The Quarter-State-Sequence Floorplan Representation

Keishi Sakanushi, Yoji Kajitani, *Fellow, IEEE*, and Dinesh P. Mehta

***Abstract*—A floorplan of a bounding box is its dissection into rectangles (*rooms*) by horizontal and vertical segments. This paper proposes a string data structure called the *Quarter-state sequence* (or *Q sequence*) to represent the floorplan. The Q sequence is a concatenation of the *states* of rooms along the *Abe order* and is related to the *VH graph*, which is the union of the vertical-constraint and horizontal-constraint graphs. It is proved that any floorplan of n rooms is uniquely encoded by a Q sequence and any Q sequence is uniquely decoded to a floorplan, both in $O(n)$ time. An exact formula for counting distinct floorplans is given and compared with existing bounds. A linear time transformation of one Q sequence to another is defined. An n-room *packing* algorithm based on simulated annealing was implemented and found to compare favorably with existing packing algorithms.**

***Index Terms*—Floorplan combinatorics, floorplan representation, graph theory, Q-sequence, simulated annealing.**

I. Introduction

A FLOORPLAN of a rectangle (*chip*) is a dissection of the chip into n rectangles (*rooms*) by horizontal and vertical line segments (*segs*) that meet in T junctions (but are not permitted to cross). Floorplanning is important in the design of very large-scale integration (VLSI) systems that use IPs or clustered functional modules because it provides the first estimates of performance and cost. Multiple objectives such as the minimization of the chip area, estimated wire-length, critical-path wire length, length of parallel-running wires, clock skew, etc., must be simultaneously optimized in modern floorplanners. Heuristic search techniques such as simulated annealing appear to be the best strategy for tackling this multi-objective optimization problem. These approaches successively generate and evaluate candidate floorplans to determine the best flooplan encountered so far. In this state space search strategy, floorplans are represented by codes that can be efficiently perturbed to generate new floorplan states. It is desirable that the following hold.

1) Any legal floorplan be represented by a unique code so that optimal floorplans are not missed and so that the same floorplan is not generated multiple times.
2) Any code be decoded to exactly one floorplan so that infeasible codes are not generated and so that codes can be evaluated uniquely.
3) A transformation on the current solution to obtain a new solution (i.e., a move) be efficiently implemented so that few moves are required to transform a solution to an arbitrary solution.

When floorplans are restricted to the slicing structure, they can be coded by the *slicing tree* [1] and the *normalized polish expression* [2]. However, the slicing structure does not capture all floorplans. Sequence pairs (SPs) [3] were used for the packing problem [4], [5]. The bounded-sliceline grid (BSG) [6] is technically a floorplanning data structure as the number of nonzero-area rooms equals the number of modules. The difficulty arises in determining which (of many possible) rooms should have nonzero area. SP, BSG, and the more recent transitive closure graph (TCG) [7] suffer because of redundancies which lead to larger solution spaces. O trees [8] and B^* trees [9] were proposed to represent packings. However, they only provide an incomplete description of the topological information and require knowledge of the module dimensions to complete the description of a packing. Q sequences [10]–[12] and corner block lists [13] are general floorplan representations that were discovered independently in 2000. Subsequently, in 2001, the twin binary tree representation [14] and, in 2002, an equivalent (but more convenient) representation called the twin binary sequence [15] were proposed. The twin binary sequence and the extended corner block list [16] have extended the corresponding floorplan representations so that they can be used effectively for the packing problem.

This paper describes the *Quarter-state sequence* (*Q sequence*) that (along with corner block lists) was the first representation to meet the criteria listed above. A room in a floorplan is characterized by the configuration of one of its four corners (hence the term "quarter"), which is called its *state*. The floorplan is represented by a sequence of states of rooms and their adjacency relations. A key question that arises in coding a floorplan is how rooms are to be ordered in the sequence. This was addressed by Abe [17] (see also [18], [19]) in 1930!

An important graph theoretic contribution of our paper is that the union of the horizontal-constraint and vertical-constraint graphs has a unique Hamiltonian path that starts with the vertex corresponding to a room at a corner and ends with the vertex corresponding to the room at the diagonally opposite corner. The Q-sequence data structure is defined as the concatenation of the configuration of one corner with the states of all the rooms along the Hamiltonian path. Any floorplan can be encoded by a unique Q sequence and any Q sequence is decoded to a unique floorplan in linear time. The Q sequence also makes it possible to have a counting formula for the number of floorplans. Also, a

Manuscript received October 19, 2000; revised November 19, 2001. This work was supported in part by the National Science Foundation under Grant CCR-9988338, in part by the Tokyo Institute of Technology under Project of CAD21, and in part by the Research and Development Aid of Support Center for Advanced Telecommunications Technology Research. This paper was recommended by Associate Editor K. Thulasiraman.

K. Sakanushi is with the Department of Information System Engineering, Osaka University, Osaka 560-8531, Japan (email: sakanusi@ist.osaka-u.ac.jp).

Y. Kajitani is with the Department of Information and Media Science, The University of Kitakyushu, Wakamatsu-ku (email: kajitani@env.kitakyu-u.ac.jp).

D. P. Mehta is with the Mathematical and Computer Sciences Department, Colorado School of Mines, Golden, CO 80401-1887 USA (email: dmehta@mines.edu).

Digital Object Identifier 10.1109/TCSI.2003.809442

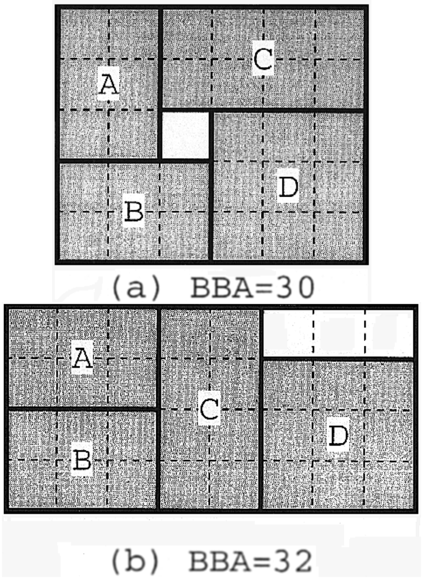


Fig. 1. Area minimized placement. (a) Unconstrained packing (b) Four-room packing (floorplanning).

linear time transformation (move) is defined such that n moves can transform an initial Q sequence to any Q sequence.

Section II clarifies some of the commonly used definitions and assumptions in floorplanning literature. It shows that the floorplanning problem is a constrained version of the general packing problem. The structure of the floorplan is studied using prime graphs in Section III. In Section IV, the Q-sequence data structure is introduced and the decoding algorithm is described. A counting formula is described in Section V. In Section VI, additional important properties of the Q sequence are discussed. In Section VII, a floorplan-based packing algorithm is implemented and experimented with. Even though Q sequences provide a constrained packing method, experiments show that the results are no worse than SPs and BSGs and are obtained much faster. Finally, Section VIII concludes the paper.

II. Basic Concepts

In this section, we clarify the difference between the terms *packing* and *floorplanning*, which are often used interchangeably in the literature.[1] In a packing, the objective is to accommodate the n modules in a bounding box of minimum area. In floorplanning, we wish to compute a bounding box of minimum area that is dissected into *exactly* n rectangular rooms such that each module can be accommodated in a unique room. Fig. 1 illustrates the difference for $n = 4$.

Fig. 1(a) shows an optimal *packing* of modules A, B, C, and D in a bounding box of area 30. However, Fig. 1(a) is not a floorplan, as it dissects the bounding box into $n + 1$ (and not n) rooms, one of which is empty. Fig. 1(b) is an optimal *floorplan* as the chip rectangle is dissected into four rooms, each of which accommodates a module. (Here, there is some whitespace left over in the room containing module D.) The area of the bounding box for the optimal floorplan is 32, which is greater than that of an optimal packing. Thus, floorplanning may be viewed as a constrained version of the unconstrained packing problem (i.e., an n-room packing problem). Consequently, we don't expect a direct application of floorplanning to yield an optimal packing. The Q-sequence representation proposed in this paper, solves the floorplanning problem or the n-room packing problem. In spite of this apparent disadvantage, we will show in Section VII, that it yields good results for the packing problem.

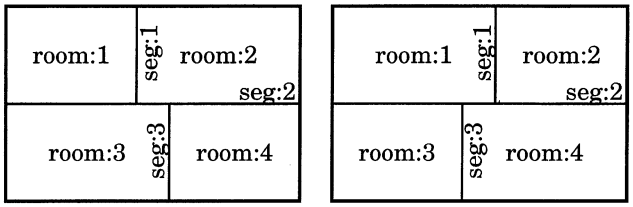


Fig. 2. In both floorplans, rooms 1 and 2 are adjacent to seg 1, and rooms 3 and 4 are adjacent to seg 3, and all four rooms are adjacent to seg 2. Thus, the two floorplans are equivalent under the room–seg adjacency relation. However, rooms 2 and 3 (or 1 and 4) are adjacent to each other in one floorplan, but not in the other. Thus, the two floorplans are not equivalent under the room–room adjacency relation.

