape polygon. Floorplans have many important applications including VLSI layout. Since the size of floorplans may be huge, compact encodings of floorplans are desired. Several compact encodings of floorplans are known where each rectilinear polygon is a rectangle only.
In this paper we design two compact encodings for L-floorplans with $6f + 3L + 2n_2 - 2$ and $5f + 6L - 4$ bits, respectively, where f is the number of polygons, L is the number of L-shape polygons and n_2 is the number of vertices with degree 2. The encoding techniques are simple and both encoding and decoding can be performed in $O(f) = O(n)$ time.

1 Introduction

Compact encodings of grpahs have been studied for many classes of graphs [S03], for instance, trees [MR97], [T84] and plane graphs [CLL01], [CGHKL98], [KW89], [PY85]. See a nice textbook [S03].

The well known naive coding of ordered trees is as follows. Given an ordered tree T we traverse T starting at the root with depth first manner. If we go down an edge then we code it with 0, and if we go up an edge then we code it with 1. Thus any ordered tree T with n vertices has a code with $2(n-1) = 2m$ bits, where m is the number of edges in T. Some examples are shown in Fig 1.

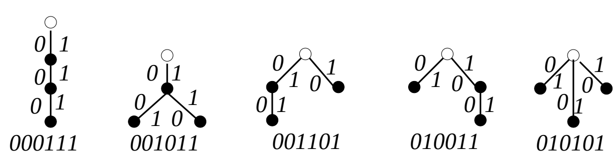


Fig. 1. A code for ordered trees.

On the other hand, the number of ordered trees with n vertices is known as the Catalan number C_{n-1}, and it is defined as follows [R00,GKP94].

$$C_n = \frac{1}{n+1}\frac{(2n)!}{n!n!} = \frac{4^n}{(n+1)\sqrt{\pi n}}\left(1 - \frac{1}{8n} + \frac{1}{128n^2} + \frac{5}{1024n^3} - \frac{21}{32768n^4} + O(n^{-5})\right)$$

For example, the number of ordered trees with four vertices is $C_{4-1} = 5$ as depicted in Fig. 1. We need at least $\log C_{n-1} = 2n - o(n) = 2m - o(n)$ bits to code an ordered tree with n vertices. So the naive coding using $2m$ bits for each ordered tree is asymptotically optimal.

A floorplan is a partition of a rectangle into a set of rectilinear polygons. Floorplans have many important applications including VLSI layout [KH97], [KK84], [Nak02], [RNG04], [RMN09]. Since the size of floorplans

Vol.2012-AL-140 No.1
2012/5/14

may be huge, compact encodings are desired. Several compact encodings of floorplans are known where each rectilinear polygon is a rectangle. See [CGHKL98], [CLL01], [GKP94], [KH97], [KK84], [KW89], [MR97], [Nak02], [TFI09], [YN06].

In this paper we consider compact encoding of L-floorplans, where an L-floorplan is a floorplan in which each rectilinear polygon is either a rectangle or an L-shape polygon. A floorplan can be considered as a plane graph G naturally, where the corners of the polygons are the vertices of G, the sides of the polygons are the edges of G and polygonal regions are the faces of G. Through the paper we thus use vertex, edge and face instead of corner, side and polygon, respectively, when it looks convenient. However, treating floorplan simply as plane graph is not enough for encoding since we need to preserve the direction of edges in a floorplan. For example, the two floorplans in Figs. 2(a) and 2(b) represent the same plane graph but they are different as floorplans; the two faces F_a and F_b share a horizontal side in the floorplan in Fig. 2(a) whereas they share a vertical side in the floorplan in Fig. 2(b).

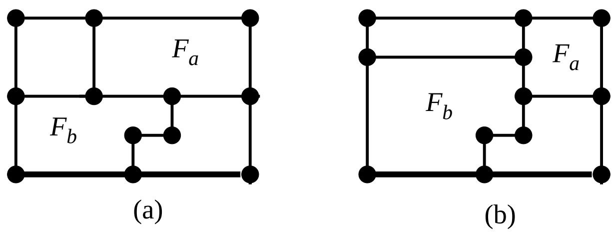


Fig. 2. Two floorplans corresponding to the same plane graph.

In this paper we design two compact encodings for L-floorplans. In our first method, we construct a tree from the given floorplan F and using a DFS on the tree we perform an encoding of the floorplan with $6f+3L+2n_2-2$ bits, where f is the number of faces, L is the number of L-shape faces and n_2 is the number of degree two vertices in F. In our second method, we divide each L-shape polygons into two rectangles by adding a dummy edge and obtain an encoding with $5f+6L-4$ bits.

