A Study on Hierarchical Floorplans of Order k

A THESIS

submitted by

SAJIN KOROTH

for the award of the degree

of

MASTER OF SCIENCE

(by Research)

DEPARTMENT OF COMPUTER SCIENCE
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

July 2011

THESIS CERTIFICATE

This is to certify that the thesis titled **A Study on Hierarchical Floorplans of Order** k, submitted by **Sajin Koroth**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Science**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Shankar Balachandran
Research Guide
Assistant Professor
Dept. of Computer Science and Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 25th July 2011

माता, पिता, गुरुर दैवं

ACKNOWLEDGEMENTS

अज्ञानतिमिरान्धस्य ज्ञानाञ्जनशलाकया ।
चक्षुरुन्मीलितं येन तस्मै श्रीगुरवे नमः ॥

Salutation to the noble Guru, who has opened the eyes blinded by darkness of ignorance with the collyrium-stick of knowledge.

I am dedicating my thesis to my wonderful thesis adviser Dr. Shankar Balachandran. He has been a great adviser, a good friend, an awesome mentor and above all a source of extreme optimism. There were times during my graduate research where I was staring into the void, depressed of not knowing what to do next. During those times my adviser, used to sit with me for hours trying to work out a solution and most of the times he was able to come up with great ideas which would help me solve the problem. I am greatly thankful to Dr. N. S. Narayanaswamy who has been of great help throughout my research, and was like a co-guide to me. The courses I took at Institute of Mathematical Sciences(IMSc) prepared me for graduate level research in theoretical computer science. Of the six courses I took there one requires a special mention. This course was directly related to my research, titled "Discrete Mathematics", taught by Prof. V. Arvind. It really pushed my ability to think - thanks to the challenging problem sets. I am thankful to the people at AIDB lab, especially Sadagopan Narasimhan and G. Ramakrishna for maintaining a good research environment and helping me with my research. I would like to thank my friends Murali Krishna Karnam, Anup Pydah, M. V. K. Chaithanya, Sarang Bharadhwaj, Vishwanath Avinash, Bharath Ram, Viswanath K. S., Ankit Kagliwal, Anju Srinivasan, R. Subhashini , Krithika, M. P. K. Pawan, Anish Kumar and Jayaraj Radhakrishnan for making my stay at IIT Madras a memorable and wonderful episode of my life. I would like to thank Dr. Ravi Krishna Menon a theoretical physicist, who coached me in mathematics and physics during classes XI and XII and inspired me to do research in theoretical sciences. I would like to thank my mathematics coach during high school, Dr. Laxmanan for giving me right training to pursue my passion - mathematics. I consider my self blessed to be part of Vivekananda Study Circle, IIT Madras which enriched my spiritual side. I would to thank the Ramakrishna Math-Chennai, especially Swami Atmashraddhananda for the spiritual support. I thank the CS Office for providing administrative support. Finally, I would like to conclude my acknowledgments by thanking my family. This journey would have been impossible without immense motivation and support extended by my grandfather P. Balan, my father Sreedharan who departed from this world four years back and my mother Gomathy. All the goodness in me, all the heights I have attained is because of them.

ABSTRACT

KEYWORDS: Combinatorics; VLSI Design; Hierarchical Floorplan; Simulated Annealing.

A floorplan is a rectangular dissection which describes the relative placement of electronic modules on the chip. It is called a mosaic floorplan if there are no empty rooms or cross junctions in the rectangular dissection. We study a subclass of mosaic floorplans called hierarchical floorplans of order k (abbreviated HFO$_k$). A floorplan is a hierarchical floorplan of order k if it can be obtained by starting with a single rectangle and recursively embedding mosaic floorplans of at most k rooms inside the rooms of intermediate floorplans. When $k = 2$ this is exactly the class of slicing floorplans as the only distinct floorplans with two rooms are a room with a vertical slice and a room with a horizontal slice respectively. Embedding such a room is equivalent to slicing the parent room vertically/horizontally.

In this thesis we characterize permutations corresponding to the Abe-labeling of HFO$_k$ floorplans and also give an algorithm for identification of such permutations in linear time for any particular k. We also prove that Hierarchical floorplans of order k are in bijective correspondence with Skewed Generating trees of Order k. From this result we give a recurrence relation for exact number of HFO$_5$ floorplans with n rooms which can be easily extended to any k also. Based on this recurrence we provide a polynomial time algorithm to generate the number of HFO$_k$ floorplans with n rooms. Considering its application in VLSI design we also give moves on HFO$_k$ family of permutations for combinatorial optimization using simulated annealing etc. We also explore some interesting properties of Baxter permutations which have a bijective correspondence with mosaic floorplans.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

HFO$_k$ Hierarchical Floorplan of Order k

NOTATION

$\lvert\pi\rvert$	Length of the permutation π
S_n	Set of all permutations of length n
f_π	Mosaic floorplan corresponding to the Baxter permutation π
$i, \ldots, j$	All natural numbers from i to j, inclusive of both i and j
π^r	Reverse of the permutation π

CHAPTER 1

INTRODUCTION

Floorplanning is an important phase of VLSI design. It is usually formulated as the problem of placing a given set of rectangular circuit modules on the plane to minimize some objective functions like area of the bounding rectangle containing all modules or total interconnection wire length. At an early stage of physical design like floorplanning the shape and and dimensions of the modules are in general not fixed. Hence a floorplan usually captures relative placement of modules alone and so is represented using rectangular dissections. The number of feasible solutions for a given instance of floorplanning problem is very large. The introduction of an objective function like interconnection wiring length lets us select superior floorplans from among the set of feasible solutions. But the introduction of an objective function makes floorplanning problem a combinatorial optimization problem thus increasing the hardness of the problem.

It is already known that floorplanning problem is NP-Hard. This is because floorplanning problem is a generalization of placement problem which is a generalization of quadratic assignment problem which is known to be NP-Hard (Sait and Youssef (1999)).

Due to the algorithmic hardness of the problem, stochastic search methods like Simulated annealing, Genetic algorithms etc. are used to find a floorplan which near-optimizes the objective function. These algorithms work by using a code to represent the floorplan and by making small perturbations on the codes to do a neighbourhood search for better codes in terms of the objective function. To work with a family of floorplans it is necessary to have a coded representation for that family. Since floorplan describes the relative placement of the blocks, it is modeled mathematically as a dissection of a rectangle with axis parallel (horizontal/vertical) non-intersecting line segments which captures the relative placement of the blocks. It is called a mosaic floorplan if there are no empty rooms or cross junctions in the rectangular dissection.

In this thesis we study a subclass of mosaic floorplans called hierarchical floorplans of order k (abbreviated HFO$_k$). A floorplan is a hierarchical floorplan of order k if it can be obtained by starting with a single rectangle and recursively embedding mosaic floorplans of at most k rooms inside the rooms of intermediate floorplans (Wong and Sakhamuri (1989)). When $k = 2$, this is exactly the class of slicing floorplans as the only distinct floorplans with two rooms are a room with a vertical slice and a room with a horizontal slice respectively. Embedding such a room is equivalent to slicing the parent room vertically/horizontally.

In this thesis, we characterize permutations corresponding to the Abe-labeling of HFO_k permutations and also give algorithms for identification of such permutations in polynomial time for any arbitrary k. These results can be used to create stochastic search algorithms on the family of HFO_k floorplans. We also explore interesting characteristics of Baxter permutations which have a bijective correspondence with mosaic floorplans.

1.1 History

Wong and Liu (1986) were the first to consider the use of stochastic search methods like simulated annealing for floorplan optimization problem. In their seminal paper on simulated annealing based search on the family of slicing floorplans (Wong and Liu (1986)), they introduced slicing trees and proved that there is a one-one correspondence between slicing floorplans with n rooms and skewed slicing trees with n leaves. They also proved that there is a one-one correspondence between skewed slicing trees with n leaves and normalized polish expression of length $2n - 1$. These normalized polish expressions were the coded representation of slicing floorplans in their simulated annealing search . The set of moves they defined on these normalized polish expressions defined the neighbourhood relation among floorplans in the search space. Wong and The (1989) gave a representation of hierarchical floorplans of order 5 extending the normalized polish expressions of slicing floorplans to incorporate wheels which are the only non-slicing floorplans with at most five rooms. They also described neighbourhood moves for simulated annealing search on HFO_5 floorplans based on this representation.

Since then, there has been attempts to find codes for larger families of floorplans like mosaic floorplans. Murata *et al.* (1995) were the first to come up with a nice representation of mosaic floorplans. They used a representation called a sequence pair to represent a mosaic floorplan uniquely. A sequence pair is a set of two permutations of length n which uniquely capture a mosaic floorplan. In the same paper they introduced the notion of **P-admissibility** for a coded representation of a floorplan family. These are a set of desirable properties for the solution space represented by the coded representation which would result in faster and better search on that family of codes. These properties are :

- The solution space is finite.
- Every solution is feasible.
- Evaluation of each code is possible in polynomial time and so is the realization corresponding to the packing.
- The floorplan corresponding to the best evaluated code in the space coincides with an optimal solution of the floorplanning problem.

They later improved sequence-pair representation to contain just one sequence (Murata *et al.* (1996)) where in which they utilized the value of the labels in the sequence to capture the first sequence in sequence pair and the relative position of these labels in the sequence to capture the second sequence in sequence pair. This sparked off a great interest in better and faster representations for non-slicing floorplans. Hong *et al.* (2000) came up with an entirely different data structure called Corner block list to represent mosaic floorplans.

The algorithms started looking at bigger families of floorplans because it is known that to capture the optimum solution based on an objective function like interconnection wire length for a floorplanning problem it is necessary to consider topologies with empty rooms also. Young *et al.* (2002) characterized that while searching for optimal floorplans, empty rooms need to be present only at the center of wheel structures(Figure 1.11). In fact characterizing and enumerating permutations that are mapped to optimal floorplans is one of the biggest open problems in designing codes corresponding to non-slicing floorplans. Zhou *et al.* (2001) came up with an extension to corner block list idea to incorporate the concept of empty rooms. They give an extension factor λ to corner block list to contain general non-slicing floorplan containing empty rooms. But to guarantee that the solution space contains the optimal solution their λ is to be set to n, the number of modules to be placed on the plane, thus making it a costly data structure.