A. Adjacency Relations in Floorplan Representations

In this subsection, we explicitly differentiate between two types of adjacency relations that are used in floorplan representations. In a *room–seg* adjacency relation, a relation is defined between a room r and a segment s if and only if s is the left, right, top, or bottom border of r. A *room–room* adjacency relation is defined between two rooms r and s if and only if they share a nonzero portion of a segment. Fig. 2 uses two floorplans to illustrate the (subtle) difference between these two types of relations. Notice that both floorplans are equivalent under the room–seg adjacency relation, but not under the room–room adjacency relation.

The room–room adjacency relation was used in initial efforts in the 1970s and 1980s to cast the floorplanning problem in terms of graph duals [20]–[22]. The input to the floorplanning problem is a graph (representing a circuit). If a pair of vertices is joined by an edge, then the corresponding rooms in the floorplan are required to be adjacent. It was shown that a graph admits a floorplan that satisfies this property only if the graph is planar triangulated and does not contain any complex triangles. However, this was not a practical formulation as the input graph was not necessarily planar and the room–room adjacency requirement was an overconstraint. Further, no effective optimization techniques were developed to minimize the number of graph edges that could not meet the adjacency requirement. The room–seg relation was made explicit in 1994 after the BSG, and SP representations were introduced. The room–seg adjacency relation is used as the constraint imposed on the one-dimensional compaction. This paper also uses the room–seg adjacency relation.

III. Prime Graph of a Floorplan

A floorplan with n rooms consists of $n-1$ internal segs (maximal vertical or horizontal line segments). A vertical seg and a horizontal seg are not permitted to cross one other, but can touch

[1] The term "floorplan" is often used in the literature to describe the concept that we refer to as "packing" and the term "mosaic" floorplan is used to refer to the concept that we call a "floorplan."

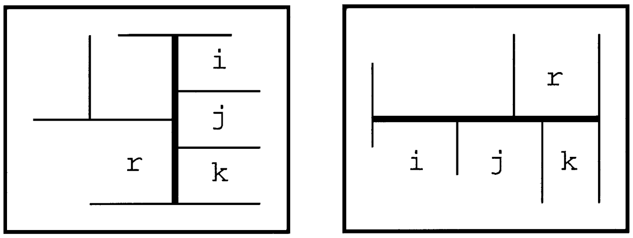


Fig. 3. In both floorplans, the prime seg of r is the bold line, and the associated rooms are i, j, and k. i is the next room of r.

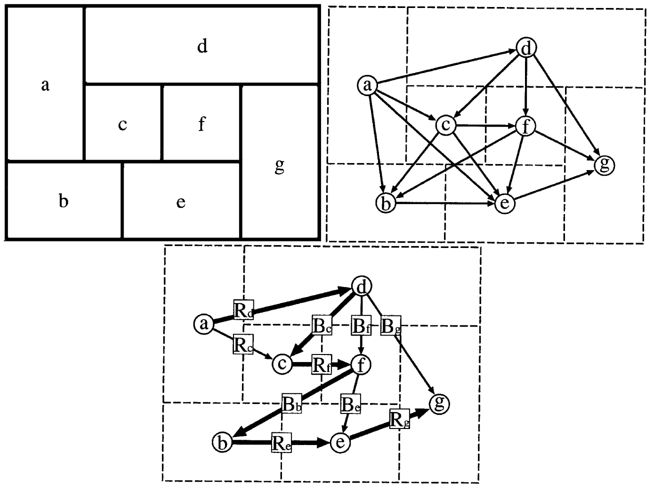


Fig. 4. Floorplan of seven rooms, its VH graph, and pVH graph. Henceforth, the vertex label in our figures also denotes the name of the corresponding room.

to form a T junction. The four bounding segments of a chip are also segs, but are called *walls* if it is necessary to distinguish them from internal segs. In the following, we will focus on the right-bottom and left-top corners of the chip. If the right seg and bottom seg of a room r are both walls, it is the right-bottom room and is denoted by rb. (The room whose left seg and top seg are both walls is the room at the left top, which we name lt.) For any room r except rb, two segs meet at the right-bottom corner of r and one ends there forming a T junction. The seg that ends at the T junction is called the *prime seg* of r. (Note that rb does not have a prime seg.)

The rooms that are adjacent to r's prime seg on the opposite side of r are called the *associated rooms* of r. The topmost (leftmost) of the associated rooms is called the *next room* of r. See Fig. 3 for an illustration of these definitions.

Definition VH graph $VH(F)$: Each vertex in $VH(F)$ corresponds to a room in floorplan F. For a room r, v_r denotes the corresponding vertex in $VH(F)$ and for a vertex v in $VH(F)$, r_v denotes the corresponding room. For two rooms x and y, a directed edge (v_x, v_y) exists in $VH(F)$ if and only if there is either a vertical seg s such that s is the right border of x and the left border of y, or a horizontal segment t such that t is the bottom border of x and the top border of y. Fig. 4 illustrates the definition.□

For a floorplan F of n rooms, let $VH'(F)$ be the graph obtained from $VH(F)$ by deleting vertex v_{lt} and the edges incident to it. Let F' be the floorplan obtained from F by *shifting* the prime seg s_{lt} of room lt to the left (if s_{lt} is vertical) or to the top (if s_{lt} is horizontal) until it merges with the wall. F' consists of $n-1$ rooms. Its left-top room was the next room of lt before the prime seg was shifted.

Lemma 1: **(Deletion)** $VH'(F) = VH(F')$.

Proof: The edges of the VH graph correspond to the adjacency relations between rooms that are on opposite sides of a seg. Shifting lt's prime seg does not change any of these relationships, except for those involving lt. These are precisely the edges that are eliminated in $VH'(F)$. ■

Lemma 2: **(DAG)** $VH(F)$ is a directed acyclic graph (DAG) with a single source (vertex with no incoming edge) v_{lt} and a single sink (vertex with no outgoing edge) v_{rb}.

Proof: There is no incoming edge to a vertex v if and only if the left border and top border of r_v are both walls. Room lt is the only room that satisfies this condition. Hence, v_{lt} is the only source in $VH(F)$. A similar argument shows that vertex v_{rb} is the only sink. If $n = 1$, $VH(F)$ is trivially a DAG. Assume that any floorplan with $n - 1$ or less rooms has a VH graph that is a DAG. It is a well-known fact that a directed graph is a DAG if and only if it contains at least one source and the graph obtained from the graph by deleting the source is a DAG. Since a VH graph contains one source and $VH'(F)$ is a VH graph of a floorplan of $n - 1$ rooms (Lemma 1), we have proved by induction that $VH(F)$ is a DAG. ■

For a room r, let $\mathrm{next}(r)$ denote its next room and $\mathrm{prev}(r)$ the previous room; i.e. $\mathrm{prev}(\mathrm{next}(r)) = r$. $\mathrm{next}(r)$ is unique unless $r = rb$ and so is $\mathrm{prev}(r)$ unless $r = lt$.

Lemma 3: **(Hamiltonian Path)** $VH(G)$ has a unique directed Hamiltonian path π whose vertex sequence corresponds to the room sequence: lt, $\mathrm{next}(lt)$, $\mathrm{next}(\mathrm{next}(lt)), \ldots, \mathrm{next}^k(lt), \ldots, \mathrm{next}^{n-1}(lt) = rb$.

Proof: Since there is an edge $(v_r, v_{\mathrm{next}(r)})$ for every room $r(\neq\ rb)$, the vertex sequence corresponding to $\pi(k) = (lt, \mathrm{next}(lt), \mathrm{next}(\mathrm{next}(lt)), \ldots, \mathrm{next}^k(lt))$ is a path. Since $VH(F)$ is a DAG, no vertex appears twice in this path. Then, $\pi(k)$ can be extended until the unique sink v_{rb} is encountered. If a vertex v is not contained in $\pi(k)$, then consider the path of *next*s starting with v. This path must encounter a vertex v_q on $\pi(k)$ since v_{rb} is the only sink. This means that $prev(q)$ is not unique, a contradiction. ■

Next, we define the prime graph which is a subgraph of $VH(F)$.

Definition Prime Graph $pVH(F)$: The vertex set is the same as that of $VH(F)$. An edge $e = (v_p, v_q)$ exists if and only if q is an associated room of p. In other words, the edges of $pVH(F)$ are defined only by the rooms and their prime segs. An edge of $pVH(F)$ is called $\mathcal{B}$ edge or $\mathcal{R}$ edge depending on whether the prime seg s_p is horizontal or vertical, respectively. Letters $\mathcal{B}$ and $\mathcal{R}$ are called the positional symbols and stand for "Below" and "Right," respectively. The definition is illustrated by Fig. 4.□

Theorem 1: $pVH(F)$ has a unique Hamiltonian path starting with v_{lt} and ending at v_{rb}.

Proof: From Lemma 3, the Hamiltonian path π in $VH(F)$ consists of edges $(v_u, v_{\mathrm{next}(u)})$ which are defined by the rooms and their prime segs.These edges are retained in $pVH(F)$. ■

Since the Hamiltonian path π is unique, each room can be assigned an integer label corresponding to its position in π:

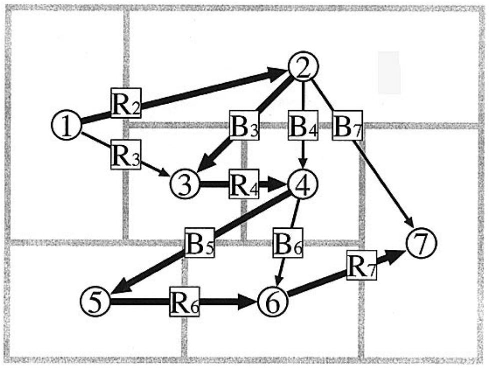


Fig. 5. pVH graph of Fig. 4 normalized by the Abe order.