The rest of the paper is organized as follows. Section 2 gives some definitions. Section 3 explains our first encoding of L-floorplans using $6f+3L+2n_2-2$ bits. Section 4 explains our second encoding of L-floorplans using $5f+6L-4$ bits. Finally section 5 is a conclusion.

2 Preliminaries

In this section we give some definitions. Let G be a connected graph. A *tree* is a connected graph with no cycle. A *rooted* tree is a tree with one vertex chosen as its root.

A drawing of a graph is *plane* if it has no two edges intersecting geometrically except at a vertex to which they are both incident. A plane drawing divides the plane into connected regions called *faces*. The unbounded face is called *the outer face*, and other faces are called *inner faces*.

A *floorplan* is a partition of a rectangle into a set of rectilinear polygons. An *L-floorplan* is a floorplan in which each rectilinear polygon is either a rectangle or an L-shape polygon. There are only four types of L-shape polygons as illustrated in Fig. 3. If two L-floorplans R_1 and R_2 have a one-to-one correspondence between faces in which the sequence of directions of common boundary of any pair (F_1, F_2) of adjacent faces match, then we say R_1 and R_2 are isomorphic.

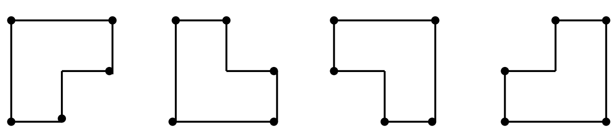


Fig. 3. Four different types of L-shape polygons.

Let n be the number of vertices of a given L-floorplan, m the number of edges and f the number of inner faces. In this paper we only consider L-floorplans in which no vertex has degree four. This is a typical assumption for floorplans [TFI09,YN06]. Let L be the number of L-shape faces and n_d the number of vertices with degree d. Note that $n_2 + n_3 = n$ and $n_0 = n_1 = n_4 = 0$ hold. Then an L-floorplan has $n_3 = n - n_2$ vertices with degree 3, and n_2 vertices with degree two, so $2m = 3(n - n_2) + 2n_2 = 3n - n_2$ holds. This equation and the Euler's formula $n - m + f = 1$ gives $n = 2f + n_2 - 2$ and $m = 3f + n_2 - 3$.

3 Compact Code I

In this section we give our first compact encoding for L-floorplans, based on the depth first search on a rooted tree. The encoding needs $2m + 3L + 4 = 6f + 3L + 2n_2 - 2$ bits for each L-floorplan.

Let R be an L-floorplan. We obtain a graph R' from R as follows. We first append either three or zero dummy edges into each L-shape face as illustrated in Fig. 4, where the dummy edges are depicted by dotted lines. Then we append two vertical dummy edges at the lower left corner and the lower right corner of the outer face as shown in Fig. 5. Finally, we replace the lower right corner of each face as shown in Fig. 6. See Fig. 7(c) for a complete example.

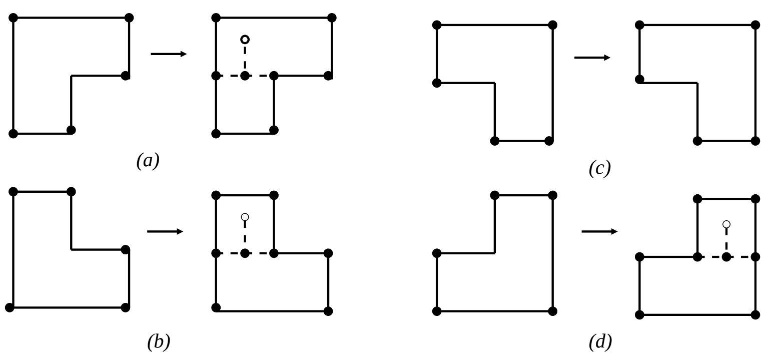


Fig. 4. Insertion of dummy edges into a L-shape face.

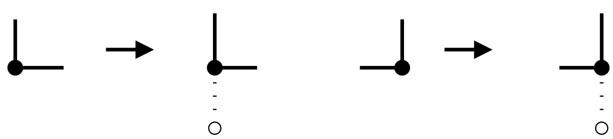


Fig. 5. Insertion of dummy edges at the lower left and lower right corner vertex

We now have the following lemma.