Several other works addressed combinatorial properties of these codes like count, characterization, enumeration etc. Sakanushi *et al.* (2003) were the first to consider the number of distinct mosaic floorplans. They found a recursive formula for this number. Yao *et al.* (2003) showed a bijection between mosaic floorplans and twin binary trees whose number is known to be the number of Baxter permutations (Dulucq and Guibert (1998)). They have also shown that the number of distinct slicing floorplans containing n blocks is the $(n-1)$th Shrőder number. Later Ackerman *et al.* (2006) constructed a bijection between mosaic floorplans with n-rooms to Baxter permutations on $[n]$. They also proved that this bijection when restricted to Slicing floorplans gives a bijection from slicing floorplans with n-rooms to separable permutations on $[n]$. And with this bijection we can obtain a unique permutation corresponding to any mosaic floorplan or naturally for a floorplan which belongs to a subclass of mosaic floorplans.

To characterize the class of permutations corresponding to HFO$_k$ floorplans we need some results on a well studied class of permutations called Simple permutations. Simple permutations and their properties were studied first by Albert and Atkinson (2005). They proved a crucial theorem about exceptionally simple permutations using a result from a paper by Schmerl and Trotter (1993) about critically indecomposable partially ordered sets.

Another important combinatorial property of codes corresponding to floorplan families, which has been studied extensively in literature is the number of distinct floorplans with n rooms in a floorplan family . Shen and Chu (2003) presented a generating function based approach to count skewed slicing trees, to obtain a tight bound on number of slicing floorplans with n rooms. Chung *et al.* (1978) obtained closed form expression for the number of Baxter permutations of length n using a generating tree based approach.

1.2 Motivation

Hierarchical floorplans of order k are well studied in the context of area optimization problem. Area optimization problem for floorplans is, given a floorplan and modules which are rectangles with a finite set of implementations (in terms of dimensions of the module) select the optimal implementation for each module such that the total area is minimized. The area minimization for hierarchical floorplans of order 5 and above is proved to be NP-Complete (Pan and Liu (1995)). But for Hierarchical floorplans of order k, a near optimal algorithm for area minimization is given by Wang and Wong (1992). Even though area optimization problem for hierarchical floorplans of order k is well studied in the literature, to our best knowledge there are no floorplanning algorithms which work on HFO_k floorplans alone and there is no efficient computer representation of HFO_k floorplans. There are algorithms and representations for HFO_k for $k = 2$ and $k = 5$ by Wong and Liu (1986) and Wong and The (1989) respectively. But these representations cannot be easily generalized to any k. Hierarchical floorplans are an interesting family because of their hierarchical structure which simplifies physical design and synthesis of these floorplans and allows the synthesis to be carried out in a nice top-down fashion.

1.3 Preliminaries

1.3.1 Mosaic floorplans

Mosaic floorplans are rectangular dissections satisfying the following properties (Hong *et al.* (2000)) :

1. No empty rooms.
2. Topological equivalence on sliding line segments.
3. **Non-degenerate-topology** : No degenerate case where two distinct T junctions forms a + junction.

The first property says that in the floorplan there should be exactly as many rooms as there are modules which are to be placed on the chip. For example the floorplan shown in Figure 1.1 is not a mosaic floorplan because there are only 2 modules to be placed on the chip and there are 3 rooms in the given floorplan. The second property says that two floorplans are equivalent if one can be obtained from the other by sliding one or more line segments in such a way that all the other line segments which ends at line segment are also pulled along with the line segment when sliding it. By this definition floorplans (a) and (b) in Figure 1.4 are equivalent but (c) is different from both.

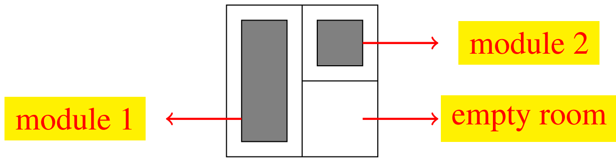


Figure 1.1: Floorplan with empty rooms

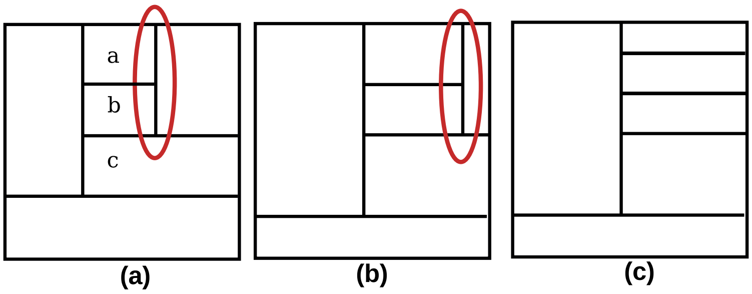


Figure 1.2: Topological Equivalence on sliding

The floorplan given in Figure 1.3 is not a mosaic floorplan because it violates the non-degenerate topology condition. The non-degenerate topology is needed because mosaic floorplans are defined so as to capture the following relative placement relations between blocks (Murata *et al.* (1997)). Let f be a mosaic floorplan and let b_1 and b_2 be blocks in f. We say that b_1 is to the *left of*(respectively, *above*) b_2 if there exists a line segment l of f which contains the *right* (respectively, *lower*) edge of b_1 and the *left*(respectively, *top*) edge of b_2, or if there exists a block b_3 such that b_1 is to the *left of*(respectively, *above*) block b_3 and b_3 is to the *left of*(respectively, *above*) block b_2. The degenerate case of a cross junction is not allowed because it will lead to multiple representation of the same floorplan for most of the codes in the literature. This is because a coded representation like the corner block list captures exactly one of $\{$*above*,*below*,*left*,*right* $\}$ relation between any two blocks. If there is a cross junction in a mosaic floorplan there can be more than one of these relations between blocks constituting the cross junction, resulting in multiple valid codes representing the same floorplan. The first floorplan in Figure 1.4 has a cross junction. But between blocks labeled a and b in this floorplan both *above* and *left of* relations hold as there is a line segment containing right edge of a and left edge b and there is another containing bottom edge of a and top edge of b. Hence cross junctions are not allowed in mosaic floorplans. The degenerate case is avoided by sliding one side of a line segment involved in a cross junction by a small amount to make it a T-junction as shown in second floorplan of Figure 1.4.

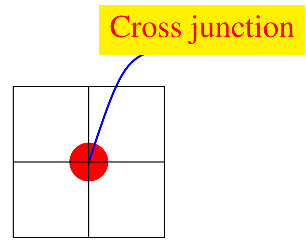


Figure 1.3: Non-Moaic Floorplan

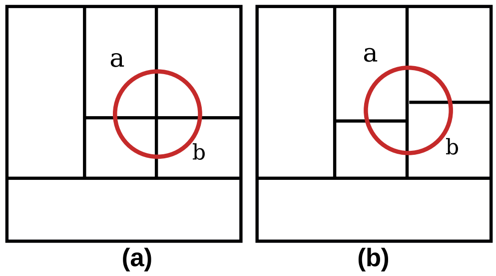


Figure 1.4: Handling non degenerate topology

1.3.2 Pattern Matching Problem on Permutations

Pattern matching problem for permutation is given a permutation $\pi \in S_n$ called *text* and another permutation $\sigma \in S_k$ called *pattern* we would like to know if there exists k indices $i_1 < i_2 < i_3 < i_4 < \ldots < i_k$ such that the numbers $\pi[i_1], \pi[i_2], \pi[i_3], \pi[i_4], \ldots, \pi[i_k]$ are in the same relative order as $\sigma[1], \sigma[2], \sigma[3], \sigma[4], \ldots, \sigma[k]$, that is $\pi[i_h] > \pi[i_l]$ if and only if $\sigma[h] > \sigma[l]$. If π contains such a sub-sequence we call **text** π contains the **pattern** σ, and the sub-sequence is said to **match** the **pattern**.

1.3.3 Baxter Permutations

A Baxter permutation on $[n] = 1, 2, 3, \ldots, n$ is a permutation π for which there are no four indices $1 \leq i < j < k < l \leq n$ such that

1. $\pi[k] < \pi[i] + 1 = \pi[l] < \pi[j]$; or
2. $\pi[j] < \pi[i] = \pi[l] + 1 < \pi[k]$

Thus π is a Baxter permutation if and only if whenever there is a sub-sequence matching the pattern 3142 or 2413, then the absolute difference between the first and last element of the sub-sequence is always greater than 1. For example 2413 is not Baxter as the absolute difference between 2 and 3 is 1 and 41352 is Baxter even though the sub-sequence 4152 matches the pattern 3142 but the absolute difference between first and last of the sub-sequence is $|4-2| = 2 > 1$.

1.3.4 Algorithm FP2BP

Ackerman *et al.* (2006) showed the existence of a direct bijection between mosaic floorplans with n rooms and Baxter permutations of length n. They did this by providing two algorithms, one which takes a mosaic floorplan and produces the corresponding Baxter permutation and another which takes a Baxter permutation and produces the corresponding mosaic floorplan. To explain the algorithm the following operations on a mosaic floorplan are defined.

Definition 1 (Top-Left Block Deletion)**.** Let f be a mosaic floorplan with $n > 1$ blocks and let b be the top left block in f . If the bottom-right corner of b is a $\dashv$ (resp., $\perp$) junction, then one can delete b from f by shifting its bottom(resp., right) edge upwards(resp., leftwards), while pulling the T-junctions attached to it until the edge hits the bounding rectangle.

Definition 2 (Bottom-Left Block Deletion)**.** Let f be a mosaic floorplan with $n > 1$ blocks and let b be the bottom left block in f . If the top-right corner of b is a $\dashv$ (resp., T) junction, then one can delete b from f by shifting its top(resp., right) edge downwards(resp., leftwards), while pulling the T-junctions attached to it until the edge hits the bounding rectangle.

Note that the above deletion procedures do not change the relative ordering among the remaining blocks.

Input : A mosaic floorplan f with n blocks
Output: A (Baxter) permutation of length n
Label the rooms in their top-left deletion order from $\{1, \ldots, n\}$;
Obtain the permutation by arranging the room labels in their bottom-left deletion order ;

Algorithm 1: Algorithm FP2BP

The algorithm captures all the information contained in a mosaic floorplan. Given any two blocks in the floorplan, by looking at their labels and relative positions in the permutation it can be exactly decided which one of the *above*, *below*, *left* or *right* relations hold between those two blocks. This is possible because the labeling is done starting from the top-left corner and extraction of the permutation from the other diagonal, that is starting from bottom-left corner. The following two lemmas prove why this is true.

Lemma 1 (Ackerman *et al.* (2006))**.** If a block b_1 precedes a block b_2 in both top-left deletion ordering and bottom-left deletion order then b_1 is to the left of b_2.

Proof. If a block b_1 precedes a block b_2 in top-left deletion ordering implies that there is an intermediate floorplan obtained by deleting the top-left blocks successively, where b_1 is the top-left block and b_2 is contained within that floorplan. Hence b_1 is either *above* or to the *left* of b_2. Since b_1 precedes b_2 in bottom-left deletion ordering there is an intermediate floorplan where b_1 is the bottom-left block and b_2 is contained within that floorplan. Hence b_1 is either *below* or to the *left* of b_2. In a mosaic floorplan a block b_1 cannot be both *above* and *below* block b_2. Hence b_1 is to the left of b_2. □

The following lemma can be proved in a similar fashion.