$1(=lt), 2, 3, \ldots, n(=rb)$. The existence of a unique ordering is fundamental since it permits us to determine whether or not two floorplans are structurally identical. This discovery dates back to a 1930 paper by Michio Abe that is concerned with counting distinct floorplans [17]. His contribution was to order the segs. We extended this to order rooms. We refer to this ordering of rooms and segs, which is illustrated in Fig. 5, as the Abe ordering.

$pHV'(F)$ denotes the graph obtained from $pVH(F)$ by deleting v_{lt} and its edges.

Lemma 4: **(Deletion in pVH)** $pVH(F') = pVH'(F)$. In general, the graph obtained from $pVH(F)$ by deleting the first $p-1$ vertices $v_1, v_2, \ldots, v_{p-1}$ is the prime graph of the floorplan which is obtained by $p-1$ shifts of the prime segs of the left-top room.

Proof: A proof identical to that of Lemma 1 shows that $pVH(F') = pVH'(F)$. The second claim is proved by showing that shifting eliminates the current left-top room and makes its next room the new left-top room. ■

A room whose top border (left border) is part of the top wall (left wall) is called the top (left) wall room.

Theorem 2: **(IN EDGE)** The number of incoming $\mathcal{B}$ edges ($\mathcal{R}$ edges) to any vertex v of $pVH(F)$ is either zero or one.

Proof: A $\mathcal{B}$ edge (u, v) exists if and only if the prime seg L of r_u is horizontal and r_v is an associated room. For a given r_v, such a room r_u exists only if r_v is not a top wall room. If it exists, r_u is the (unique) room whose prime seg is L. A similar discussion holds for $\mathcal{R}$ edges. ■

Theorem 3: **(OUT EDGE)** For each vertex v of $pVH(F)$ (except v_{rb}), the outgoing edges are all exclusively $\mathcal{B}$ edges or $\mathcal{R}$ edges.

Proof: The outgoing edges from a vertex v correspond to the associated rooms of r_v. They are all $\mathcal{B}$ edges or $\mathcal{R}$ edges depending on whether r_v's prime seg is horizontal or vertical, respectively. ■

Let $e = (u, v)$ be a $\mathcal{B}$ edge ($\mathcal{R}$ edge). The set of vertices $u+1$ through v in the Hamiltonian path π is called the $\mathcal{B}$ span ($\mathcal{R}$ span) of e.

Lemma 5: **(Wall Room)** A vertex v is not in any $\mathcal{B}$ span ($\mathcal{R}$ span) if and only if room r_v is a top (left) wall room.

Proof: Assume that v is in $\mathcal{B}$ span by $e = (p, q)$. It implies that room r_{p+1} and r_q are associated rooms of room r_p. Then, their top borders are at the same height as the prime seg s_p of r_p. Top borders of rooms r_{p+1} through r_q are not above s_p since otherwise there is a vertical seg that terminates L by forming a T junction to the left of r_q. Then, r_q cannot be an associated room, a contradiction. Thus, all of these rooms, one of which is r_v, are not top (left) wall rooms.

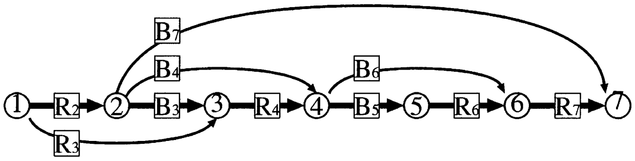


Fig. 6. Horizontal linear planar embedding of the $pVH(F)$ in Fig. 5.

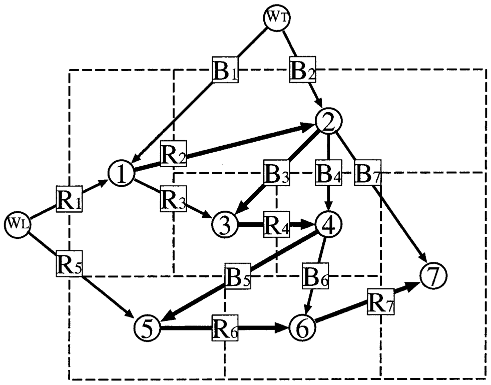


Fig. 7. The epVH graph corresponding to the pVH graph of Fig. 5.

Assume next that v is not in any $\mathcal{B}$ span ($\mathcal{R}$ span). Then, it has no incoming $\mathcal{B}$ edge ($\mathcal{R}$ edge) since incoming edge (x, v) contains v in its span by definition. This implies that there is no room above r_v, i.e., room r_v is a top wall room. ■

Theorem 4: **(SPAN)** If a vertex is in a $\mathcal{B}$ span ($\mathcal{R}$ span), it has an incoming $\mathcal{B}$ edge ($\mathcal{R}$ edge).

Proof: From Lemma 5, a vertex v in a $\mathcal{B}$ span does not correspond to a top wall room. It must be an associated room of the room above it, whose prime segment is r_v's top border. So, v has an incoming $\mathcal{B}$ edge. The discussion for $\mathcal{R}$ span is similar.■

Theorem 5: **(LPD)** Graph $pVH(F)$ has a planar embedding the linear-planar drawing (LPD) such that the Hamiltonian path π is drawn on a straight horizontal line and every $\mathcal{B}$ edge not in π is above and every $\mathcal{R}$ edge not in π is below it.

Proof: Clearly, $pVH(F)$ can be drawn as specified. We will show that this drawing is planar. Assume that two $\mathcal{B}$ edges (a, b) and (c, d) cross each other. Let $a < c < b < d$ without loss of generality. Vertex b has a single incoming $\mathcal{B}$ edge from vertex a (Theorem 2). Let $pVH'(F)$ be the graph obtained from $pVH(F)$ by deleting vertices $1, 2, \ldots, c-1$, which is also a pVH graph of some floorplan (Lemma 4). Here, vertex b belongs to the $\mathcal{B}$ span of edge (c, d) but does not have an incoming $\mathcal{B}$ edge because vertex a was deleted. This contradicts Theorem 4, proving that there is no pair of crossing $\mathcal{B}$ edges. Similarly, it can be shown that $\mathcal{R}$ edges do not cross. ■

Fig. 6 shows an LPD of the prime graph in Fig. 5. To characterize the prime graph, we define a class of graphs that contain all prime graphs. An H graph is a directed graph whose vertex set consists of n vertices labeled by integer subscripts as $\pi = v_1, v_2, \ldots, v_n$. Edges are of two kinds: $\mathcal{R}$ edges and $\mathcal{B}$ edges. There is a Hamiltonian path along the vertex sequence

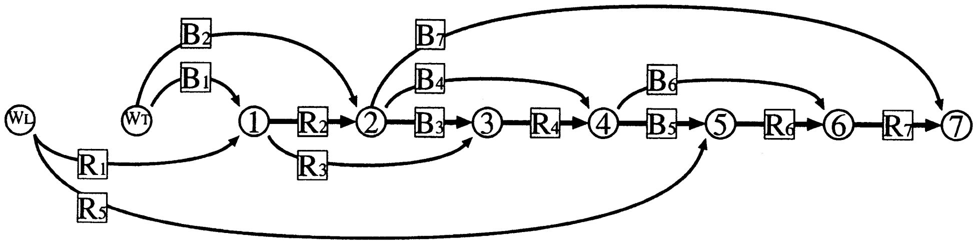


Fig. 8. Horizontal linear planar drawing of the $epVH(F)$ in Fig. 7.

$(v_1, v_2, \ldots, v_n)$. The prime graph is an H graph but an H graph may not be a prime graph. Now we have the main theorem.

Theorem 6: **(Feasibility)** An H graph H of n vertices is $pVH(F)$ of some floorplan F of n rooms if and only if H satisfies the following four properties.

LPD: H has a planar embedding such that the Hamiltonian path π is drawn on a straight horizontal line and every $\mathcal{B}$ edge not in π is above and every $\mathcal{R}$ edge not in π is below it.

SPAN: If a vertex is contained in a $\mathcal{B}$ span ($\mathcal{R}$ span), it has an incoming $\mathcal{B}$ edge ($\mathcal{R}$ edge).

IN EDGE: The number of incoming $\mathcal{B}$ edges ($\mathcal{R}$ edges) to a vertex is 0 or 1.

OUT EDGE: The outgoing edges from a vertex are exclusively $\mathcal{B}$ edges or $\mathcal{R}$ edges.

