RA-001

Two Compact Codes for Rectangular Drawings with Degree Four Vertices

Masashi Saito † Shin-ichi Nakano †

SUMMARY A rectangular drawing is a partition of a rectangle into a set of rectangles. Rectangular drawings have many important applications including VLSI layout. Since the size of rectangular drawings may be huge, compact encodings are desired. Several compact encodings of rectangular drawings without degree four vertices are known.

In this paper, we design two compact encodings for rectangular drawings with degree four vertices. We give $2m = 6f - 2n_4 + 6$ bits and $5f - 5$ bits encodings for rectangular drawings, where m is the number of edges, f is the number of faces, and n_4 is the number of vertices with degree four.

1 Introduction

Compact encodings of graphs are 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 orderd tree is as follows. Given an orderd 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 n is the number of vertices and m is the number of edges in T. Some examples are shown in Fig. 1.

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].

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

For example, the number of ordered trees with four verteces 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.

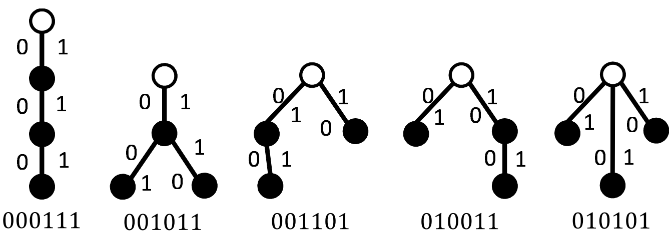

Fig. 1: A code for ordered trees.

A rectangular drawing is a partition of a rectangle into a set of rectangles. Rectangular drawings have many important applications including VLSI layout [KH97, KK84, RNG04]. Since the size of rectangular drawings may

†Department of Computer Science, Gunma Univercity

be huge, compact encodings are desired. Several compact encoding of rectangular drawings without degree four vertices are known. See [CGHKL98, CLL01, KH97, KK84, KW89, MR97, TFI09, YN06].

In this paper, we design two compact encodings for rectangular drawings with degree four vertices. We give $6f - 2n_4 + 6$ bits and $5f - 5$ bits encoding for rectangular drawings, where f is the number of faces and n_4 is the number of vertices with degree four. Note that we can not treat rectangular drawings simply as plane graphs. See two rectangular drawings in Fig. 2. They are identical as plane graphs, however different as rectangular drawings. Because in Fig. 2(a) the two faces F_a and F_b share a horizontal line, however in Fig. 2(b) they share a vertical line. We need to store the direction (horizontal or vertical) for each edge in a given rectangular drawing.

The rest of the paper is organized as follows. Section 2 gives some definitions. Section 3 explains our first encoding of rectangular drawing using $6f - 2n_4 + 6$ bits. Section 4 explains our second encoding of rectangular drawing using $5f - 5$ bits. Using the $(4f - 4)$ bit encoding in [TFI09]. Finally section 5 is a conclusion.

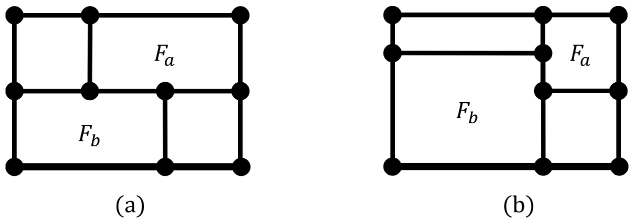

Fig. 2: Two rectangular drawings corresponding to the same plane graph.

2 Preliminaries

In this section we give some definitions. A *tree* is a connected graph with no cycle. A *rooted* tree is a tree with one vertex chosen as its root. *An ordered tree* is a rooted tree in which the set of children of each vertex is assigned a total order.

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 *rectangular drawing* is a partition of a rectangle into a set of rectangles.
Let n_d be the number of vertices with degree d. Now $n = 4 + n_3 + n_4$ holds, since every rectangular drawing has exactly four vertices with degree 2 at the four corner of the outer face. Thus $2 \cdot 4 + 3 \cdot n_3 + 4 \cdot n_4 = 2m$ holds. This equation and the Eular's formula $n - m + f = 1$ gives $2m = 6f - 2n_4 + 6$.

3 Compact code I

In this section we give our first compact encoding for rectangular drawings, based on the depth first search of an ordered tree. The encoding needs $2m = 6f - 2n_4 + 6$ bits for each rectangular drawing.

Given rectangular drawing R, we first append two vertical dummy edges at the lower left corner and the lower right corner of the outer face as shown in Fig. 3. Then, we replace the lower right corner of each face as shown in Fig. 4. See a complete example in Fig. 5.