Lemma 2 (Ackerman *et al.* (2006))**.** If a block b_1 precedes a block b_2 in top-left deletion ordering. But in the bottom-left deletion order b_2 precedes b_1 then b_1 is *above* b_2.

From the permutation π corresponding to a mosaic floorplan f the relative position of any block with respect to any other can be decided. This is because the blocks are labeled from $\{1, \ldots, n\}$ in their top-left deletion order and these labels corresponding to blocks appear in the permutation π in the same order as their bottom-left deletion order.

The action of the algorithm on a mosaic floorplan is illustrated by the figures 1.7 and 1.8. To illustrate how it captures the relative placement among blocks let us take blocks labeled 3 and 2 in the Figure 1.8. Since $2 < 3$ block 2 is either to the left of or above 3. Since 3 appears before 2 in the permutation corresponding to the floorplan block 3 is either to the left of or below 2, or in other words block 2 is either to the right of or above 3. But since block 2 can not be both to the right and left of block 3 in a mosaic floorplan block 2 must be above block 3, which indeed is the case in the given floorplan. The permutation thus obtained is called the **Abe-label** of the corresponding floorplan. Ackerman *et al.* (2006) also proved that the family of permutations produced by the algorithm is the class of Baxter permutations. Intuitively this is because the mosaic

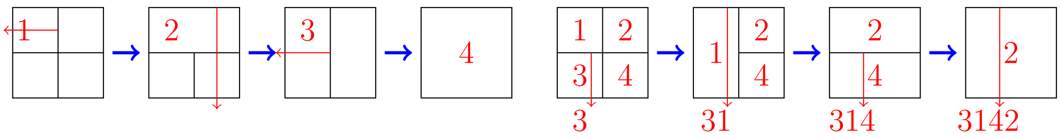


Figure 1.5: Non-Mosaic floorplan producing 3142, a non Baxter permutation

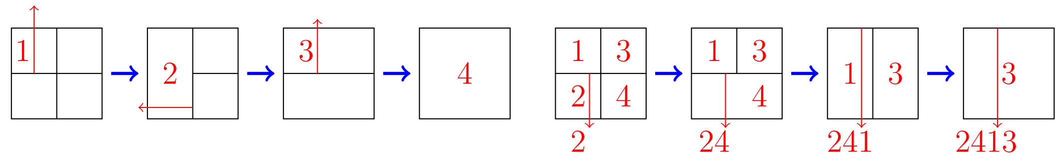


Figure 1.6: Non-Mosaic floorplan producing 2413, a non Baxter permutation

floorplan by definition cannot have any cross junctions and if there are cross junctions in the floorplans then the algorithm will produce a permutation having a pattern which is forbidden for Baxter permutations. To see this let us take a simple non-mosaic floorplan with 4 rooms. This floorplan has two permutations corresponding to it (illustrated in figure 1.5 and figure 1.6). These permutations are 2413 and 3142 respectively, which are also the forbidden patterns of Baxter permutations. Hence intuitively it must be the case that wherever there is a cross junction in the floorplan in the corresponding permutation either 2413 or 3142 must appear as a pattern in the corresponding sub-sequence. For a more formal proof of equivalence between mosaic floorplans and Baxter permutations see Ackerman *et al.* (2006).

1.3.5 Slicing Floorplans

A floorplan is called a slicing floorplan if it can be obtained from a rectangle by dissecting it recursively horizontally or vertically. All the floorplans in Figure 1.9 are slicing floorplans.

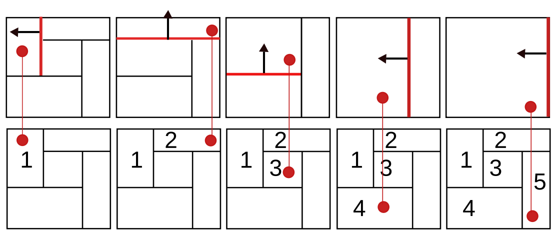


Figure 1.7: FP2BP Labeling Phase

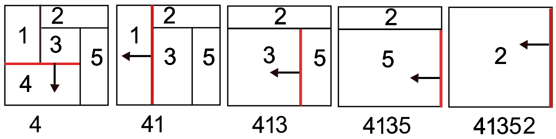


Figure 1.8: FP2BP Extraction of permutation Phase

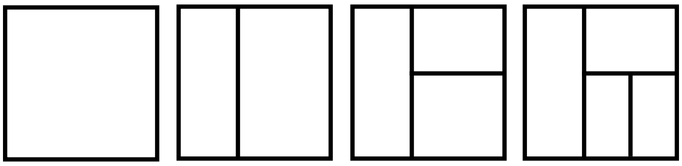


Figure 1.9: Slicing Floorplans

1.3.6 Slicing Tree

A slicing floorplan can be represented by a rooted tree called slicing tree(Wong and Liu (1986)). A slicing tree is a **rooted binary tree** with the following properties:

- Every internal node is labeled either V or H representing the vertical and horizontal slice respectively.
- Each leaf node corresponds to a basic room in the final floorplan.
- The sub-tree rooted at left child(resp., right child) of of a V node represents the floorplan contained in the left half(resp., right half) of the room which was cut vertically into two.
- The sub-tree rooted at left child(resp., right child) of of an H node represents the floorplan contained in the lower half(resp., upper half) of the room which was cut vertically into two.

A slicing tree captures the order in which the basic rectangle was divided recursively to obtain the final floorplan. But as shown in figure 1.10 there can be multiple slicing trees corresponding to the same floorplan. To avoid this problem we define a sub-class of slicing trees called **skewed slicing trees** which are essentially slicing trees which also obey the following rule:

- An internal node (labeled from $\{V, H\}$) and its left child cannot have the same label.

This rule produces a unique tree corresponding to slicing floorplan by eliminating symmetry associated with horizontal and vertical cuts by ensuring that always the first operation from left to right and top to bottom is the parent node at that level. This is achieved by the extra rule above as it says that a V node has to have an H node or a

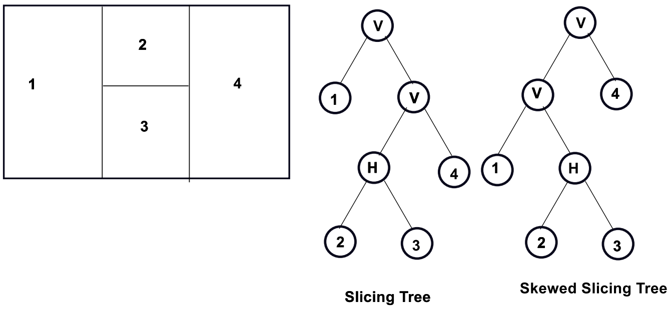


Figure 1.10: Slicing Trees and Skewed Slicing Trees

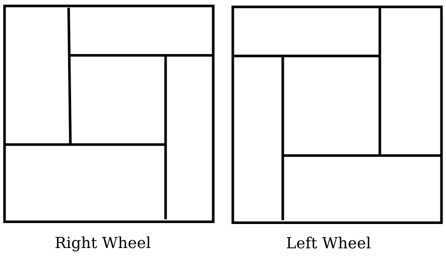


Figure 1.11: Wheels

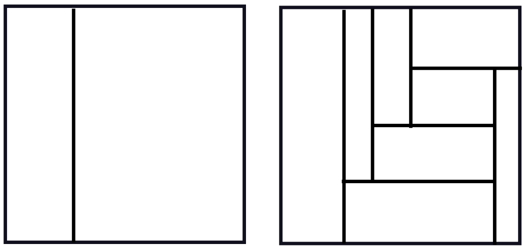
leaf as the right child, because if the V was not the first one from left to right then V ought to have another V as its right child. Similarly symmetry associated with H is also removed by skewness.

1.3.7 HFO$_5$ floorplans

A floorplan is said to be hierarchical of order 5 if it can be obtained from a rectangle by recursively sub-dividing each rectangle into either two parts by a horizontal or a vertical line segment or into five parts by a wheel (which can be of two types as shown in the Figure 1.11).

1.3.8 Hierarchical Floorplans of Order k

We can generalize the concept of hierarchical floorplans of order 5 to any k by defining hierarchical floorplans of order k as all mosaic floorplans which can be obtained from a rectangle by recursively sub-dividing each rectangle into l parts ($l \leq k$) by embedding a mosaic floorplan with l rooms. It is easy to observe that when $k = 2$ this becomes the class of slicing floorplans and when $k = 5$ it becomes HFO$_5$. All floorplans in Figure 1.12 are HFO$_7$ floorplans by this definition.



(a) (b)

Figure 1.12: HFO_7 floorplans

$\boxed{5\ 3}\ 6\ 1\ 4\ 2$ $\quad 5\ \boxed{3\ 6}\ 1\ 4\ 2$ $\quad 5\ 3\ \boxed{6\ 1}\ 4\ 2$ $\quad 5\ 3\ 6\ \boxed{1\ 4}\ 2$ $\quad 5\ 3\ 6\ 1\ \boxed{4\ 2}$

$\boxed{5\ 3\ 6}\ 1\ 4\ 2$ $\quad 5\ \boxed{3\ 6\ 1}\ 4\ 2$ $\quad 5\ 3\ \boxed{6\ 1\ 4}\ 2$ $\quad 5\ 3\ 6\ \boxed{1\ 4\ 2}$

$\boxed{5\ 3\ 6\ 1}\ 4\ 2$ $\quad 5\ \boxed{3\ 6\ 1\ 4}\ 2$ $\quad 5\ 3\ \boxed{6\ 1\ 4\ 2}$

$\boxed{5\ 3\ 6\ 1\ 4}\ 2$ $\quad 5\ \boxed{3\ 6\ 1\ 4\ 2}$

Figure 1.13: Example of a simple permutation

1.3.9 Simple Permutations

A **block** in a permutation is a set of consecutive positions (called segments) which is mapped to a range of values. The trivial block of a permutation are singleton blocks and the block $1 \ldots n$. For example in the permutation $\pi = 3421$ segment $1 \ldots 3$ is a block as π maps $1 \ldots 3$ to $\{2, 3, 4\}$ which is a range but the segment $2 \ldots 4$ is not a block as it is mapped to $\{1, 2, 4\}$ which is not a range as 3 is missing. A permutation is called simple when all its blocks are trivial blocks. An example of a simple permutation is $\pi = 41352$. Also note that π above is the Abe-label of right rotating wheel. A one point deletion on a simple permutation $\pi \in S_n$is deletion of a single element at some index i and getting a new permutation $\pi' \in S_{n-1}$ by decreasing each element of π greater than $\pi[i]$ by one. For example one-point deletion at index 3 of 41352 gives the permutation 3142. Because after deletion of 3 from 41352 we get 4152, then decreasing the numbers which are greater than 3 in 4152 by one we obtain 3142. Figure 1.13 shows a simple a permutation and Figure 1.14 shows a permutation which is not simple.