Proof: We only have to prove sufficiency; i.e., that an H graph that satisfies these conditions (a feasible H graph) corresponds to a floorplan. Assume that there is a floorplan for any feasible H graph if the number of vertices is less than n. Let H be a feasible H graph of n vertices. H' is the graph obtained from H by deleting vertex v_1 and its incident edges. H' trivially satisfies LPD, IN EDGE, and OUT EDGE since H satisfies those properties. Assume that SPAN is not satisfied, i.e. there is a vertex v in H' which is in a $\mathcal{B}$ span of edge e but has no incoming edge. Then, from Theorem 4, there is an incoming $\mathcal{B}$ edge f from v_1 to v since H satisfies SPAN. Then, e and f cross each other, a contradiction to LPD. So H' is feasible and, from the induction hypothesis, it has a corresponding floorplan F' whose rooms are labeled with $2, \ldots, n$. In H', a vertex that has no incoming $\mathcal{B}$ edge is a top wall room by Lemma 5. (The case when there is no incoming $\mathcal{R}$ edge can be discussed analogously.) Moreover, the top wall is a sequence of top borders of these rooms ordered by increasing labels from the left. The outgoing edges from vertex v_1 are exclusively $\mathcal{B}$ edges or $\mathcal{R}$ edges (by OUT EDGE). We assume $\mathcal{B}$ edges without loss of generality. A vertex that has no incoming $\mathcal{B}$ edge in H' is called a $\mathcal{B}$ candidate, meaning that only such a vertex has the possibility to have an incoming $\mathcal{B}$ edge from v_1 (by IN EDGE). Let m be the maximum room label among all of the $\mathcal{B}$-candidates that have an incoming $\mathcal{B}$ edge from v_1. Then, every $\mathcal{B}$ candidate $v_x (x \leq m)$ has an incoming $\mathcal{B}$ edge from v_1 (by SPAN). The sets of these vertices and corresponding rooms are denoted as P and r_P, respectively. Since vertices in P are not in any $\mathcal{B}$ span in H' (otherwise LPD is violated), their top borders form a continuous left part of the top wall of F'. Place room 1 so that its bottom border is P. Then, by shifting the top borders of rooms in r_P downwards by the same distance, room 1 can be inserted to get a new rectangular floorplan. Since the relation with respect to room 1 is correctly realized and no relations in $pVH(F')$ are changed, the new floorplan is F with $pVH(F)$. ■

The prime graph and its encoding to a string (the Q sequence) are used in a practical solution to the floorplanning problem. However, the prime graph is asymmetric in that an edge $\mathcal{R}_k(\mathcal{B}_k)$ sometimes does not appear in the graph because room k borders a left (top) wall. This asymmetry makes it difficult to derive combinatorial results about Q sequences. This is remedied by defining the extended prime graph.

Definition Extended Prime Graph $epVH(F)$: This is obtained from the prime graph $pVH(F)$ by adding two vertices W_L and W_T, and edges (W_L, v_r) for every room r that is adjacent to the left wall and (W_T, v_r) for every room that is adjacent to the top wall.□

See Fig. 7 and its horizontal linear planar embedding (Fig. 8).

A Z graph is a directed graph whose vertex set consists of n vertices labeled by integer subscripts as $v_1, v_2, \ldots v_n$ and two vertices W_L and W_T. Edges are of two kinds: $\mathcal{R}$ edges and $\mathcal{B}$ edges. There is a path along the vertex sequence $\pi_N = (v_1, v_2, \ldots, v_n)$. An epVH graph is a Z graph but a Z graph may not be an epVH graph.

Theorem 7: A Z graph Z is $epVH(F)$ for a floorplan F of n rooms if and only if Z satisfies the following properties:

LPD: Z has a planar embedding such that W_L, W_T, and the path π_N are drawn on a straight horizontal line and every $\mathcal{B}$ edge not in π_N is above and every $\mathcal{R}$ edge not in π_N is below it.

OUT EDGE: The outgoing edges from a vertex are exclusively $\mathcal{B}$ edges or $\mathcal{R}$ edges. W_L has only outgoing $\mathcal{R}$ edges and W_T has only outgoing $\mathcal{B}$ edges.

IN EDGE: Every vertex v_k has exactly two incoming edges: one $\mathcal{R}_k$ and one $\mathcal{B}_k$. W_L and W_T do not have incoming edges.

Proof: The proof is similar to that of Theorem 6. In the inductive step, we delete v_1 from the graph as before, but instead of deleting its outgoing edges, we modify them so that they go out from W_L if they are $\mathcal{R}$ edges and from W_T if they are $\mathcal{B}$ edges. The resulting Z graph clearly satisfies all of the properties specified in the theorem and, from the induction hypothesis, corresponds to a floorplan. The floorplan may now be modified as in Theorem 6 so that the resulting floorplan corresponds to the given Z graph. ■

IV. Coding and Decoding

A. The Q sequence

In this section, we show how to represent $epVH(F)$ by a sequence as it is more efficient to manipulate a sequence than a graph when using a technique such as simulated annealing to search the state space.

Definition State of a Room, W_L, or W_T:

For a room r, its Q state $Q(r)$ is the sequence of the room label r followed by positional symbols $\mathcal{R}$ (if the prime seg is vertical) or $\mathcal{B}$ (if the prime seg is horizontal) with subscripts of the associated rooms in decreasing order.

For symbol W_L, $Q(W_L)$ is the sequence W_L followed by $\mathcal{R}$ with subscripts of rooms that adjoin the left wall in decreasing order. $Q(W_T)$ is similarly defined.

The state of a room may be defined on the LPD of $epVH(F)$ as the sequence consisting of v_r (or W_L or W_T) followed by the labels of out edges from v_r (or W_L or W_T) arranged in anti-clockwise order for $\mathcal{R}$ edges or clockwise order for $\mathcal{B}$ edges.

□

Definition Q sequence:

The concatenation of the states of W_L, W_T, and rooms in Abe order. It is written as

$$((Q(W_L)Q(W_T)Q(1)Q(2)\cdots Q(n)).$$

□

The Q sequence of the floorplan in Fig. 7 is

$$(W_L R_5\mathcal{R}_1 W_T\mathcal{B}_2\mathcal{B}_1\, 1\mathcal{R}_3\mathcal{R}_2\, 2\mathcal{B}_7\mathcal{B}_4\mathcal{B}_3\, 3\mathcal{R}_4\, 4\mathcal{B}_6\mathcal{B}_5\, 5\mathcal{R}_6\, 6\mathcal{R}_7 7).$$

Since W_L and W_T have no corresponding rooms and can be inferred from the context, they are often omitted in a Q-sequence representation as follows:

$$(R_5\mathcal{R}_1\,\mathcal{B}_2\mathcal{B}_1\, 1\mathcal{R}_3\mathcal{R}_2\, 2\mathcal{B}_7\mathcal{B}_4\mathcal{B}_3\, 3\mathcal{R}_4\, 4\mathcal{B}_6\mathcal{B}_5\, 5\mathcal{R}_6\, 6\mathcal{R}_7\, 7).$$

Next, we introduce the Z seq to describe necessary and sufficient conditions for a sequence to be a Q sequence.

Definition Z seq:

A Z seq contains W_L, W_T, $1, \ldots, n$ which appear in this order in the string. It also contains positional symbols $\mathcal{R}_1, \cdots, \mathcal{R}_n$ and $\mathcal{B}_1, \cdots, \mathcal{B}_n$. These may appear anywhere except before W_L. One or more $\mathcal{R}$s appear between W_L and W_T and one or more $\mathcal{B}$s between W_T and 1.

□

A Q sequence is clearly a Z seq.

Theorem 8: A Z seq is a Q sequence if and only if the following two conditions are satisfied.

Unique-Symbol: There is a sequence of length at least one of positional symbols between two adjacent room labels consisting exclusively of $\mathcal{R}$s or $\mathcal{B}$s.

Parenthesis: Subscripts are assigned to the $\mathcal{R}$s and $\mathcal{B}$s so that the sequence is a *parenthesis system* under the ordered pairing of $(\mathcal{R}_k, k)$, and also under the ordered pairing of $(\mathcal{B}_k, k)$ for $k = 1, \ldots, n$.[2]

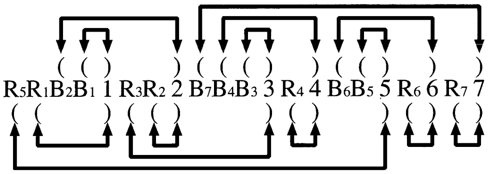


Fig. 9. Illustration of the parenthesis system of the Q sequence corresponding to Fig. 5.

Fig. 9 illustrates the parenthesis systems pairing $(\mathcal{B}_i, i)$ (shown above) and (R_i, i) (shown below).

Proof: Consider a Q sequence that (by definition) corresponds to some graph $epVH(F)$. Property **Unique-Symbol** reflects the property that the outgoing edges from a vertex are either all $\mathcal{B}$s or all $\mathcal{R}$s and the existence of a path along the Abe order. The Q sequence can be used to construct an epVH graph (Fig. 9) by replacing the Q-state of each room, say $k\mathcal{R}_i \ldots \mathcal{R}_j$, by a vertex v_k and out edges labeled $\mathcal{R}_i, \ldots$, and $\mathcal{R}_j$ going to vertices $v_i, \ldots, v_j$. Thus, each edge represents a pair of parentheses. Property **Parenthesis** of a Q sequence is then a direct consequence of the planarity of an epVH graph. Thus, it is clear that for a Z seq to be a Q sequence, it must have these properties. To see that it is sufficient for Z seq to have these properties, we describe the construction of a floorplan from a string (decoding) that has these properties in the next section. ■

B. Decoding

Next, we show how to decode a given Q sequence to a floorplan. The sequence of positional symbols between k and $k+1$ is denoted by $I(k)$ and the number of elements by $|I(k)|$. In the epVH graph, $I(k)$ and $|I(k)|$ correspond to the set of outgoing edges from v_k and the outdegree of v_k, respectively. $I(k)$ is called the *$\mathcal{R}$-interval* or *$\mathcal{B}$-interval* depending on the symbols it consists of. The decoding procedure in the following is illustrated by the example in Fig. 10.