Lemma 3.1 *The resulting graph is a tree.*

Fig. 3: Insertion of dummy edges at the lower left and lower right corner.

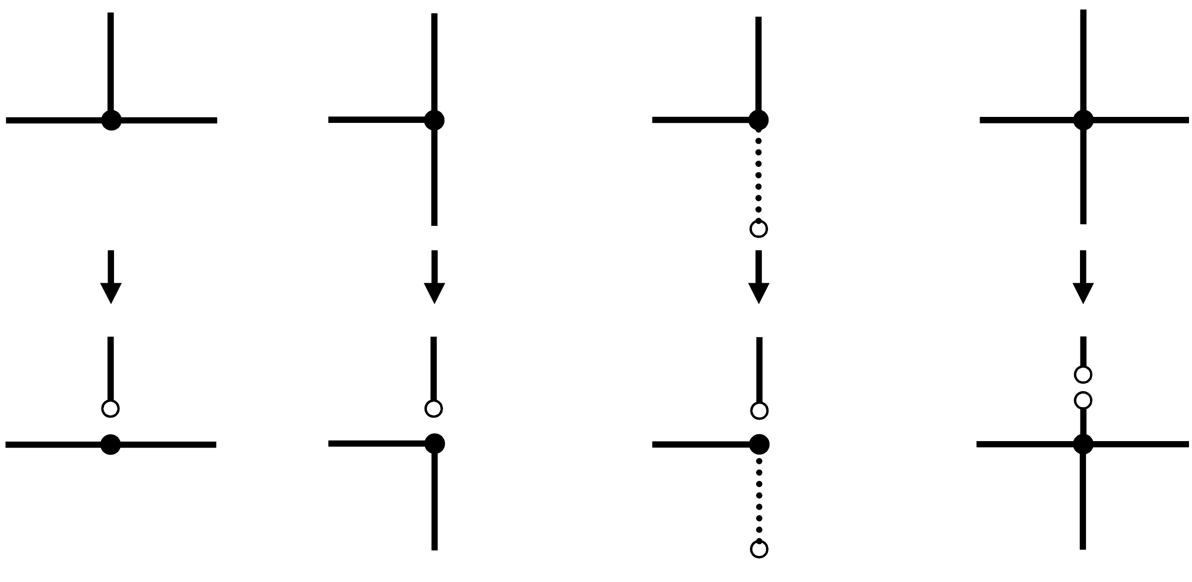

Fig. 4: Replacement of the lower right corner of each face.