Lemma 1. *R' is a tree.*

Proof. Since we only break each cycle corresponding to each inner face at the lower right corner, the resulting graph has only one face and is still connected. $\mathcal{Q.E.D.}$

We now generate the code by traversing R' starting at the upper left corner of R'. We traverse the tree R' with depth first manner (with right priority). See Fig. 7(d). Each edge is traced exactly twice in both directions. When we arrive at each vertex from some direction, we always have only "two" choices for the next direction to trace, as shown in Fig. 8, even though there are four possible directions, up, down, left and right. We accomplish this by two ideas, the insertion of dummy edges and the introduction of the two

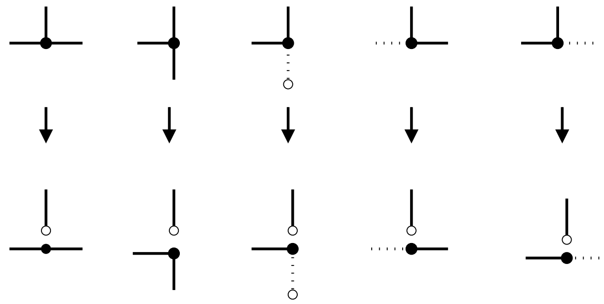


Fig. 6. Replacement of lower right corner vertex of an inner face

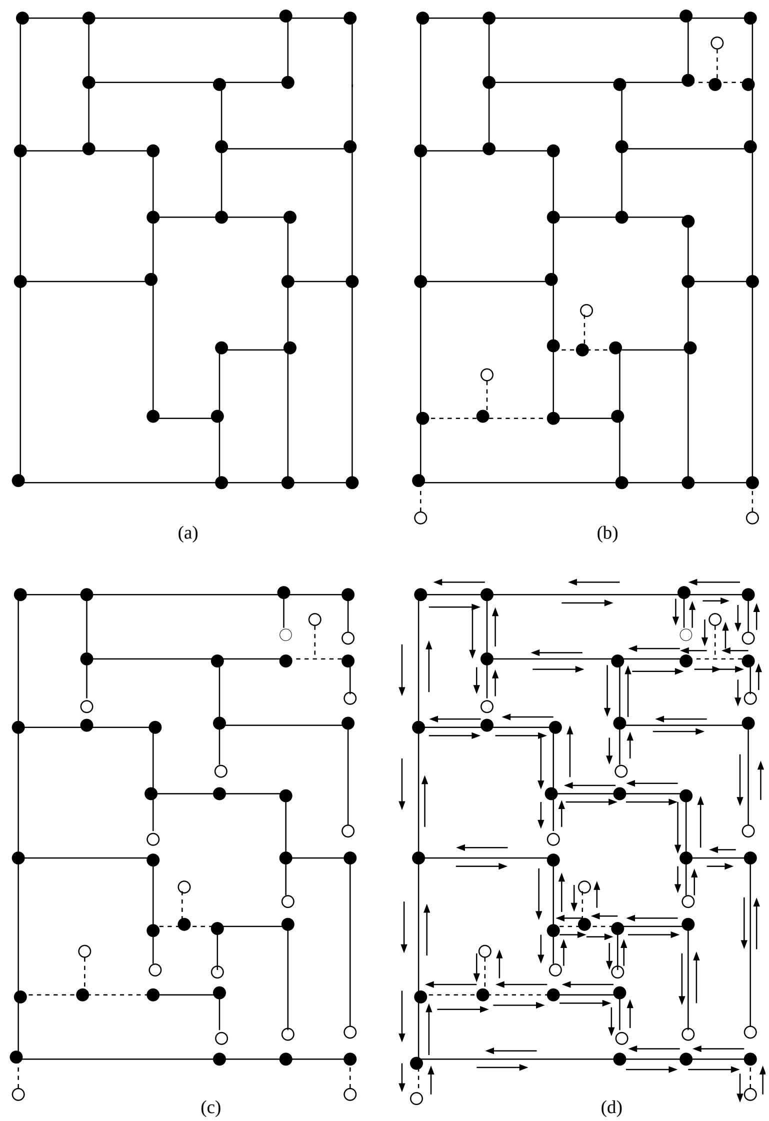


Fig. 7. (a) Given L-floorplan R, (b) insertion of dummy edges, (c) replacement of lower right corners, and (d) the trace.

cases for upward traces. Note that when we arrive at a vertex by tracing a vertical edge upward, we check whether the trace of the edge is (case $4(a)$) for the first time or (case $4(b)$) for the second time. Since each edge has traced exactly twice, basically we need two bits for each edge. However we have a chance to save. By the way of the insertion of dummy edge, in Case $4(a)$, the direction of the next two traces are always down then left. Thus for each L-shape face in Fig. 4(a), (b) or (d) we need not encode those traces.