Algorithm Decoding of a Q sequence to a Floorplan
Input: Q, a Q sequence
Output: F, a floorplan.
Initialization:
Let F be a floorplan consisting of one room with label n.
Main Step:
Repeat the following step for $k = n-1, n-2, \ldots 1$.

1) $I(k)$ **is an $\mathcal{R}$-interval**: Insert room k so that it becomes the left-top room such that its left seg is the left wall and its right seg is the left seg of top $|I(k)|$ rooms.
See Fig. 10(A), (C), and (E).

2) $I(k)$ **is a $\mathcal{B}$-interval**: Insert room k so that it becomes the left-top room such that its top seg is the top wall and its bottom seg is the top segs of left $|I(k)|$ rooms.
See Fig. 10(B), and (D).

□

The algorithm is based on the principles enunciated in the proof of Theorem 6. A formal proof of correctness is omitted. Instead, an informal explanation of the step illustrated in Fig. 10(E) is provided. This is the stage when room 1 is going to be inserted.

[2] The "parenthesis system under a specified ordered pairing (x, y)" is conventional if we imagine x and y to be the parenthesis pair "(" and ")". Formally, it is a sequence that satisfies the hierarchical property that successive deletion of the specified pairs that have no inside pairs results in a null sequence.

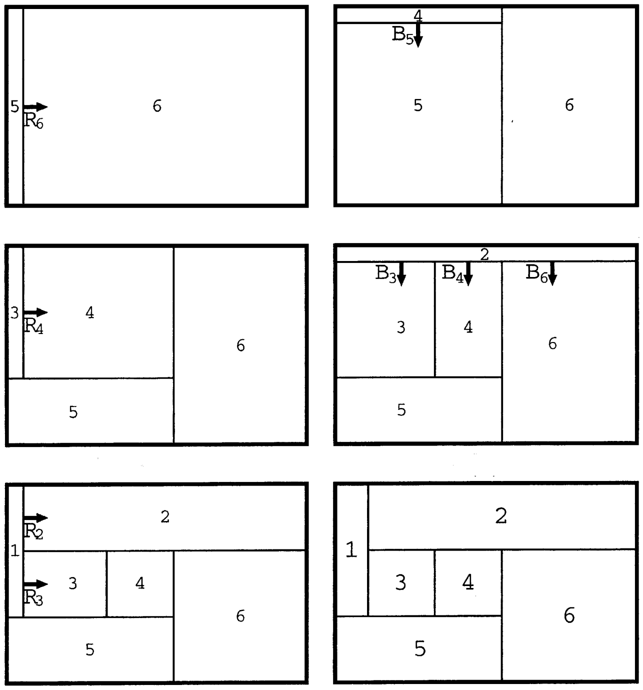


Fig. 10. Decoding of a Q sequence. (The interval under consideration is underlined): (A): $\mathcal{R}_5\ \mathcal{R}_1\ \mathcal{B}_2\ \mathcal{B}_1 1\ \mathcal{R}_3\ \mathcal{R}_2 2\ \mathcal{B}_6\ \mathcal{B}_4\ \mathcal{B}_3 3\ \mathcal{R}_4 4\ \mathcal{B}_5 5\ \underline{\mathcal{R}_6} 6$ (B): $\mathcal{R}_5\ \mathcal{R}_1\ \mathcal{B}_2\ \mathcal{B}_1 1\ \mathcal{R}_3\ \mathcal{R}_2 2\ \mathcal{B}_6\ \mathcal{B}_4\ \mathcal{B}_3 3\ \mathcal{R}_4 4\ \underline{\mathcal{B}_5} 5\ \mathcal{R}_6 6$ (C): $\mathcal{R}_5\ \mathcal{R}_1\ \mathcal{B}_2\ \mathcal{B}_1 1\ \mathcal{R}_3\ \mathcal{R}_2 2\ \mathcal{B}_6\ \mathcal{B}_4\ \mathcal{B}_3 3\ \underline{\mathcal{R}_4} 4\ \mathcal{B}_5 5\ \mathcal{R}_6 6$ (D): $\mathcal{R}_5\ \mathcal{R}_1\ \mathcal{B}_2\ \mathcal{B}_1 1\ \mathcal{R}_3\ \mathcal{R}_2 2\ \underline{\mathcal{B}_6\mathcal{B}_4\mathcal{B}_3} 3\ \mathcal{R}_4 4\ \mathcal{B}_5 5\ \mathcal{R}_6 6$ (E): $\mathcal{R}_5\ \mathcal{R}_1\ \mathcal{B}_2\ \mathcal{B}_1 1\ \underline{\mathcal{R}_3\mathcal{R}_2} 2\ \mathcal{B}_6\ \mathcal{B}_4\ \mathcal{B}_3 3\ \mathcal{R}_4 4\ \mathcal{B}_5 5\ \mathcal{R}_6 6$ (F): Unique resultant floorplan.

Rooms 6, 5, 4, 3, and 2 have already been inserted. Thus, five closed parentheses have been used. Two open parentheses corresponding to $\mathcal{R}_4$ and $\mathcal{R}_6$ have been used so far. So, $3 = 5 - 2$ rooms have their left segs on the left wall. Room 1 pushes two rooms inside (corresponding to $\mathcal{R}_3$ and $\mathcal{R}_2$).

C. Computational Complexity

The computational complexity of coding and decoding are determined as follows. Coding consists of traversing the rooms of the floorplan in Abe order to generate the Q-sequence representation, which requires $O(n)$ time. Decoding requires the sequence to be searched backward to determine the order in which rooms are to be inserted. This is also possible in $O(n)$.

Theorem 9: Coding (from a floorplan to a Q sequence) and decoding (from a Q sequence to a floorplan) are both possible in $O(n)$ time.

V. Counting Floorplans

A. Exact Number

Counting, listing, or generation of distinct floorplans of n rooms has been a major concern in combinatorics. See [18], [19]. Based on the previous sections, the problem is reduced to counting distinct Q sequences. A classification of the Q sequences is introduced.

Definition Class $C(m,r,b)$ of Q sequences: For $m = 1,2,\ldots,n$, $C(m,r,b)$ is the set of Q sequences that contain $r\,\mathcal{R}$'s and $b\,\mathcal{B}$'s in the subsequence consisting of m rooms labeled from $n-m+1$ to n.

For example, $(\mathcal{R}_5\mathcal{R}_1\mathcal{B}_2\mathcal{B}_1 1\mathcal{R}_3\mathcal{R}_2 2\mathcal{B}_6\mathcal{B}_4\mathcal{B}_3 3\mathcal{R}_4 4\mathcal{B}_5 5\mathcal{R}_6 6)$ is the Q sequence of the floorplan in Fig. 10. This belongs to $C(6,4,4)$, $C(5,2,4)$, $C(4,2,1)$, $C(3,1,1)$, $C(2,1,0)$, and $C(1,0,0)$.

If $C(m,r,b)$ is not empty, $m \le n$ and $r,\ b \le n-1$. No two sets $C(m,r,b)$ and $C(m,r',b')$ contain a Q sequence in common if $r \ne r'$ or $b \ne b'$. Hence, the number $F(n)$ of distinct floorplans with n rooms is given by

$$F(n) = \sum_{r=0}^{n-1}\sum_{b=0}^{n-1} |C(n,r,b)|. \tag{1}$$

To calculate this numerically, we use a recursive formula for $|C(n,r,b)|$. Any Q sequence of $C(n,r,b)$ is obtained from either

1) $C(n-1,r',b)$, $r' < r$ by adding a Q- state of a room in the form (*room label*) $\mathcal{R}_x \ldots \mathcal{R}_y$ where the number of $\mathcal{R}$s is $r - r'$, or
2) $C(n-1,r,b')$, $b' < b$ by adding a Q state of a room in the form (*room label*) $\mathcal{B}_x \ldots \mathcal{B}_y$ where the number of $\mathcal{B}$s is $b - b'$.

Hence, $|C(n,r,b)| = \sum_{r'=0}^{r-1} |C(n-1,r',b)| + \sum_{b'=0}^{b-1} |C(n-1,r,b')|$.

The boundary conditions are obtained by observing that $C(1,0,0)$ consists of exactly one floorplan and $C(n,r,b) = \emptyset$ if $r \ge n$ or $b \ge n$

$$F(n)=\sum_{r=0}^{n-1}\sum_{b=0}^{n-1} |C(n,r,b)|, \text{ where} \tag{2}$$

$$|C(n,r,b)|=\begin{cases} 1, & n=1, r=b=0\\ 0, & r,b \ge n\\ \sum_{r'=0}^{r-1}|C(n-1,r',b)| & \\ +\sum_{b'=0}^{b-1}|C(n-1,r,b')|, & \text{otherwise.}\end{cases} \tag{3}$$

It must be noted that $F(n)$ counts the number of *structurally* distinct floorplans. If we assign labels to room numbers, the total number of distinct floorplans is $n!F(n)$.