$5\ \boxed{3\ 1\ 2}\ 4\ 6$

Figure 1.14: A non-simple permutation

1.3.10 Block Decomposition of a permutation

Simple permutations are an interesting class of permutations for the reason that arbitrary permutations can be built just using simple permutations. A **block decomposition** (Albert and Atkinson (2005)) of a permutation σ is a partition of the set of positions of σ into blocks. A block decomposition is non-trivial if there is at least one block which is non-trivial. Given the block decomposition of σ, its pattern is the permutation defined by the relative order of the blocks. For example 451362 has the non-trivial decomposition $(45)(1)(3)(6)(2)$ with the pattern of decomposition being 41352. We can think of 453162 being constructed from 41352 by inflating each of the elements $12, 1, 1, 1$ and 1 into blocks. This can be represented as wreath product of permutations as $451362 = 41352\,[12, 1, 1, 1, 1]$.

1.3.11 Exceptionally Simple Permutations

The following simple permutations are called **exceptional :**

$$246\ldots(2m)135\ldots(2m-1) \tag{1.1}$$

$$(2m-1)(2m-3)\ldots1(2m)(2m-2)\ldots2 \tag{1.2}$$

$$(m+1)1(m+2)2\ldots(2m)m \tag{1.3}$$

$$m(2m)(m-1)(2m-2)\ldots1(m+1) \tag{1.4}$$

They are called exceptionally simple because no one-point deletion of an exceptionally simple permutation can give a simple permutation. For example 246135 is an exceptionally simple permutation of length 6. If we delete 2 from our example we get 35124 which is not simple as the segment $3\ldots4$ is mapped to 12 hence is non-trivial block. It can be easily verified that every one point deletion from above permutation results in a non-trivial block. Figure 1.15 shows an exceptionally simple permutation and also illustrates all single point deletions of this permutation. You can verify from the figure that all of these single point deletions result in permutations which are not simple. The interesting thing about exceptionally simple permutations is that there is always a two point deletion which yields a simple permutation of length $n-2$ (Albert

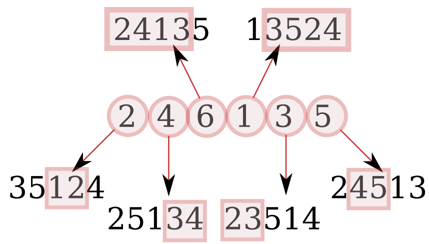


Figure 1.15: An exceptionally simple permutation

and Atkinson (2005)). For example upon deleting $1, 2$ from 246135 we get 2413, a simple permutation. Shmerl and Trotter(Schmerl and Trotter, 1993, 197) proved that there are no exceptionally simple permutations of odd length and also there are exceptionally simple permutations of even length for any even number greater than or equal to 4.

1.4 Contribution of the Thesis

We have obtained the following results related to HFO$_k$ floorplans:

1. We have proved that there exists an HFO$_k$ floorplan which is not an HFO$_j$ floorplan for any $j < k$, thus proving that they form an infinite hierarchy.

2. We obtained that the permutations corresponding to Abe-labeling of HFO$_k$ floorplans are those Baxter permutations which avoid patterns from Simple permutation of length $k + 1$ and exceptionally simple permutations of length $k + 2$.

3. We prove that HFO$_k$ floorplans are in bijective correspondence with skewed generating trees of order k.

4. We provide a linear-time algorithm for recognizing a permutation corresponding to an HFO$_k$ floorplan.

5. We prove that there are exponentially many HFO$_k$ floorplans with n rooms than there are HFO$_{k-1}$ floorplans for sufficiently large values of n.

6. We give a recurrence relation for the exact number of distinct HFO$_5$ floorplans with n rooms which can be easily extended to any k.

7. We give an $O(n^{k+1})$ time algorithm for generating the count of HFO$_k$ floorplans with n rooms.

Using the results $2, 3$ and/or the linear time algorithm for recognition one can design moves for the family of HFO$_k$ floorplans and build stochastic search algorithms on the family of HFO$_k$ floorplans.

We have obtained the following interesting results on Baxter permutations:

1. Baxter permutations are closed under inverse.
2. The floorplan corresponding to the inverse of a Baxter permutation is the floorplan obtained by taking the vertical flip of the floorplan corresponding to the original permutation.

1.5 Organization of Thesis

The thesis is organized into five chapters. The first and current chapter defines the problem then details existing literature related to this problem and explain the notations and definitions needed to explain our results. The rest of the thesis is organized as follows. Chapter 2 presents the work we have done on HFO$_k$ Floorplans, mainly characterization, counting and algorithm for recognition. Chapter 3 focuses on a special class of HFO$_k$ hierarchy, that is HFO$_5$ floorplans. We prove that they are in bijective correspondence with a class of trees (similar to the trees defined by Wong and The (1989)) and give a non-linear recurrence relation for the exact number of distinct HFO$_5$ floorplans using the bijection. Based on this recurrence we obtain a polynomial algorithm for generating the count. And then we generalizes these results for any k. Chapter 4 details some interesting properties of Baxter permutations that we have obtained which also relates to the properties of the mosaic floorplans corresponding to them. Finally, we conclude with future directions and open problems for further discussions on Hierarchical Floorplans of Order k.

CHAPTER 2

HFO$_k$ Floorplans

In this chapter we focus on the characterization, counting and algorithm for recognition for HFO$_k$ family of floorplans. We first prove that HFO$_k$ floorplans form a non-trivial hierarchy, that is there is an HFO$_k$ floorplan which is not HFO$_j$ for any $j < k$. Then we characterize floorplans of this type, that is HFO$_k$ floorplans with k rooms which are not HFO$_j$ for any $j < k$. We call such floorplans Uniquely HFO$_k$. Using this characterization we also characterize HFO$_k$ floorplans. Next we prove that HFO$_k$ floorplans are in bijective correspondence with a family of trees which we call skewed generating trees of order k. This bijection leads to a linear time algorithm for recognizing permutations corresponding to HFO$_k$ floorplans. We provide the algorithm and analysis of running time. For a floorplan family it is interesting to know the number of distinct floorplans in the family having n rooms. We prove that there are at least 3^{n-k} HFO$_k$ floorplans which are not HFO$_j$ for any $j < k$. But we were not able to obtain a closed form expression for the number of distinct HFO$_k$ floorplans with n rooms.

2.1 An Infinite Hierarchy

Hierarchical floorplans form an infinite hierarchy whose levels are HFO$_k$ floorplans for a specific value of k and it is such that each level has at least one floorplan which is not contained in the level below.

Theorem 3. *For any $k \geq 7$, $HFO_k \setminus HFO_{k-1} \neq \phi$.*

Proof. An HFO$_k$ floorplan which is not HFO$_j$ for $j < k$ by definition should be such that no proper subset of basic rectangles are contained in an enveloping rectangle, because if such a set of rectangles exist then it will be possible to construct this floorplan hierarchically by starting with the floorplan with such a set of rectangles removed.

We will first show that for any odd number $k \geq 7$ there is a hierarchical floorplan of order k which is not hierarchical floorplan of order j for any $j < k$. The proof is evident from the geometric construction given in Figure:2.1. The procedure is to start with an HFO_7 floorplan which is not HFO_j for any $j < 7$ shown in the left half of the figure 2.1 and take the vertical line segment supporting the left bottom basic rectangle then cut it half-way through as shown in the figure and insert a **T**-junction. It is easy to verify that the newly introduced basic rectangles do not create with its neighbouring rectangles a proper subset of basic rectangles which are contained in an enveloping rectangle. Hence the resulting floorplan is not contained in any lower levels of Hierarchy. The procedure increases the number of rooms in the floorplan by 2. Note that the in the floorplan obtained using the above procedure there exists a line-segment which touches the bounding box of all rectangles so that there are no parallel line-segments to its left. Hence the above procedure can be applied inductively to get an HFO_k having the above mentioned properties for k odd where the base case is the floorplan in figure 2.1.

We will show for the odd case how to transform this geometric construction into a proof by induction.

Induction Hypothesis For any $2k + 1 \geq 7$ there exists an HFO_{2k+1} floorplan such that it bottom-left corner has three rooms as shown in figure 2.3 and also it is not contained in any lower levels of the hierarchy.

Base Case :$2k + 1 = 7$ Floorplan in figure 2.1 serves as the base case as it satisfies both the properties.

Induction Step : We are guaranteed existence of an HFO_{2k-1} which satisfies the desired properties by induction hypothesis. Now will show to how to construct an HFO_{2k+1} satisfying the desired properties from the floorplan on $2k - 1$ rooms guaranteed by the induction hypothesis. As shown in figure 2.4 we transform the bottom left corner. This increases the number of rooms by 2, so the total number of rooms is $2k - 1 + 2 = 2k + 1$. It remains to prove that the floorplan thus obtained is not contained in any lower levels of the hierarchy. Since we are changing only the three rooms near bottom left corner, if at all there is an enveloping rectangle containing a proper set of basic rectangles then they must involve the newly introduced two rooms. It is easy to verify that these two rooms does not form such a rectangle with neighbouring three rectangles, hence if such an enveloping rectangle exists then other than these two rooms it must involve these neighbouring three rooms but this would imply that such an enveloping rectangle is there in the HFO_{2k-1} floorplan guaranteed by the induction

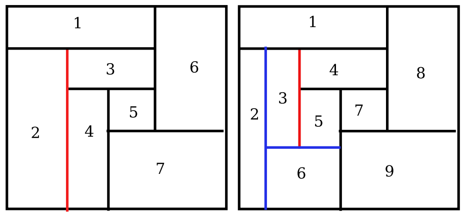


Figure 2.1: Constructing HFO_9 from HFO_7

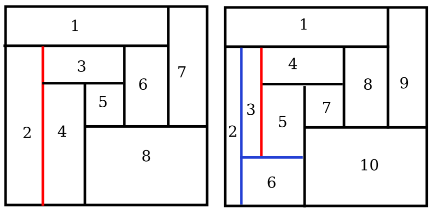


Figure 2.2: Constructing HFO_{10} from HFO_8

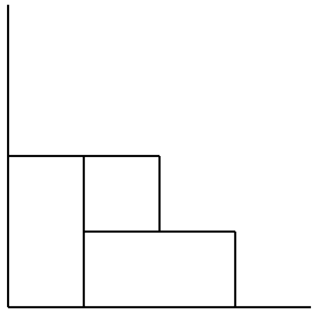


Figure 2.3: Bottom-left corner of an HFO_{2k+1} which is not contained in any $\text{HFO}_j, j < 2k+1$

hypothesis. But this is impossible. Hence the newly obtained floorplan is not contained in any lower levels of the hierarchy.