For each L-shape face in Fig. 4(c), we need no dummy edges. By rotating the given L-floorplan R either by 0, 90, 180, or 270 degree so that we have the maximum number of L-shape faces in Fig. 4(c) we can minimize the number of dummy edges. After the proper rotation we can assume that the number of L-shapes in Fig. 4(c) is at least $\lceil L/4 \rceil$. Thus we can assume that the number of dummy edge is at most $2 + 3 \cdot \frac{3}{4}L$ after the rotation. We can encode the rotation by two bits.

Now we have the following theorem.

Theorem 1. *There is an encoding of L-floorplan with $6f + 3L + 2n_2 - 2$ bits.*

Proof. We can encode an L-floorplan with $2m + 2(2 + 3 \cdot \frac{3}{4}L) - 2 \cdot \frac{3}{4}L = 2m + 3L + 4 = 6f + 3L + 2n_2 - 2$ bits. $\mathcal{Q.E.D.}$

Given the $6f + 3L + 2n_2 - 2$ bits code we can easily reconstruct the original floorplan R with a simple linear-time algorithm with a stack.

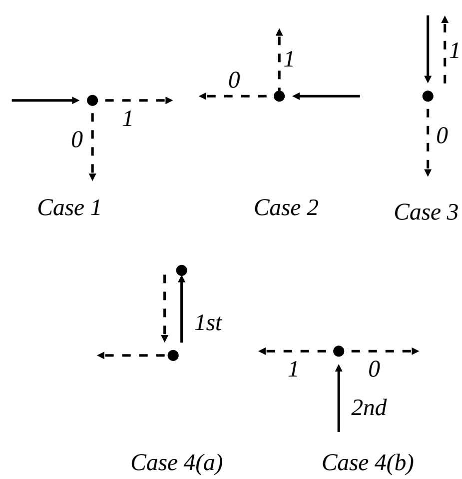


Fig. 8. The code for the DFS.

4 Compact Code II

In this section,we give our second encoding for L-floorplan using the $(4f - 4)$ bits encoding in [TFI09] for ordinary floorplan. Given a floorplan R, append a dummy edge into each L-shape face as shown in Fig. 9. Let

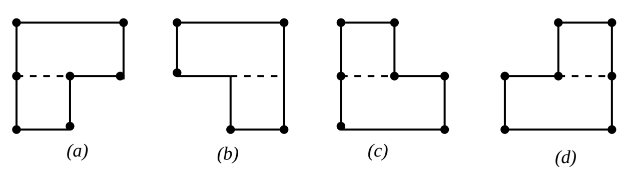
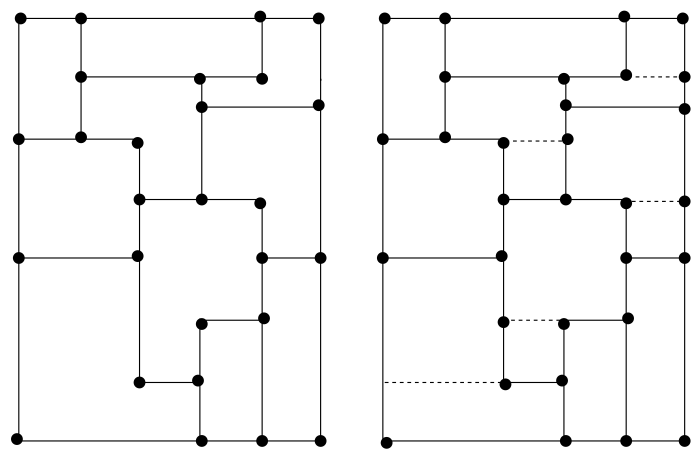


Fig. 9. Transformation of a L shape face to a rectangular face.

R' be the derived floorplan. R' is an ordinary floorplan with $f + L$ rectangles as inner faces. See an example in Fig. 10. We encode R' into a bit string B using the method in [TFI09]. The length of B is $4(f+L)-4$ bits. To reconstruct the original L-floorplan from B, we need to encode some information to indicate inserted dummy edges. So, we encode whether each rectangle of R' is corresponds to the lower part of some L-shape face in R, or not into $f+L$ bits, and also encode, for each lower part of L-shape face F having two or more upper neighbors in R', the upper part of F is whether the leftmost upper neighbor, as illustrated in 9 (c) or the rightmost upper neighbor, as illustrated in in Fig. 9(d), into L bits.