B. Upper Bounds

Let Q^* be the sequence obtained from a Q sequence Q by deleting all $\mathcal{B}$'s. Q^* is a parenthesis system with respect to pairing $(\mathcal{R}_k, k)$. It is known that the variety of the parenthesis system of n parentheses is [23, p. 495]

$$\text{Catalan}(n) \sim \frac{2^{2n}}{(n+1)\sqrt{\pi n}.}$$

Given Q^*, consider reconstructing the corresponding Q sequence. The positional symbol $\mathcal{B}$ can be inserted before room 1 and between two successive rooms, except when the intervals have one or more $\mathcal{R}$s. Further, every interval without $\mathcal{R}$ must be filled with at least one $\mathcal{B}$. For a given Q^*, the number of possibilities is bounded by the number of ways $n\,\mathcal{B}$s can be placed in $n-m$ intervals, where m is the number of intervals that contain $\mathcal{R}$s. This quantity is ${}_{n-1}C_{n-m-1}$, which is bounded by 2^{n-1}. Multiplying this quantity with the Catalan number yields

$$F(n) \le \text{Catalan}(n) \times 2^{n-1} \le 2^{2n} \times 2^{n-1} \le 2^{3n} \tag{4}$$

This is tighter than the upper bound $\text{Catalan}(n) \times n!$ in [18] and [19].

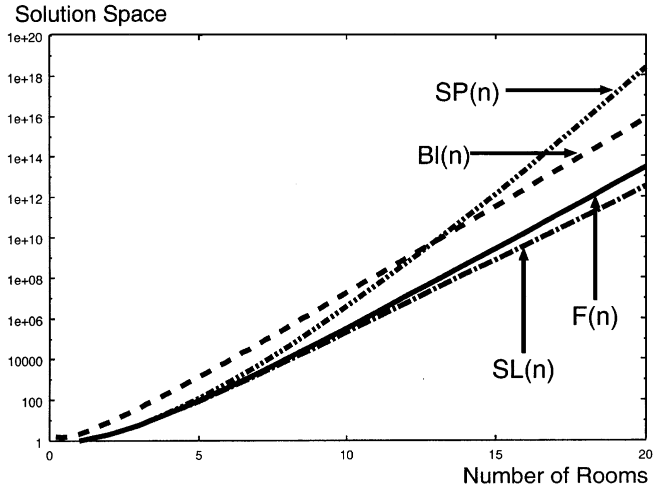


Fig. 11. Numerical comparison of focused functions (normalized by $1/n!$).

C. Comparison With Other Representations

The number of distinct binary trees with n leaves and two kinds of inner node labels is $BI(n) = \mathrm{Catalan}(n) \times 2^{n-1} \times n!$. (The number of structurally distinct binary trees is $\mathrm{Catalan}(n)$ and the number of internal nodes $n-1$ each of which can be of two types. The leaves can be labeled in $n!$ ways.) Since any slicing floorplan is represented by this class of binary trees (though there is some redundancy), $BI(n)$ is an upper bound of the number of slicing floorplans. However, our result (4) shows that $BI(n)$ is also an upper bound of the number of general floorplans. So, we continue with a more detailed analysis.

Let $SL(n)$ be the exact number of distinct slicing structure floorplans. Clearly, inequality is $SL(n) \leq F(n) \leq BI(n)$. Our interest is in the difference between $F(n)$ and $SL(n)$. We computed $SL(n)$ numerically using the idea of *normalized binary trees* [2]. $F(n)$ was numerically calculated by (2) and (3). Up to $n = 20$, they are approximated as

$$F(n) \sim \mathrm{Catalan}(n) \times 2^{0.6518n-2.26}$$
$$SL(n) \sim \mathrm{Catalan}(n) \times 2^{0.5072n-1.45}.$$

Fig. 11 shows the graph. $BI(n)$ and $SP(n) = (n!)^2$ (the number of packings using SPs) are also shown for comparison purposes.

From these data, we observe that little significant reduction of the solution space is achieved by restricting the floorplan to the slicing structure. It could also be concluded that the SP produces a great amount of redundancy with respect to n-room floorplanning, or, equivalently, n-room floorplans cover an insufficient part of the unconstrained packing space.

VI. Remarks on Q Sequence

There are several natural questions that arise about the Q-sequence structure and possible variations. Below, we list four questions and provide answers.

A. Q sequences Starting With Noncorner Room

The proposed Q sequence starts with the room at a corner of the chip and ends at the room at the diagonally opposite corner. The key was that the "next room" is well defined for each room and that this can be extrapolated to the global property of the Abe ordering. Is it possible to define a sequence by starting with an arbitrary room and merging the two sequences leading from this room in opposite directions? Does such a sequence cover all the rooms and encode the floorplan?

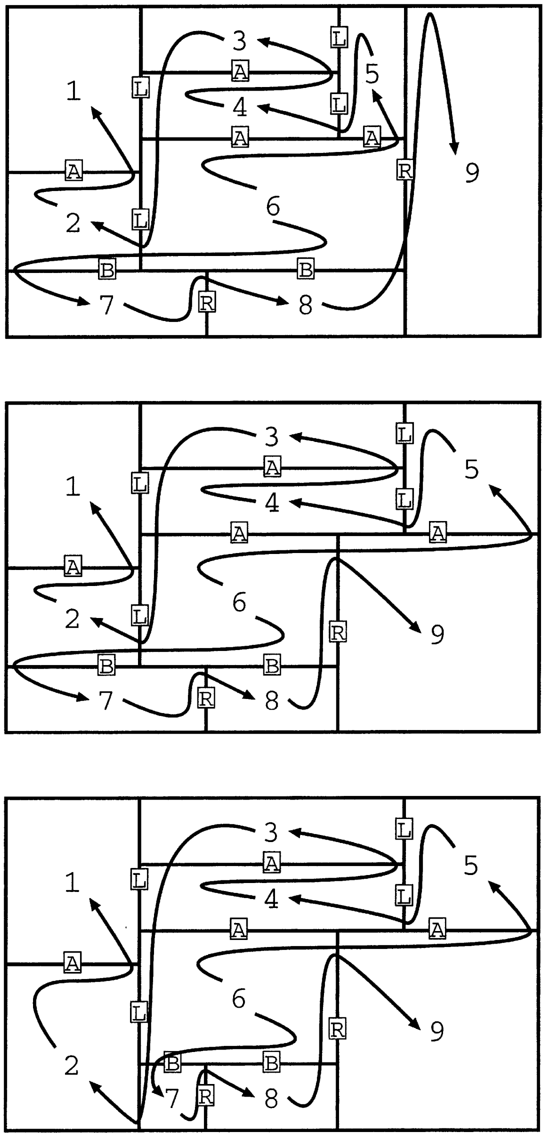


Fig. 12. Merged Q sequence with origin room 6 is common to three floorplans: $(1\mathcal{A}_1 2\mathcal{L}_2\mathcal{L}_1 3\mathcal{A}_3 4\mathcal{L}_4\mathcal{L}_3 5\mathcal{A}_5\mathcal{A}_4 \underline{6}\mathcal{B}_8\mathcal{B}_7 7\mathcal{R}_8 8\mathcal{R}_9 9)$.

It is true that the proposed sequence covers all the rooms. However, the following counterexample shows that such a sequence does not uniquely encode a floorplan. In Fig. 12, three distinct floorplans are shown. The sequence starting with room 6 toward the right bottom is $(6\mathcal{B}_8\mathcal{B}_7 7\mathcal{R}_8 8\mathcal{R}_9 9)$ for all three. The sequence $(6\mathcal{A}_4\mathcal{A}_5 5\mathcal{L}_3\mathcal{L}_4 4\mathcal{A}_3 3\mathcal{L}_1\mathcal{L}_2 2\mathcal{A}_1 1)$ starting with room 6 and leading toward the left-top corner is also common to all three. (Here, positional symbols $\mathcal{A}$ and $\mathcal{L}$ point to the rightmost room above and bottom-most room on the left, respectively.)

B. Double Q State (Half State)

We have so far focused on the sequence of rooms from the left top to the right bottom. A symmetric discussion holds for the sequence of rooms from the right top to the left bottom, which could be used to define another Q sequence. It is apparently redundant to use two Q sequences to represent a floorplan when one is enough. However, redundancy sometimes has benefits. Is there any merit to maintaining more than one Q sequence to reduce computation time or to study other properties of the floorplan?