For an even $k \geq 8$ we use the same proof technique but we start from an HFO_8 which is not an HFO_j for any $j < 8$. Figure:2.2 demonstrates the construction. The construction can applied recursively to prove the existence of an HFO_k which not HFO_j for any $j < k$ and k even. □

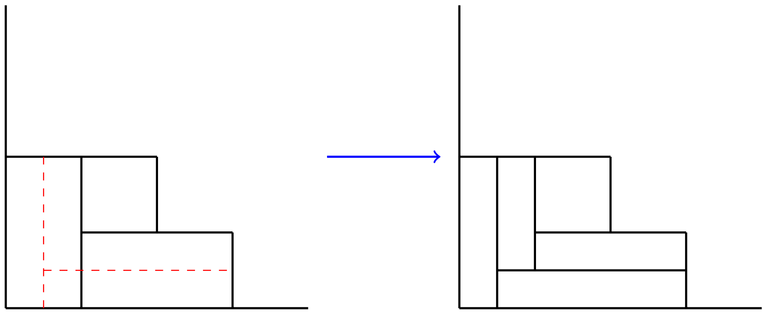


Figure 2.4: Obtaining HFO_{2k+1} from HFO_{2k-1}

2.2 Characterization of HFO$_k$ permutations

2.2.1 Uniquely HFO$_k$

We call an HFO$_k$ floorplan with k-rooms which is not a HFO$_j$ floorplan for any $j < k$, Uniquely HFO$_k$. We will prove that they are in bijective correspondence with those Baxter permutations of length k which are simple permutations of length k. Given an HFO$_k$ floorplan with k rooms if there is a set of $j, 1 < j < k$ basic blocks which are contained in an enveloping rectangle then it is possible to generate this floorplan in the following way. Consider the floorplan obtained by replacing these j basic blocks by just the enveloping rectangle and then place the HFO$_j$ floorplan formed by these basic rectangles inside that room. Hence it is clear that the resulting floorplan belongs to HFO$_{\max\{k-j,j\}}$, and since both $k-j$ and j are strictly greater than one we get that the resulting floorplan is not Uniquely HFO$_k$. If a non-trivial set of at most $k-1$ basic rectangles cannot be found in the mosaic floorplan then it is a Uniquely HFO$_k$ floorplan. We need the following crucial observation for formal proof of the characterization.

Observation 1. In the permutation π produced by the FP2BP algorithm run on a floorplan f if there exists **block**[1] then there is an enveloping rectangle containing the rooms labeled by the numbers in the block and nothing else, in f.

Figure 2.5 illustrates the intuition behind the observation. In the floorplan of Figure 2.5 rooms 1 and 5 together does not form a rectangular shape. The values 1 and 5 appear in the corresponding Abe-label in contiguous positions. But the $\{1, 5\}$ does not form a block of the permutation because values $2, 3$ and 4 are missing between 1 and 5. The positions $2 \ldots 4$ contain values from $3 \ldots 5$ and hence constitutes a block of the permutation. The rooms corresponding to the elements of this block, $3, 4$ and 5 together form a rectangular shape in the floorplan.

Proof. Let π be the Baxter permutation produced by algorithm FP2BP when run on the mosaic floorplan f. Suppose there is a **block** at consecutive positions $i, \ldots, j$ in π. If the block is a trivial **block**, then the observation is correct as there will be either just one number in the block or all the numbers from $1 \ldots n$ and in both cases rectangles labeled by the numbers in the block are contained inside trivial enveloping rectangles.

[1] See 1.3.9 for the definition of a block

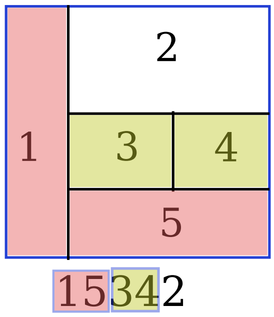


Figure 2.5: Blocks of the permutation and corresponding rooms in the floorplan

The remaining case is that the block is a non-trivial block. That is there is at least one number in $[n]$ which is not contained in the block. Since the basic blocks in a mosaic floorplan are rectangular in shape, if the rooms which are labeled by the numbers in the block do not form an enveloping rectangle it must be forming a shape with at least one **T** shaped corner or they form disconnected clusters. If the rectangles form disconnected clusters and if there is at at least one cluster with a **T** shaped corner then this reduces to the case that the shape formed by the basic rectangle has one **T** shaped corner. Hence all of them must be forming clusters which are rectangular in shape. Take any two such disconnected clusters and take all the basic rectangles between them, it is obvious that after labeling the top cluster the basic rectangles between two clusters will be labeled before reaching the second cluster since it is not connected to the first. Hence it contradicts our assumption that the basic rectangles in consideration where labeled by elements in a **block** of a permutation as they do not form a range together. Hence it remains to prove that if there is **T** shaped corner in the shape formed by the basic rectangles labeled by the numbers in the block, it also leads to a contradiction. Since there are no empty rooms in a mosaic floorplan and the block is a non-trivial block there should be at least one basic rectangle adjacent to this **T** shaped corner which is labeled with a number not contained in the block. Let us consider case 1 in Figure 2.6 where basic rectangles 'a' and 'b' are part of the block in the permutation π whereas 'c' is not. In this case it is clear that among these three the algorithm will label 'a' first, 'c' second and label 'b' the last. Hence it contradicts our assumption that there exists a block in π containing labels of 'a' and 'b' but not 'c' as the label corresponding to 'c' will be a number between the labels of 'a' and 'b'. Hence this case is not possible. Let us consider case 2 in Figure 2.6, again 'a' and 'b' are part of the assumed block in π whereas 'c' is not. Here the order in which the basic rectangles 'a','b','c' will be deleted is: 'b' first, 'c' the second and 'a' the last. Hence it contradicts our assumption

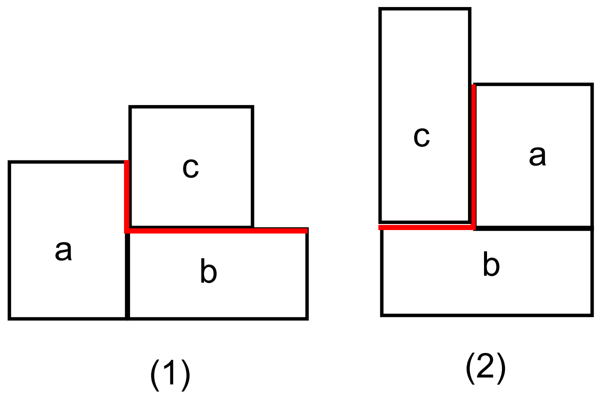


Figure 2.6: T-shaped corners

that there is a block in π containing 'a' and 'b' but not 'c' as in π label of 'c' will appear in between labels of 'a' and 'b'. Similarly it can be proved that any such **T**-corner configuration will result in a contradiction to our assumption that there is a block in π, such that the rooms labeled by the numbers in that block is not contained inside an enveloping rectangle in the corresponding mosaic floorplan. Hence the observation. □

Now we will prove the characterization of Uniquely HFO$_k$ floorplans based on the permutations corresponding to them.

Theorem 4. *Uniquely HFO$_k$ floorplans are in bijective correspondence with permutations of length k which are both Baxter and Simple.*

Proof. The bijection is the bijection described by Ackerman *et al.* (2006) from mosaic floorplans to Baxter permutations, restricted to Uniquely HFO$_k$ floorplans. Since Uniquely HFO$_k$ permutations are a subclass of HFO$_k$ permutations which are in-turn a subclass of mosaic floorplans we know that Uniquely HFO$_k$ floorplans correspond to a sub-family of Baxter permutations. So it remains to prove that they are also a sub-family of simple permutations of length k. Suppose π is the Abe-label of a Uniquely HFO$_k$ floorplan which is not a simple permutation, then there exists a non-trivial block in π consisting of $j, 1 < j < k$ numbers. By observation 1 there is an enveloping rectangle containing just the rooms which are labeled by the numbers in the non-trivial block. Now we can obtain the HFO$_k$ floorplan corresponding to π, by removing the rooms labeled by numbers in the non-trivial block and then placing the mosaic floorplan constituted by the rooms labeled by the numbers in the non-trivial block of π. Thus the floorplan is in HFO$_{\max\{k-j,j\}}$ contradicting our assumption that it is Uniquely HFO$_k$.

Hence the Abe-label corresponding to a Uniquely HFO$_k$ permutation has to be a simple permutation of length k. $\square$

2.2.2 Generating trees of Order k

A generating tree for a mosaic floorplan is a rooted tree which represents how the basic rectangle was embedded with successive mosaic floorplans to obtain the final floorplan. A generating tree is called a generating tree of order k if it satisfies the following properties:

- All internal nodes are of degree at most k.
- Each internal node is labeled by a Uniquely HFO$_l$ permutation($l \leq k$), representing the mosaic floorplan which was embedded.
- Out degree of a node whose label is a permutation of length l is l.
- Each leaf node represents a basic room in the final floorplan and is labeled by the Abe-label of that room in the floorplan.

The internal nodes are labeled by permutations corresponding to Uniquely HFO$_l$ floorplans because they are the only HFO$_l$ floorplans which cannot be constructed hierarchically with HFO$_j$ floorplans for $j < l$. By this definition there is at least one generating tree of order k for any HFO$_k$ floorplan. But the problem is that due to the symmetry associated with vertical and horizontal cut operations there could be multiple generating trees representing the same floorplan. To avoid this problem we define skewed generating trees. An order k generating tree is called a **skewed generating tree** of order k if it satisfies additional to the above rules the following rule:

- The right child of a node cannot be labeled the same as parent if the parent is labeled from $\{12, 21\}$.

Theorem 5. *HFO$_k$ floorplans with n rooms are in bijective correspondence with skewed generating trees of order k with n leaves.*

Clearly the additional rule introduced above removes the symmetry associated with vertical(permutation 21) and horizontal(permutation 12) cut operations as we have seen in Slicing trees. Hence it remains to prove that for any other embedding such a symmetry doesn't exist thus making the skewed generating tree unique for an HFO$_k$ floorplan.