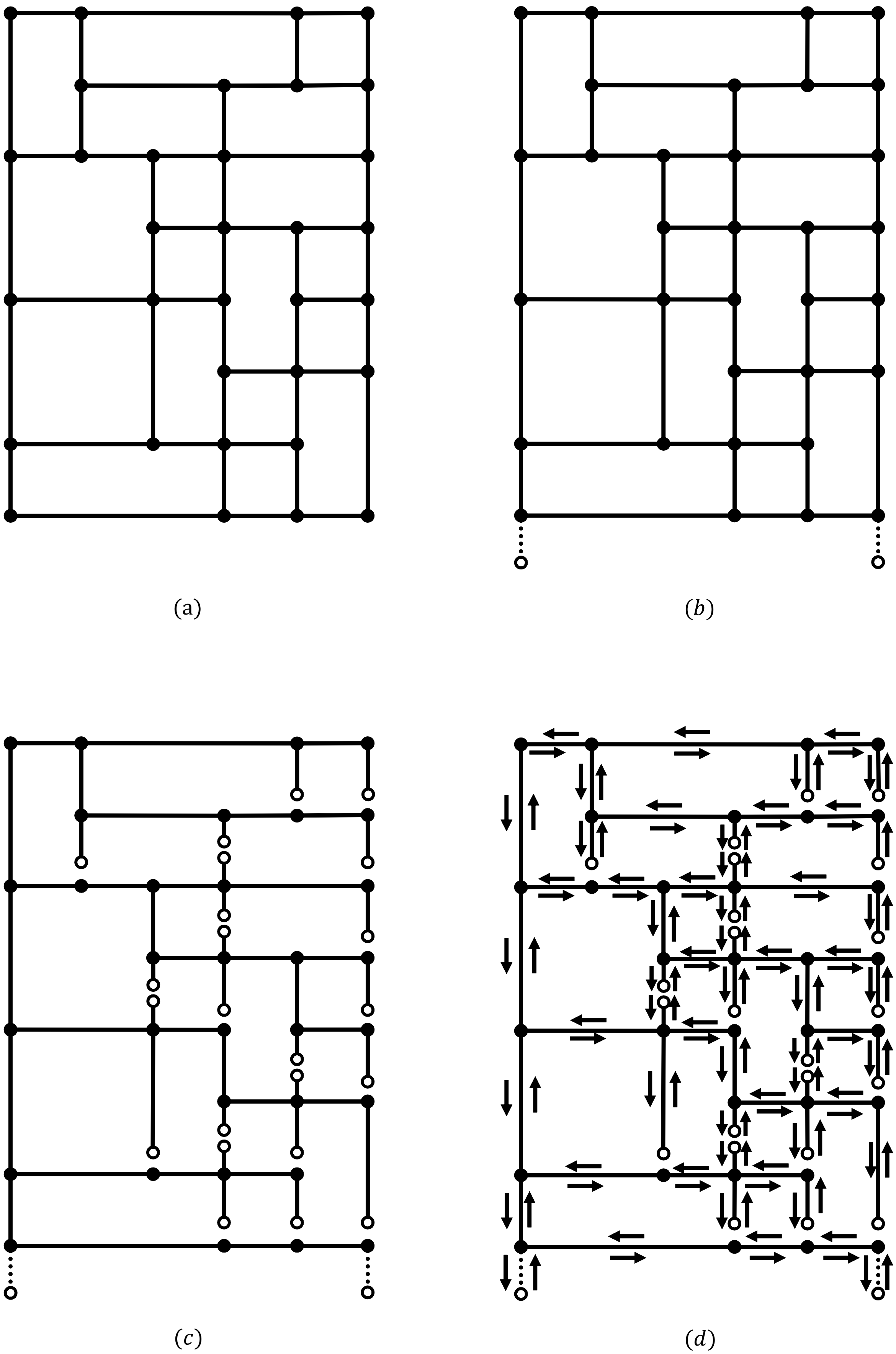

Fig. 5: (a)Given rectangular drawing R, (b)insertion of dummy edges, (c)replacement of lower right corners, (d)the trace.

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.

Q.E.D.

Let R' be the resulting tree. Starting at the upper left corner of the other face, we traverse the tree R' with depth first manner (with right priority). See Fig. 5(d). Each edge is traced exactly twice in both directions. When we arrive at a vertex by tracing an edge, we always have only "two" choices for the next direction to trace, as shown in Fig. 6, 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 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. Case $4(a)$ occurs only if vertex u has degree four and vertex v has degree one, and Case $4(b)$ occurs only if vertex v has degree either two, three, or four.

Since we trace each edge exactly twice, we need two bits for each edge. However we have a chance to save. Whenever Case 4(a) occurs, the direction of the next two traces are always down then left. Thus above each degree four vertex in R' we can save those traces.

Now we have the following theorem.

Theorem 3.2 *There is an encoding of rectangular drawings with degree four vertices using* $6f - 2n_4 + 6$ *bits.*

Proof. Since Case $4(a)$ occurs at each degree four vertices, R' has $m + 2 + n_4$ edges. Thus the length of the code is $2(m + 2 + n_4) - 2n_4 = 2m + 4 = 6f - 2n_4 + 6$ bits.

Q.E.D.

The encoding time is linear. Given the $6f - 2n_4 + 6$ bits code we can easily reconstruct the original rectangular drawing R by a simple linear time algorithm with a stack.

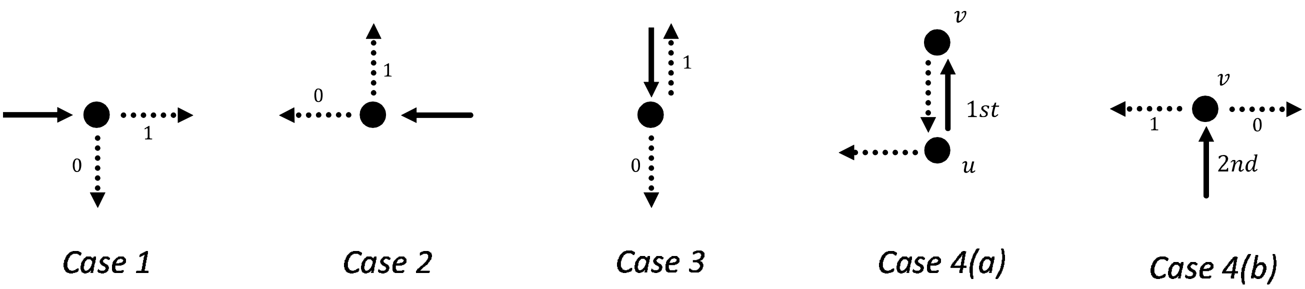

Fig. 6: The choice for the next trace.