Define the Half state (*H state*) of a room as the concatenation of its right-bottom Q state (as proposed in this paper) and the left-above Q state using symbols $\mathcal{L}$ and $\mathcal{A}$. The *Half-state sequence* (*H sequence*) is the concatenation of H states of all rooms. In the example in Fig. 10(F), the H state of room 2 is $\mathcal{L}_1 2\mathcal{B}_6\mathcal{B}_4\mathcal{B}_3$ and that of room 5 is $\mathcal{A}_4\mathcal{A}_3\mathcal{A}_1 5\mathcal{R}_6$. The H sequence is $(\mathcal{R}_5\mathcal{R}_1\mathcal{B}_2\mathcal{B}_1 1\mathcal{R}_3\mathcal{R}_2\mathcal{L}_1 2\mathcal{B}_6\mathcal{B}_4\mathcal{B}_3\mathcal{A}_2 3\mathcal{R}_4\mathcal{L}_3 4\mathcal{B}_5\mathcal{A}_4\mathcal{A}_3\mathcal{A}_1 5\mathcal{R}_6\mathcal{L}_5\mathcal{L}_4 6\mathcal{L}_6\mathcal{L}_2\mathcal{A}_6\mathcal{A}_5)$.

The length is $5n$, 5/3 times larger than that of the original Q sequence. However, the interval $I(k)$ between k and $k+1$ contains the information about the rooms that are adjacent to the prime seg of k. Thus, this representation has a potential in application for VLSI place-and-route design by channel routing scheme. For the above example, $I(4) = (\mathcal{B}_5\mathcal{A}_4\mathcal{A}_3\mathcal{A}_1)$ provides us with an information that the seg between rooms 4 and 5 is horizontal, and its adjoining rooms are 4, 3, and 1 above and room 5 below. Thus, it can be quickly determined whether two rooms are adjacent. If the Q sequence was being used, $O(n)$ time would be required to answer these questions.

It is an important problem to specify the *boundary condition* in layout design so that specified modules are laid out along the boundary of the chip to facilitate I/O pin connections [24]. In an H sequence, the rooms on the boundary are present in the pre and post sequences. In our example, pre sequence $\mathcal{R}_5\mathcal{R}_1\mathcal{B}_2\mathcal{B}_1$ implies that rooms 5 and 1 are adjacent to the left wall, and rooms 2 and 1 to the top wall. Similarly, post sequence $\mathcal{L}_6\mathcal{L}_2\mathcal{A}_6\mathcal{A}_5$ shows that rooms 6 and 2 are adjacent to the right wall, and rooms 6 and 5 to the bottom wall. This feature can be exploited when one searches floorplans under the boundary constraint using simulated annealing.

C. Coding by the Tree Pair

The parenthesis system is equivalent to a rooted tree representation of the data. The Q sequence embeds two parenthesis systems. Is it possible to represent a Q sequence or a floorplan by a pair of rooted trees?

Consider the floorplan in Fig. 5 and the parenthesis configuration of its Q sequence in Fig. 9. We can interpret the $\mathcal{B}$-parenthesis system as the rooted tree described as follows: The root has two children 2 and 7, of which 2 has one child 1, and 1 is a leaf. Additionally, 7 has two children 4 and 6, of which 4 has one child 3. The child 3 is a leaf. Additionally, 6 has one child 5, and 5 is a leaf. This tree is drawn on the floorplan as shown in Fig. 13 (left) where nodes denote the rooms and the region to the right of the chip. The rooted tree may be constructed by the rule: each node v_r that corresponds to a room r with right seg s_r has exactly one incoming edge. It is from a node that corresponds to the top room of the rooms to the right of s_r. This is equivalently stated as follows: for each seg s_p, every node corresponding to a room on the left has an incoming edge from the node corresponding to room $p+1$. Node R is regarded as $n+1$. The same claim holds for the parenthesis system with respect to $\mathcal{R}$. Its rooted tree is shown in the figure (right). It is a trivial but interesting exercise to find the relations with the graph $VH(F)$, $pVH(F)$, or $epVH(F)$.

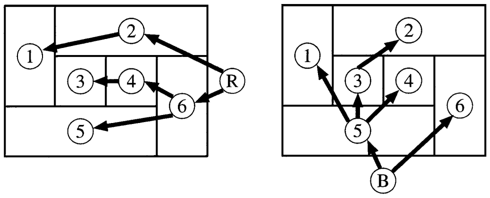


Fig. 13. Tree-pair representation of the floorplan in Fig. 5.

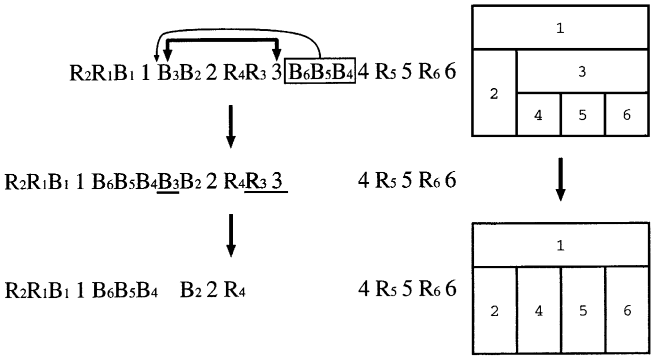

R2R1B1 1 B3B2 2 R4R3 3 B6B5B4 4 R5 5 R6 6
R2R1B1 1 B6B5B4B3B2 2 R4R3 3 4 R5 5 R6 6
R2R1B1 1 B6B5B4 B2 2 R4 4 R5 5 R6 6
1
2
3
4
5
6
1
2
4
5
6

Fig. 14. Room deletion $(i = 3)$.

D. Space Requirements of the Q-Sequence Representation

In practical applications, the space complexity is usually not as important as the time complexity. Sometimes, the computation time may be reduced by storing a precomputed result which increases the space complexity. However, it is always of theoretical interest to determine the minimum number of bits needed to most efficiently store information. In packing, it has been reported that the O-tree data structure [8] and LOT [25] require $2n + n\log n$ bits. Recall that the O tree only contains partial topological information and requires information about module sizes to reconstruct the floorplan. Moreover, we are considering a complete floorplan representation rather than a partial packing representation.

The space complexity estimation of a straightforward implementation is $3n\log 3n$ since a Q sequence consists of $3n$ letters and it needs $\log 3n$ space to identify $3n$ kinds of labels. A more efficient representation stores the two parenthesis systems using $2\times 2n$ bits. Since $n\log n$ bits are necessary to store room labels, the space complexity is $4n + n\log n$. Consider the example in Fig. 9. The parentheses systems can be stored by denoting an open parenthesis with a "1" and a closed parenthesis with a "0" as follows:

$\mathcal{R}$-parenthesis: 11 011 001 001 010;
$\mathcal{B}$-parenthesis: 11 001 110 011 000.

The room sequence is stored as 1 234 567. Note that if we are only interested in the floorplan *structure*, the room sequence can be eliminated and only $4n$ bits are needed.

VII. Packing by Floorplanning and Experiments

As discussed previously, a floorplanning approach to the packing problem may not yield an optimal solution even if

TABLE I
EXPERIMENTAL RESULTS WITH DIFFERENT REPRESENTATIONS

Circuits	Data structures	Area ratio(%)			CPU Time(s)		
		best	worst	average	best	worst	average
ami49	Q-sequence: A_Q	96.9	95.3	96.0	27	29	28
ami49	Seq-pair: A_{SP}	97.2	95.7	96.3	174	.178	178
ami49	BSG(49x49):A_{BSG}	97.3	96.5	96.9	1788	1789	1789
ami33	Q-sequence: A_Q	96.4	94.2	95.3	21	22	21
ami33	Seq-pair: A_{SP}	96.9	94.1	95.9	86	88	87
ami33	BSG(33x33):A_{BSG}	98.2	96.4	97.2	668	670	669

the entire floorplanning space is examined. Since it is not practical to search the entire solution space, the quality of the packing depends significantly on the time required to evaluate one solution. We implemented a simulated annealing packing algorithm based on the Q sequence. The input consists of n rectangular modules whose dimensions are provided. A floorplan is represented by a Q sequence. Modules are assigned to the rooms in one-to-one fashion. Each room must be large enough to contain the corresponding module. The process of computing the bounding box area is conventional and omitted here.

A key element of the search is the procedure *move* that transforms one Q sequence to another. Our proposed move is called the *del-ins move* which consists of deleting and inserting a room. The deletion procedure is described below. See Fig. 14.

```
Procedure Room Deletion $(i)$
    If $I(i)$ is $\mathcal{R}$- ($\mathcal{B}$-)interval and $|I(i-1)| > 1$, move $I(i)$ to the place before $\mathcal{R}_i(B_i)$.
    Delete $i$, $\mathcal{R}_i$, and $\mathcal{B}_i$.
```

Room Insertion is the inverse of deletion and its details are omitted here.

The *solution space* may be defined as the pair (a set of solutions, move). A solution space is said to be *reachable* if any solution may be obtained by successive applications of the move starting with any initial solution. The maximum number of applications needed to get one solution from any other solution is the *diameter* of the solution space. A small diameter is desirable.

Theorem 10: The solution space (floorplans defined by Q sequence,del-ins move) is reachable and the diameter is n.
□

Our packing algorithm is named A_Q. For comparison, we used two competitive algorithms A_{SP} (which is based on the SP)[3] and A_{BSG} (which is based on the BSG)[6]. Both provide a solution space guaranteed to contain an optimum solution. The benchmark circuits "ami33" and "ami49" were used. The annealing schedule for these three algorithms are the same. The program is written in C++ and ran on Pentium III 910-MHz CPU. Packing was tried ten times for each instance. The best, worst, and average area ratios (sum of area of modules/area of floorplan) are listed in Table I.