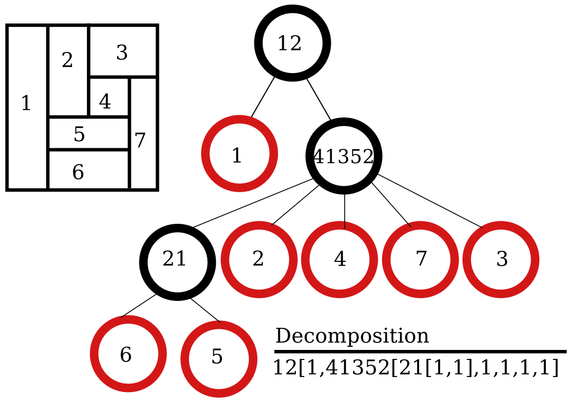


Figure 2.7: Generating Trees of Order k

Note that the generating tree provides a hierarchical decomposition of the permutation corresponding to the floorplan into blocks as illustrated by the figure 2.7. Albert and Atkinson Albert and Atkinson (2005)proved the following :

Theorem 6 (M.H Albert, M.D Atkinson)**.** *For every non-singleton permutation π there exists a unique simple non-singleton permutation σ and permutations $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \ldots, \alpha_n$ such that*

$$\pi = \sigma[\alpha_1, \alpha_2, \alpha_3, \alpha_4, \ldots, \alpha_n]$$

Moreover if $\sigma \neq 12, 21$ then $\alpha_1, \alpha_2, \alpha_3, \alpha_4, \ldots, \alpha_n$ are also uniquely determined. If $\sigma = 12$(respectively 21) then α_1 and α_2 are also uniquely determined subject to the additional condition that α_1 cannot be written as $(12)[\beta, \gamma]$(respectively as $(21)[\beta, \gamma]$.

The proof is completed by noting that the decomposition obtained by skewed generating tree of order k satisfies the properties of their decomposition, that is if $\sigma = 12/21$ its right child cannot be $12/21$ hence the block corresponding to the right child, α_1 cannot be $(12)[\beta, \gamma]/(21)[\beta, \gamma]$. Since such a decomposition is unique the skewed generating tree also must be unique. Hence the theorem. This bijection between HFO$_k$ floorplans is very crucial for the characterization of HFO$_k$ floorplans in terms of permutations corresponding to it and also for getting a coded representation of HFO$_k$ floorplans for stochastic search methods.

2.2.3 Characterization of HFO$_k$

Theorem 7. *HFO$_k$ floorplans with n rooms is in bijective correspondence with Baxter permutations of length n which avoids patterns from Simple permutations of length $k+1$ and Exceptionally simple permutations of length $k+2$.*

Proof. The bijection is the bijection defined by Ackerman *et al.* (2006) from mosaic floorplans to Baxter permutations restricted to HFO$_k$ floorplans. Since HFO$_k$ is sub-class of mosaic floorplans the bijection will map them to a sub-class of Baxter permutations. It is easy to prove that if a permutation corresponds to an HFO$_k$ floorplan then it cannot contain text which matches patterns from Simple permutations of length $k+1$ and exceptionally simple permutation of length $k+2$. Suppose in the permutation π corresponding to an HFO$_k$ floorplan there is text at comprising of points $(i_1, i_2, i_3, i_4, \ldots, i_j)$ which matches a simple permutation σ of length $j, j > k$. Then in the generating-tree of order k of the HFO$_k$ floorplan corresponding to the given permutation π, it is clear that no proper subset of $\{\pi[i_m] | 1 \leq m \leq j\}$ could be inside a single sub-tree because in the generating tree the elements of the sub-tree will always form a range(root of node of the subtree corresponds to the enveloping rectangle of all leaf nodes in the subtree) and no proper subset of a simple permutation can form a range. Consider the smallest(in the number of vertices) subtree which contains all of $\{\pi[i_m] | 1 \leq m \leq j\}$. In this subtree let the root node be r and let its children be $\{r_1, r_2, r_3, \ldots, r_l\}$. None of the subtrees rooted at $r_i, 1 \leq i \leq l$ can contain all of $\{\pi[i_m] | 1 \leq m \leq j\}$ because then r_i will be the smallest subtree containing all of $\{\pi[i_m] | 1 \leq m \leq j\}$. And for the above mentioned reason no $r_i, 1 \leq i \leq l$ can contain a proper subset of $\{\pi[i_m] | 1 \leq m \leq j\}$. Hence there should be j children of r, each containing exactly one node from $\{\pi[i_m] | 1 \leq m \leq j\}$. Hence there are at least j children for the root. Since $j > k$ this leads to a contradiction to our assumption that the permutation corresponds to an HFO$_k$ floorplan because there can no internal node of degree strictly greater than k in a generating tree of order k. . So it remains to prove that any HFO$_l$, $l > k$ floorplan which is not HFO$_k$ will contain a text matching the patterns from either Simple permutations of length $k+1$ or $k+2$. Let the floorplan be HFO$_l$ for $l > k$ and which is not HFO$_k$, and l be the smallest such integer that the floorplan is HFO$_l$. That is in the floorplan tree for this floorplan there is an internal node with out-degree l. This node will correspond to a Uniquely HFO$_l$ permutation and the ranges formed by subtrees rooted at the children of this node will be form the pattern which is the Uniquely HFO$_l$ permutation corresponding to the root node. Hence to obtain the **text** matching the **pattern** in the permutation corresponding to the floorplan we can pick one arbitrary leaf node from each subtree and then choose the Abe-label of that node. Hence every simple permutation of length l contains a pattern from either simple permutations of length $l-1$ when the original permutation is not exceptionally simple or simple permutations of length $l-2$ when the original permutation is exceptionally simple as deletion of an element from a permutation preserves the relative ordering among the other elements of the permutation. So we can find in an HFO$_l$, $l > k$ permutation a pattern which is a simple permutation of length $k+1$ or $k+2$ by applying the above observation recursively. □

2.3 Algorithm for Recognition

The algorithm is based on the bijection we obtained above. If a given permutation is Baxter then it is HFO$_j$ for some j. Suppose it is HFO$_k$ then we know that there exits an order k generating tree corresponding to the permutation. And in a generating tree of order k the label of leaves of any sub-tree will always form a range as the root of the sub-tree is an enveloping rectangle which contains all the rooms corresponding to the leaves. The algorithm 2 tries to iteratively reduce the sub-trees of the generating tree to nodes, level by level.

We will prove the correctness of the algorithm by use of the following loop invariant.

Loop Invariant: At the end of each iteration of the **for loop** of lines 2-13, all sub-trees of the generating tree containing leaf nodes which are labeled only from $\{\pi[j] | 1 \leq j \leq i\}$ are replaced with a single node(correspondingly pushed onto the stack as a range of numbers which are labels of the leaf nodes of the sub-tree).

Initialization: When $i = 1$, $\{\pi[j] | 1 \leq j \leq i\}$ is equal to $\pi[1]$. Since all internal nodes are of out-degree 2 or more the only sub-tree containing only $\pi[1]$ is the leaf node itself so there is nothing to be reduced hence the condition is trivially met.

Maintenance: We will assume that all the sub-trees whose leaves are labeled from $\{\pi[j] | 1 \leq j \leq i\}$ is reduced to a node before iteration $i+1$ and then prove that at iteration $i+1$ the condition is maintained by the **for loop** for all sub-trees whose leaves are labeled from $\{\pi[j] | 1 \leq j \leq i+1\}$. Suppose if there is a sub-tree whose leaves are labeled only from $\{\pi[j] | 1 \leq j \leq i+1\}$ and does not contain $\pi[i+1]$ then by the induction hypothesis it has been reduced to a node. Suppose there exists sub-trees which also contains $\pi[i+1]$ as a leaf node then choose the sub-tree which has $\pi[i+1]$ as an immediate child node. In this sub-tree all its other children are reduced to nodes by induction hypothesis, so at iteration $i+1$ there must exist $j \leq k$ elements at the top of the stack corresponding to the children of this sub-tree as it has $\pi[i+1]$ as the right most leaf node which also is the current stack top. Now the algorithm will reduce those j elements to a range and then try to reduce the tree further by scanning the top k elements of the stack.

Termination: When $i = n$ the tree itself is a sub-tree containing leaf nodes labeled from $[n]$ hence it must be reduced to a single node. Hence if at the end of the algorithm the stack contains just one element that would mean that the given permutation is HFO$_k$. Suppose the algorithm is able to reduce it to a single element on the stack then by retracing the stack operations carried out by the algorithm we can build an order k generating tree as at any point of time we merged at most k elements together which together formed a range and was a Baxter permutation(thus correspond to a mosaic floorplan). Hence upon acceptance by the algorithm for a given permutation it is clear that there is an order k generating tree corresponding to the given permutation. If the permutation is not HFO$_k$ algorithm would not be able to find a generating tree of order k. Hence it would reject such a permutation.

Figure 2.8 illustrates the working of the algorithm on an HFO$_5$ permutation. The figure shows the generating tree of order 5 corresponding to the floorplan, and trace of the stack used by the algorithm (to be read from left). The permutation is scanned from left to right and each time an insertion takes place in the stack, the top 5 elements are searched to see if they form a range. In the example shown in the figure until 3 is inserted onto the stack this doesn't happen. At the instant 3 is inserted it is combined with the other four elements to a range corresponding to the internal node labeled 41352 in the generating tree. Then this is combined with 1 to form another range and finally it is reduced to a single node by combining with 7. This final node corresponds to the root node of the generating tree.

```
Input: A permutation π of length n
Stack S ← φ;
for i = 1 to n do
    S.push(Range(π[i]));
    while There exists a j, j ≤k such that j is the least such number for which top j
    elements of S form a range do
        if S [top . . . (top − j)] is a Baxter permutation then
            R = Range(S [top . . . (top-j )]);
            for l =j downto 1 do
                S.pop();
            end
            S.push(R);
        end
    end
end
if S.size() = 1 then
    Accept;
else
    Reject;
end
```

Algorithm 2: Algorithm for checking if a permutation is HFO_k

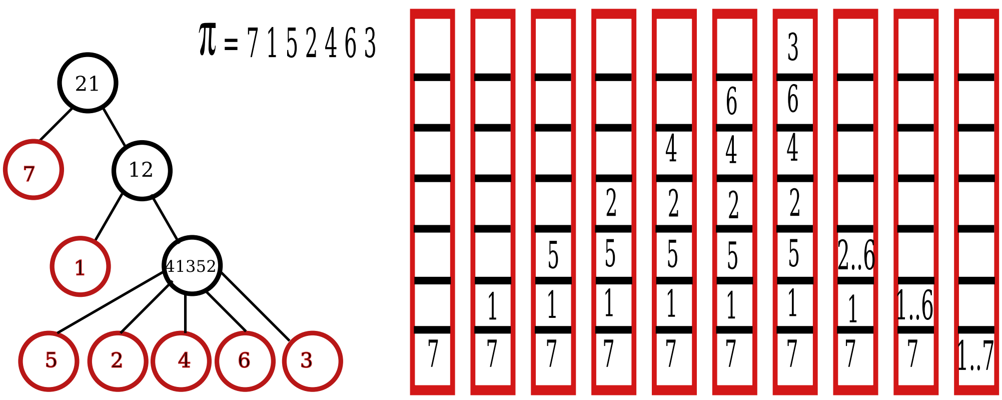


Figure 2.8: Example : HFO_5 recognition algorithm

2.3.1 Analysis of the recognition algorithm

The algorithm runs in both linear time and space for a fixed predetermined value of k which does not change with the input length. Linear space is easy to observe as the stack at any point of the execution of the algorithm contains no more than n elements. To prove that the algorithm runs in linear time we assign an amortized cost of k^2-units to each node(including leaf nodes) in the generating tree. We also use the observation that in a tree of n nodes there can not be more than $n-1$ internal nodes. Hence the total nodes of the generating tree is bounded by $2n-1$. So if the algorithm spends at most k^2 units of time with each node then the total time taken by the algorithm is $O(n)$.