(a) (b)

Fig. 10. (a) An L-floorplan R, and (b) insertion of dummy edges

Note that here we use a natural ordering of faces defined in [TFI09]. Thus we require $4(f+L)-4+(f+L)+L = 5f+6L-4$ bits for encoding and hence the following theorem holds.

Theorem 2. *There is an encoding of L-floorplan with $5f+6L-4$ bits.*

The encoding and decoding time is linear with a suitable data structure.

5 Conclusion

In this paper, we gave two simple compact codings for L-floorplans. The codings need only $6f+3L+2n_2-2$ or $5f+6L-4$ bits for each L-floorplan. Code I is more compact for large L, and code II is more compact for small L . The running time for encoding and decoding is $O(f)$ or $O(n)$.

The number of floorplans where each face is a rectangle is $\Omega(11.56^f) = \Omega(2^{3.53f})$ [ANY07]. So, we need at least $3.53f+c$ bits to encode a floorplan for some constant c. Can we have a similar result for L-floorplans?

References

[ANY07] K. Amano, S. Nakano, and K. Yamanaka, *On the number of rectangular drawings: Exact counting and lower and upper bounds*, IPSJ SIG Notes, AL-115(5):3340, 2007.

[CGHKL98] R. C. -N. Chuang, A. Garg, X. He, M.-Y.Kao, and H.-I Lu, *Compact encodings of planar graphs via canonical orderings and multiple parentheses*, Proc. 25th International Colloquium on Automata, Languages, and Programming, LNCS 1443, pp. 118-129, 1998.

[CLL01] Y. T. Chiang, C.-C. Lin, and H.-I Lu, *Orderly spanning trees with applications to graph encoding and graph drawing*, Proc. 12th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 506-515, 2001.

[GKP94] R. L. Graham, D.E. Knuth, and O. Patashnik. *Exercise 9.60 in Concrete Mathematics*, 2nd ed., Addison-Wesley, 1994.

[KH97] G. Kant and X. He, *Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems*, Thero. Comput. Sci,., Vol.172, pp.175-193, 1997.

[KK84] K. Kozminiski and E. Kinnen,*An algorithm for finding a rectangular dual of a planar graph for use in area planning for VLSI integrated circuits*, Proc. 21st DAC, pp.655-656, Albuquerque, 1984.

[KW89] K. Keeler and J. Westbrook, *Short encodings of planar graphs and maps*, Discrete Appl. Math., vol. 58, no.3, pp. 239-252, 1995.

[MR97] J. I. Munro and V. Raman, *Succinct representations of balanced parentheses, static trees and planar graphs*, Proc. 38th IEEE symposium on Foundations of Computer Science, pp. 118-126, 1997.

[Nak02] S. Nakano, *Enumerating floorplans with n rooms*, IEICE Trans. Fundamentals, vol.E85-A, no.7, pp.1746-1750, July 2002.

[PY85] C. Papadimitriou and M. Yannakakis, *A note on succinct representations of graphs*, Inf. Comput., vol. 71, pp. 181-185, 1985.

[R00] K. H. Rosen, eds. *Handbook of discrete and combinatorial mathematics*, CRC press, Boca Raton, 2000.

[RMN09] M. S. Rahman, K. Miura, T. Nishizeki, *Octagonal drawings of plane graphs with prescribed face areas*, Computational Geometry: Theory and Applications, 42(3), pp. 214-130, 2009.

[RNG04] M. S. Rahman, T. Nishizeki, and S. Ghosh, *Rectangular drawings of planar graphs*, J. Algorithms, vol.50, pp.62-78, 2004.

[S03] J. P. Spinard, *Efficient graph representations*, AMS, 2003.

[T84] G. Turan, *Succinct representations of graphs*, Discrete Appl. Math. vol. 8, pp. 289-294, 1984.

[TFI09] T. Takahashi, R. Fujimaki,Y. Inoue, *A (4n-4) bit representations of rectangular drawing or floorplan*, Proc. of COCOON 2009, LNCS, 5609, pp.4755, 2009.

[YN06] K. Yamanaka and S. Nakano, *Coding floorplans with fewer bits*, IEICE Trans. Fundamentals, vol.E89-A, no.5, pp.1181-1185, May 2006.