The parameters of the simulated annealing are chosen so that the number of cycles is the same for all three. Thus the CPU time is proportional to the time of one cycle.

We observe that A_Q resulted in solution qualities that are comparable to those of A_{SP} and A_{BSG}. However, the CPU times of A_Q were substantially smaller than the competition.

VIII. CONCLUDING REMARKS

We began by differentiating between floorplanning and packing and between room–room and room–seg adjacencies. We have addressed the problem of floorplanning based on room–seg adjacencies. The Q sequence is a string representation of a floorplan, but is equivalent to graph representations such as $pVH(F)$ and $epVH(F)$. The Q sequence is so named because it concatenates the states of one of four corners of each room. The Q sequence is a very convenient representation of the floorplan because its rooms are arranged in Abe order. Since the Abe order is unique for a floorplan, room labels are not required to represent the *structure* of a floorplan. We introduced a simple transformation operation on the Q sequence that makes it possible to convert an initial floorplan to *any* floorplan in just n steps. This greatly facilitates searching the floorplan state space. If a floorplan is to be evaluated by the size of the area of the bounding rectangle, then we determined experimentally that, in spite of our restriction that each room contains exactly one module (i.e., no rooms are empty), our solutions compare with those obtained by BSG and SP that have the advantage of having a larger search space that includes all possible packings.

ACKNOWLEDGMENT

The authors are grateful to Associate Professor A. Takahashi, Tokyo Institute of Technology for his valuable advice and to Dr. Changwen Zhuang (Research Associate, The University of Kitakyushu) for his suggestion for the proof of Theorem 6. They would also like to thank Prof. Thulasiraman for facilitating the collaboration between the first two co-authors and the third co-author.

REFERENCES

[1] R. H. J. M. Otten, "Automatic floorplan design," in *Proc. 19th ACM/IEEE DAC*, 1982, pp. 261–267.

[2] D. F. Wong and C. L. Liu, "Floorplan design of VLSI circuits," *Algorithmica*, vol. 4, pp. 263–291, 1989.

[3] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI module placement based on rectangle-packing by the sequence-Pair," *IEEE Trans. Comput. Aided Design*, vol. 15, pp. 1518–1524, Dec. 1996.

[4] H. Murata, K. Fujiyoshi, T. Watanabe, and Y. Kajitani, "A mapping from sequence-pair to rectangular dissection," in *Proc. ASPDAC*, 1997, pp. 625–633.

[5] K. Kiyota and K. Fujiyoshi, "Simulated annealing search through general structure floorplans using sequence-pair," in *Proc. ISCAS*, vol. 3, 2000, pp. 77–80.

[6] S. Nakatake, H. Murata, K. Fujiyoshi, and Y. Kajitani, "Module packing based on the BSG-structure and IC layout applications," *IEEE Trans. Comput. Aided Design*, vol. 17, no. 6, pp. 519–530, 1998.

[7] J.-M. Lin and Y.-W. Chang, "TCG: a transitive closure graph-based representation for nonslicing floorplans," in *Proc. 38th IEEE/ACM Design Automation Conf.*, 2001, pp. 764–769.

[8] P. N. Guo, C. K. Cheng, and T. Yoshimura, "An O-tree representation of nonslicing floorplan and its applications," in *Proc.36th IEEE/ACM Design Automation Conf.*, 1999, pp. 268–273.

[9] Y. C. Chang, Y. W. Chang, G. M. Wu, and S. W. Wu, "B^*-trees: a new representation for nonslicing floorplans," in *Proc.37th IEEE/ACM Design Automation Conf.*, 2000, pp. 458–463.

[10] K. Sakanushi, K. Midorikawa, and Y. Kajitani, "A general and fast floorplaning by reduct-seq representation,", Tech. Rep. 120, IEICE (VLD2000-24), June 2000.

[11] K. Sakanushi and Y. Kajitani, "Counting of the topological dissections by reduct-seq representation,", Tech. Rep. 144, IEICE (COMP2000-17), June 2000.

[12] ——, "The Quarter-Sequence (Q-Seq) to represent the floorplan and applications to layout optimization," in *Proc. IEEE Asia Pacific Conf. on Circuits and Systems*, Tianjin, China, Dec. 2000, pp. 829–832.

[13] X. Hong, G. Huang, Y. Cai, J. Gu, S. Dong, C.-K. Cheng, and J. Gu, "Corner block list: an effective and efficient topological representation of nonslicing floorplan," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, 2000, pp. 8–12.

[14] B. Yao, H. Chen, C.-K. Cheng, and R. Graham, "Revisitng floorplan representations," in *Proc. Int. Symp. Physical Design*, 2001, pp. 138–143.

[15] E. F. Y. Young, C. C. N. Chu, and Z. C. Shen, "Twin Binary Sequences: A nonredundant representation for general nonslicing floorplan," in *Proc. Int. Symp. Physical Design*, 2002, pp. 196–201.

[16] S. Zhou, S. Dong, X. Hong, Y. Cai, and C.-K. Cheng, "ECBL: an extended corner block list with solution space including optimum placement," in *Proc. Int. Symp.Physical Design*, 2001, pp. 156–161.

[17] M. Abe, "Covering the square by squares without overlapping (in Japanese)," *J. Japan Math. Phys.*, vol. 4, no. 4, pp. 359–366, 1930.

[18] T. Shimuzu, "Plane dividing T-configurations with consequent numbering and T-symbolism for orthogonal case," *Soc. Sci. Form, Forma*, vol. 5, no. 2, pp. 173–178, 1990.

[19] ——, "The rectangular dissection (in Japanese)," in *Proc. Nippon Hyoronsya*, Apr. 1999.

[20] J. Bhasker and S. Sahni, "A linear algorithm to find a rectangular dual of a planar triangulated graph," *Algorithmica*, vol. 3, no. 2, pp. 274–278, 1988.

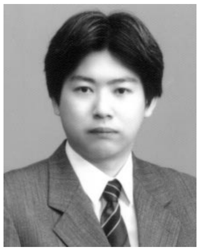
[21] S. Tsukiyama, M. Maruyama, S. Shinoda, and I. Shirakawa, "A condition for a maximal planar graph to have a unique rectangular dual and its application to VLSI floor-plan," in *Proc. Int. Symp. Circuits and Systems*, 1989, pp. 931–934.

[22] Y.-T. Lai and M. Leinwand, "A theory of rectangular dual graphs," *Algorithmica*, vol. 5, pp. 467–483, 1990.

[23] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*, 2nd ed. Reading, MA: Addison-Wesley, 1994.

[24] F. Y. Young, D. F. Wong, and H. H. Yang, "Slicing floorplans with boundary constraints," *IEEE Trans. Comput. Aided Design*, vol. 18, pp. 1385–1389, Sept. 1999.

[25] T. Takahashi, "A new encoding scheme for rectangle packing problem," in *Proc. ASPDAC*, 2000, pp. 175–178.

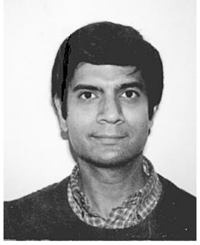


Keishi Sakanushi received the B.E, M.E., and D.E. degrees in electrical and electronics engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1997, 1999, and 2002, respectively.

He has been a Research Associate in the Department of Information Science and Technology, Osaka University, Osaka, Japan, since April 2001 when the department was founded. His research is in combinatorial algorithms with an emphasis in optimization of floorplans in VLSI layout design.

Yoji Kajitani (SM'80–F'92) received the Ph.D. degree in electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan.

He is a Professor in the Department of Information and Media Science, Faculty of Environmental Engineering, University of Kitakyushu, Kitakyushu, Japan, since April 2001 when the faculty was founded. He was previously at the Tokyo Institute of Technology, Tokyo, Japan, where he had been a Professor and Associate Professor since 1985, and 1972, respectively. All degrees, including the Ph.D. in Electronic Engineering, were from the Tokyo Institute of Technology. He started his academic research career working on the application of graph theory to electrical circuits. One of his main contributions was the discovery of the "Principal Partition of a graph", by which the minimum set of voltages and currents that can describe a circuit is determined. From the eighties, before a one-year stay at University of California at Berkeley, he shifted his interest to VLSI layout design. His major contributions since then are theoretical results about 1. Minimum width and height channel routing, 2. Minimum number of vias, and 3. Data structures Bounded-Slicing-Grid (BSG) and Sequence-Pair (SP) for placement of rectangular modules.



Dinesh P. Mehta received the B. Tech. degree in computer science and engineering from the Indian Institute of Technology, Bombay, India, in 1987, the M.S. degree in computer science from the University of Minnesota, Minneapolis, in 1990, and the Ph.D. degree in computer science from the University of Florida, Gainesville, in 1992.

He was on the faculty at the University of Tennessee Space Institute from 1992–2000. He was a Visiting Professor at Intel's Strategic CAD Labs in 1997. He has been an Associate Professor in the Mathematical and Computer Sciences Department at the Colorado School of Mines, Golden, since 2000. Dr. Mehta is a co-author of the text *Fundamentals of Data Structures in C++*. His publications and research interests are in VLSI design automation, parallel computing, and applied algorithms and data structures.

Dr. Mehta received the Vice President's Award for Teaching Excellence in 1997 while at the University of Tennessee Space Institute.