Now we will prove that the algorithm spends at most k^2 units of time with each node in the skewed generating tree of order k corresponding to the permutation, if the permutation is HFO_k. The key operation in the algorithm is scanning the top k elements of the stack to find a set of elements which form a range. It is easy to observe that the stack is scanned only when a new element is inserted onto the top of the stack. The newly inserted element can either be a number in the permutation(which corresponds to a leaf node in the order k generating tree) or a range of elements(which corresponds to an internal node in the order k generating tree). Also, observe that a node is inserted only once into the stack. And when a range corresponding to a node is inserted to the stack, it is either merged with the top $j, j < k$ elements of the stack to become another node or the top k elements of the stack are searched unsuccessfully and the node remains on the top of the stack. In both cases, at most k^2 units of time is spend. Because to check whether top i elements form a range, i units of time is needed, so doing this for all $1 \leq i \leq k$ we need $\frac{k(k+1)}{2}$ time which is clearly upper bounded by k^2. Thus distributing the costs this way, we get that for each node in the tree at most k^2-units of time is spend. Since there are only $O(n)$ nodes in the tree the total time spend by the algorithm is $O(n)$.

If the permutation is not Baxter then at some point during the execution of the algorithm it will find a set of ranges on stack top which does not form a Baxter permutation, or the algorithm would not be able to merge the elements of the permutation to a single node. Even in this case the number of nodes in the partial tree which the algorithm can find with the given permutation is bounded by $2n$, and with each node at most k^2 units of computation will be spend. Hence in this case also the algorithm runs in linear time.

If the permutation is Baxter and is HFO$_j$ for some $j > k$ and is not HFO$_k$ then again the same cost analysis is valid and hence the algorithm runs in linear time for all possible types of input permutations.

Note that checking if a set of k elements form a range can be checked in constant time for a fixed value of k by writing conditional statements to check if the elements follow any of the $k!$ arrangements. We can also check if a set of k elements form a Baxter permutation for a fixed k in constant time by writing conditional statements to check if their rank ordering is equivalent to any one of the Baxter permutations of length k(whose number is bounded by number of permutations, $k!$). Hence the above algorithm runs in linear time for a predetermined value of k.

If the value k is unknown the same algorithm can be made to run in $O(n^2 \log_2 n)$ time to find out the minimum k for which the input permutation is HFO$_k$ with some simple modifications in the implementation. The first modification we have to implement is to make the algorithm checks if the input permutation π is Baxter permutation. If it is not it cannot be HFO$_j$ for any j hence is rejected. If it is a Baxter permutation then we know that it is HFO$_k$ for some $k \leq n$. And also at each time a new element is inserted onto the stack we have to check if that forms a range with any of the top $j, j \leq |S|$ elements of the stack where $|S|$ denotes the current size of the stack. Implementing these changes alone we obtain the modified Algorithm 3. The increase in running time comes from the fact that we don't know the value of the k, thus forcing us to scan the entire stack at the insertion of a new element on top of the stack costing us $cn \log_2 n$ time to sort the elements of the stack and see if there exists a $j, j \leq |S|$ such that the current element forms a range along with $S[top, \ldots, (top - j)]$. Checking if a permutation is Baxter takes $O(n^2)$ time. And we use the same amortized cost analysis as above but with each node(internal or leaf) in the tree we associate the cost of $cn \log_2 n$ which is spent at the time it is first inserted on to the stack for sorting the current elements of the stack. The number of nodes in the tree is again bounded by $2n$. Hence the stack reduction part of the algorithm runs in $O(n^2 \log_2 n)$ time and checking if a permutation is Baxter part runs in time $O(n^2)$. So the total time taken is $O(n^2 \log_2 n)$.

Input: A permutation π of length n

```
Stack S ← φ;
if π is not a Baxter permutation then
    Reject;
end
for i = 1 to n do
    S.push(Range(π[i]));
    mergedNodes ← true;
    repeat
        Array ← sort(S[top ..., 1)]);
        /*Find the longest range containing stack top */
        for i = 1 to n do
            if Array[i] = S[top] then
                Break;
            end
        end
        for j = i to n do
            /*This loop will run until Array[i,...,j] cease
              to become a range of contiguous elements */
            if Array[j].end() + 1 ≠ Array[j + 1].start() then
                Break;
            end
        end
        for l = i downto 2 do
            /*This loop will run until Array[l,...,i] cease
              to become a range of contiguous elements */
            if Array[l].start() − 1 ≠ Array[l − 1].end() then
                Break;
            end
        end
        /*Array[l,...,j] forms a range containing stack top,
          hence S[top,...,(top − (j − l))] forms a range */
        if j ≠ l then
            R ← Range(S[top,...,top − (j − l)]);
            for m = 1 to (j − l) do
                S.pop();
            end
            S.push(R);
        else
            mergedNodes ← false;
        end
    until mergedNodes = false;
end
if S.size() = 1 then
    Accept;
else
    Reject;
end
```

Algorithm 3: Algorithm for finding the minimum k for which π is HFO_k

2.4 Counting

Given an n, it is interesting to know the number of distinct HFO$_k$ floorplans with n rooms. We call two HFO$_k$ floorplans distinct in the same way Sakanushi *et al.* (2003) defines it. Given a floorplan f, a segment s supports a room r in f if s contains one of the edges of r. We say that s and r hold a *top-,left-,right-, or bottom-seg-room* relation if s supports r from the respective direction. Two floorplans are equivalent if there is a labeling of their rectangles and segments such that they hold the same seg-room relations, otherwise they are distinct. This is the same definition of equivalent floorplans Ackerman *et al.* (2006) used. Since we are considering a restriction of the bijection they gave between mosaic floorplans and Baxter permutations to HFO$_k$ floorplans, we can say that two HFO$_k$ floorplans are distinct if they are mapped to different permutations by this bijection. And by theorem-7 there is a bijection between HFO$_k$ floorplans and Baxter permutations which avoid patterns from simple permutations of length $k + 1$ and exceptionally simple permutations of length $k + 2$. Hence we give a lower bound on number of distinct HFO$_k$ floorplans on n rooms by giving a lower bound(resp., an upper bound) on the number of HFO$_k$ permutations.

Theorem 8. *There are at least 3^{n-k} HFO$_k$ permutations of length n which are not HFO$_j$ for $j < k$.*

Proof. The proof is inspired by the insertion vector scheme introduced by Chung *et al.* (1978) to enumerate the admissible arrangements for Baxter permutations. The idea is to start with a Uniquely HFO$_k$ permutation which is of length k say π_k and successively insert $(k + 1, k + 2, k + 3, k + 4, \ldots, n)$ onto it such a way that we are guaranteed that it remains both Baxter and no patterns from simple $k + 1$ or exceptionally simple $k + 2$ is introduced so the final permutation π_n is of the desired property. It is very clear that inserting $k + 1$ onto two different positions of π_k will result in two different permutations. It is also not hard to see that by starting with two different permutations π_i' and π_i'' there is no sequence of indices to which insertion of $(i+1, i+2, i+3, i+4, \ldots, n)$ will make the resulting permutations the same. Hence by counting the number of ways to insert $(k+1, k+2, k+3, k+4, \ldots, n)$ successively, a lower bound on the number of HFO$_k$ permutations is obtained. So the problem boils down to counting the number of ways to insert $i + 1$ given a permutation π_i which is HFO$_k$ but not HFO$_j$ for $j < k$. We do not have an exact count for this but it is easy to observe that in such a permutation π_i there are always four locations which are safe for insertion of $i+1$ irrespective of relative order of elements of π_i. By safe we mean that insertion of $i+1$ to π_i would neither make the resulting permutation non-Baxter nor will it make a non-simple permutation. The four safe locations are :

1. Before the first element of π_i.
2. After the last element of π_i.
3. Before i in π_i.
4. After i in π_i.

Let us prove that these sites are actually safe for insertion of $i+1$. We will first prove that insertion of $i+1$ onto these sites cannot introduce a pattern which matches a simple permutation of length $j, j > k$. Suppose $i+1$ is inserted before or after i in π_i and the newly obtained permutation has a **text** which matches a simple permutation of length $j, j > k$ .The **text** must be involving $i+1$ as otherwise π_i will also contain the same pattern. The text matching the pattern cannot involve i also, as if it does then the pattern will have two consecutive integers corresponding to the location of i and $i+1$ in the text making it not a simple permutation . Thus the text matching the pattern must involve $i+1$ and it must not involve i, but then replacing i by $i+1$ we get a text in π_i matching the same pattern contradicting our assumption that π_i is HFO$_k$. Now it remains to prove that inserting $i+1$ before or after π_i is safe. Suppose insertion of $i+1$ before or after π introduces a text matching a simple permutation of length $j > k$, then the text must involve $i+1$. But since $i+1$ is greater than any other element in π_i in the pattern of length j, $i+1$ will be matched with the number j. But then it would mean that pattern is a permutation σ on $[j]$ which has j as its first/last element as $i+1$ is placed after or before π_i. This contradicts our assumption that the pattern is a simple permutation as σ maps either $\{2, \ldots, j\}$ to $\{1, \ldots, j-1\}$ or $\{1, \ldots, j-1\}$ to $\{1, \ldots, j-1\}$ which is a proper sub-range. Hence it is not possible that insertion of $i+1$ onto any of these locations introduces a text matching a pattern from simple permutations of length $j, j > k$. Now it remains to prove that the insertion of $i+1$ cannot introduce any text which matches $3142/2413$ with absolute difference between first and last being one. Suppose it did, then it has to involve $i+1$ since π_i is Baxter, and if it involves $i+1$, $i+1$ will have to match 4 in $3142/2413$ as there is no element greater than $(i+1)$ in π_i. But $i+1$ matching 4 is not possible because in the first case there is nothing to the left of $i+1$, in the second case there is nothing to the right of $i+1$, and in third and fourth cases this is not possible for the reason that if $2413/3142$ involves both i and $i+1$ then i has to match 3 and $i+1$ has to match 4 as they are the second largest and largest elements in the new permutation but this is not possible in these cases as i is adjacent to $i+1$ and there cannot be any element matching 1 in between them. Hence in these cases the only possibility left is that $i+1$ is matched to 4 in $3142/2413$ but the text matching the pattern does not involve i and since i is adjacent to $i+1$ and greater than any element of π_i it can be replaced for $i+1$ to get $3142/2413$ in π_i with the absolute difference between first and last being one, contradicting the fact that π_i is Baxter. Hence we have proved that introduction of $i+1$ in these sites are safe.