4 Compact Code II

In this section, we give our second encoding for rectangular drawing using the $(4f - 4)$ bits encoding in [TFI09] for ordinary rectangular drawing. Given a rectangular drawing R, we replace each degree four vertex as shown in Fig. 7. See a complete example in Fig. 8. Then we encode R into a bit string B using the method in[TFI09]. The length of B is $4f - 4$ bits. To reconstruct the original rectanglar drawing R, we need to append some information to indicate degree four vertices. So, we encode whether the upper right corner of each rectangle has degree four or not, into $f - 1$ bits. Here we use a natural ordering of inner faces defined in [TFI09].

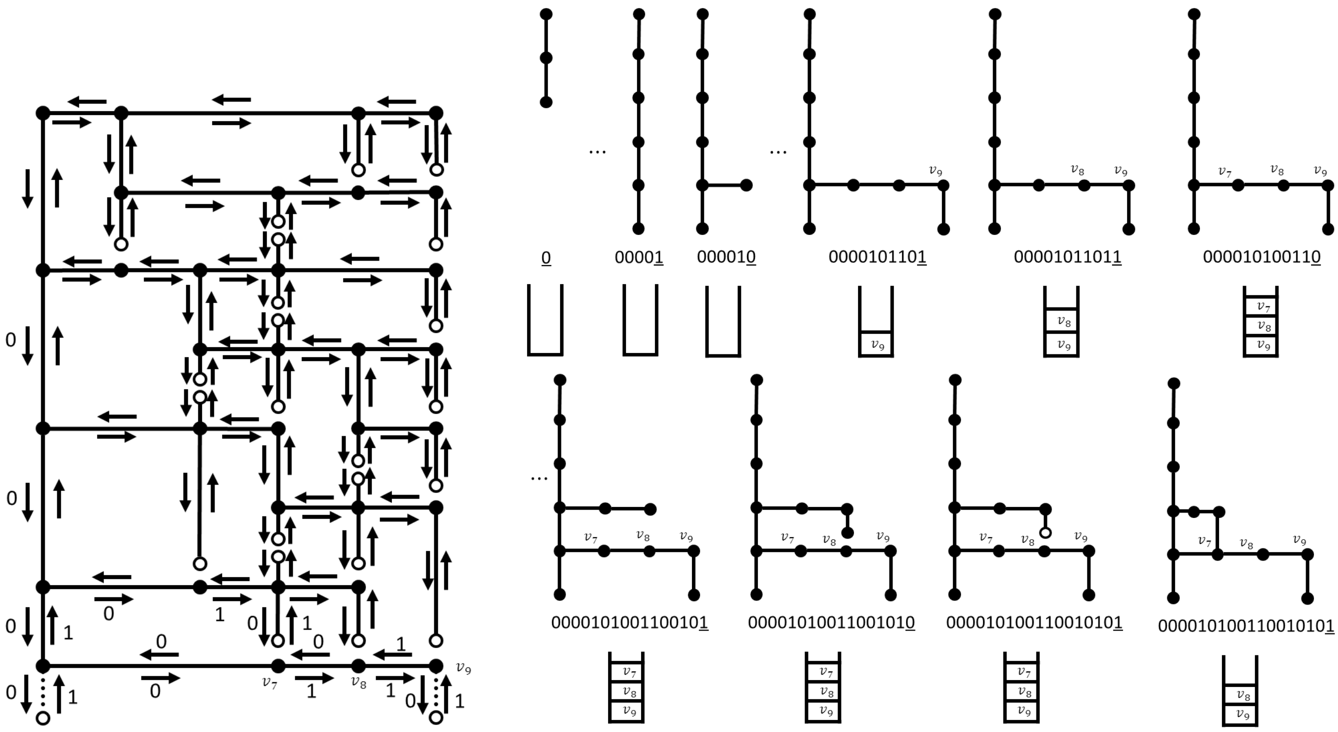

Fig. 7: Decoding of Code I.

We have the following theorem.

Theorem 4.1 *There is an encoding of rectangular drawing with* $5f - 5$ *bits.*

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

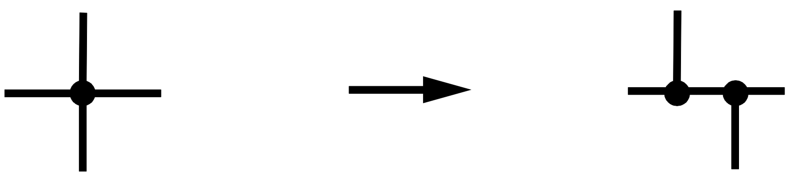

Fig. 8: Replacement of each degree four vertex.

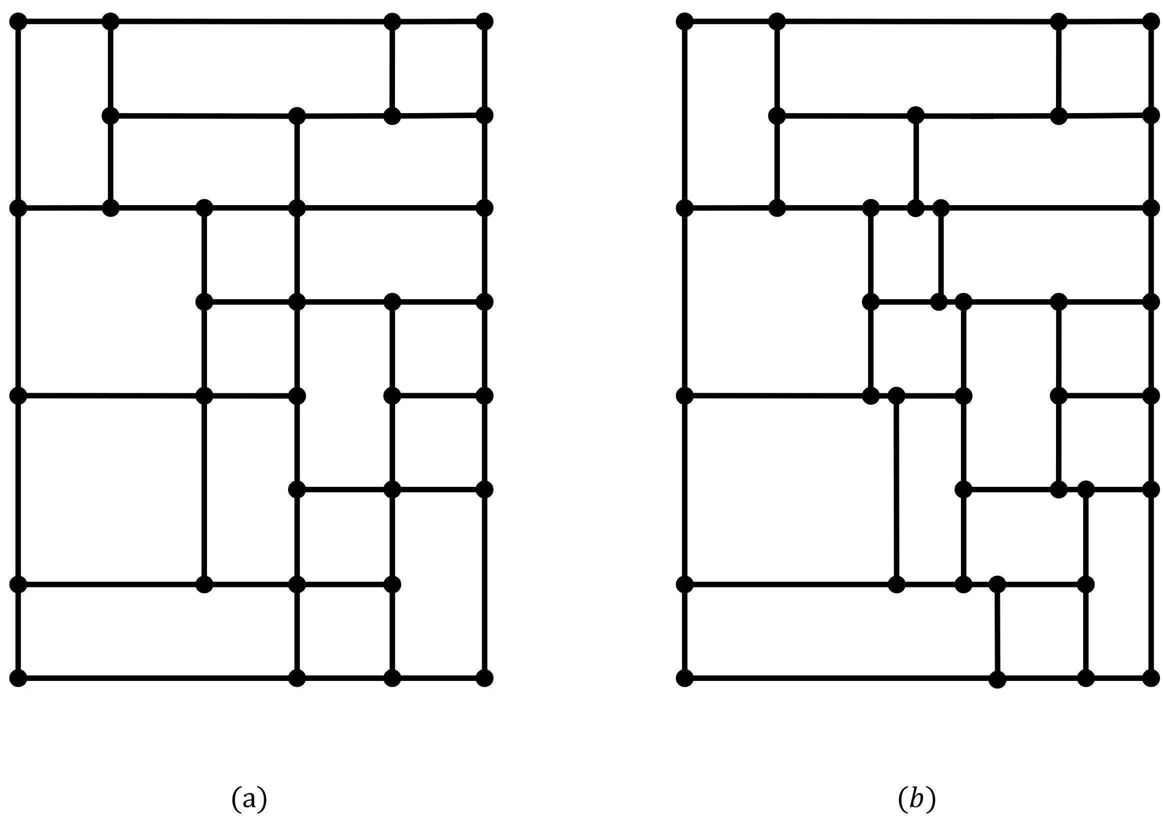

Fig. 9: Replacement of each degree four vertex in rectangular drawing R.

5 Conclusion

In this paper, we gave two simple compact codings for rectangular drawings with degree four vertices. The coding needs only $6f - 2n_4 + 6$ or $5f - 5$ bits for each rectangular drawing. Code I is more compact for large n_4, and code II is more compact for small n_4. Asymptotically if $n_4 > f/2$ then code I is more compact, otherwise code II is more compact. The running time for encoding and decoding is $O(f) = O(n)$.

The number of rectangular drawings with no degree four vertices is $\Omega(11.56^f) = \Omega(2^{3.53f})$ [ANY07]. So we need at least $3.53f + c$ bits to encode a rectangular drawing for some constant c.

References

[ANY07] K. Amano, S. Nakano, and K. Yamanaka,*On the number of rectanglar drawings: Exact couting 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 cannical ordering and multiple parentheses*, Proc. 25th International Colloquium on Automata, Languages, and Programming, LNCS 1443, pp118-129, 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. Patashinik. *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,., Vol172, pp175-193, 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, pp655-656, 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.Minro 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.

[PY85] C. Papadimitriou and M. Yannakakis, A *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 pres, Boca Raton, 2000.

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

[S03] J.P.Spinard, *Succinct 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 respresentations of rectangular drawing of floorplan,* Proc. of COCOON 2009, LNCS, 5609, pp.4455, 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.