Note that even though we have identified four safe locations for insertion of $i+1$ into a π_i sometimes i could be the first element of the permutation π_i thus making the location before i and location before π_i one and the same. Similarly if i is the last element the location after i and location after π_i also coincides. But for any permutation π_i only one of the above two conditions can occur, so there are always three distinct locations to insert $i+1$. Now by starting from a Uniquely HFO$_k$ permutation we can get 3^{n-k} different permutations by inserting successive elements from $\{k+1, k+2, k+3, k+4, \ldots, n\}$. Hence the theorem.

$\square$

CHAPTER 3

Simulated Annealing for HFO$_k$ family of floorplans

In this chapter we describe moves for stochastic search methods like simulated annealing, to define a neighbourhood relation on the family of corresponding codes. Wong and Liu (1986) designed a set moves for HFO$_2$ floorplans based on the post-order traversal of the the corresponding skewed order 2 generating tree. Later Wong and The (1989) extended this idea to HFO$_5$ floorplans. Now based on our result which unified the way HFO$_k$ floorplans are represented using generating trees we can easily extend the moves defined by Wong and The (1989) to HFO$_k$ floorplans. We also prove that the solution space thus obtained is connected and is of diameter $O(n^2)$. We also prove that our solution space is **P-admissible** except for the last property which requires the search space to include the optimal floorplan for a given floorplanning problem. Almost all of the solution spaces for floorplanning problem cannot guarantee this property. This is because the optimal solution to floorplanning problem may contain empty rooms and finding the number of optimal empty rooms for an instance of floorplanning problem is in itself an open problem.

3.1 Simulated Annealing moves for HFO$_5$ floorplans

Wong and The (1989) extended the idea of skewed slicing trees to skewed trees of order 5, which correspond to hierarchical floorplans of order 5. Skewed order 5 trees are essentially rooted trees with internal nodes having out-degree 2 or 5. The leaf nodes represents the rooms in the floorplan and are labeled from $\{1, \ldots, n\}$. Since the only non-slicing structures with at most 5 rooms are the two wheels(Figure 1.11) , the internal nodes are of four types. Internal nodes labeled by $*, +$ represent the vertical cut and horizontal cut respectively. Internal nodes labeled by ρ, σ represent the right rotating wheel and left rotating wheel respectively. It is called skewed because the left child of a node labeled from $*, +$ can not have the same label as the parent. The post-order traversal of the skewed order 5 tree is used to represent an HFO$_5$ floorplan. The polish

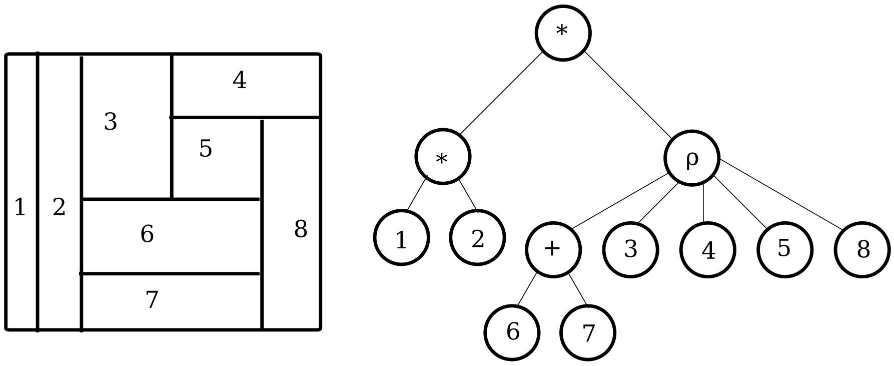


Figure 3.1: normalized 2-5 polish expression

expression thus obtained is called normalized $2-5$ polish expression. It is normalized because the skewness rule in the tree(that is left child of node labeled $*, +$ can not have the same label as the parent) will get reflected in the post-order traversal as a rule which disallows two consecutive operators in the corresponding polish expression to have the same label from $*, +$. Formally defined a normalized $2-5$ polish expression is a sequence $\alpha = \alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_m$ of elements from $\{1, 2, 3, \ldots, n, *, +, \rho, \sigma\}$ satisfying the following conditions. Let x_i, y_i and z_i represent the number of slicing operators $(*, +)$, the number of wheel operators(ρ, σ) and the number of operands respectively, in the sequence $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_i$.

- for each $j \leq n$ there exists a unique index k such that $\alpha_k = j$.
- $x_i + 4y_i < z_i$, for all $i = 1, 2, 3, \ldots, m$.
- $\alpha_i\alpha_{i+1} \neq **$ and $\alpha_i\alpha_{i+1} \neq ++$ for each i in $1, 2, 3, \ldots, m-1$.

Figure 3.1 shows the normalized $2-5$ polish expression corresponding to the HFO_5 floorplan in the figure.

The neighbourhood set for a normalized $2-5$ expression is given by defining the following set of moves on it.

1. **M1**: *Swap two adjacent elements*
   (a) operand $\longleftrightarrow$ operand
   e.g. $45312 * 2 + 6\mathbf{87}\rho* \rightarrow 4531 * 26\mathbf{78}\rho*$
   Two elements are *adjacent* if they are adjacent in the sequence obtained by removing all operators from the normalized $2-5$ expression.
   (b) operand $\longleftrightarrow$ operator
   e.g. $45312 * 2 + 68\mathbf{7}\rho* \rightarrow 4531 * 268\rho\mathbf{7}*$

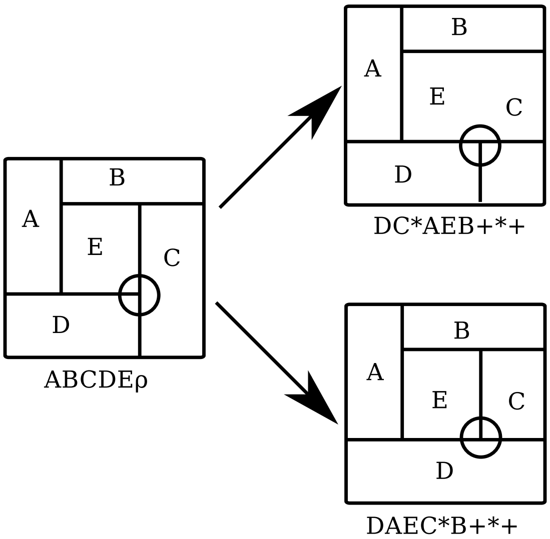


Figure 3.2: Destroying a wheel

(c) operator $\longleftrightarrow$ operator
e.g. $45312 * 2 + 687\rho* \rightarrow 4531 * 2687\rho*$
In this case it is required that at least one operator is a wheel operator.

2. **M2**: *Complimenting*
   (a) Complement a maximal chain
   e.g. $678123345\rho + *+ \rightarrow 678123345\rho * +*$
   A **maximal chain** is a sequence of slicing operators $\alpha_i, \alpha_{i+1}, \alpha_{i+2}, \ldots, \alpha_j$ in the given normalized $2-5$ expression α such that α_{i-1} and α_{j+1} if they exists should not be slicing operators.
   (b) *Complement a single wheel operator*
   e.g. $678123345\rho + *+ \rightarrow 678123345\sigma + *+$

3. **M3**: *Create a wheel*
   e.g. $678145 + 3 * 2 + * + * + * \rightarrow 67814532\rho + *+$
   Select a composite rectangle that can be partitioned into five basic/composite rectangles which are not arranged into a wheel from and re-arrange the five components into a wheel.

4. **M4**: *Destroy a wheel*
   e.g. $678123345\rho + *+ \rightarrow 678145 + 3 * 2 + * + *+$
   For each wheel, consider eight slicing floorplans that are close to the wheel. The eight slicing floorplans are obtained by two different ways of modifying the wheel at each of the four T-junctions. Figure 3.2 illustrates the process.

The moves **M2**,**M3** and *M4* always produces a normalized $2-5$ polish expression. It is not the case for **M1**, **M1(b)** and **M2(c)** might produce a sequence that violates condition 2 in the set of conditions for normalized $2-5$ polish expressions. But whether resulting expression is normalized or not can be checked efficiently.

Two normalized $2-5$ polish expressions are said to be neighbours if one can be obtained from the other via one of these four types of moves. While designing a simulated annealing algorithm one can select a neighbour by randomly choosing of one the four moves and then choosing the locations in the expression to apply the move.

3.2 Simulated Annealing Moves for HFO$_k$ floorplans

The moves described above can be easily generalized to any HFO$_k$ provided that it is possible to capture the floorplan using a floorplan tree , find out Uniquely HFO$_l$ floorplans for $l \leq k$ - the internal nodes of the tree and a nice representation of these floorplans to serve as operators in the normalized polish expression. We have already proved that HFO$_k$ floorplans are in bijective correspondence with skewed generating trees of order k. We also provided an algorithm (Algorithm 3) to find the minimum value of k for which a given Baxter permutation is also a permutation corresponding to an HFO$_k$ floorplan. Hence we can run this algorithm on all Baxter permutations of length k and find out permutations corresponding to Uniquely HFO$_k$ permutations because Uniquely HFO$_k$ are HFO$_k$ floorplans such that k is the minimum such integer for which they are HFO$_k$. The bijection given by Ackerman *et al.* (2006) can be used to represent a Uniquely HFO$_k$ floorplan as permutation of length k. We assume an implicit left-to-right ordering among the children of internal nodes in generating trees of order k and then use the post-order traversal of the tree to represent corresponding floorplan. To distinguish operators from operands we enclose permutation corresponding to Uniquely HFO$_k$ in set of [] parenthesis.

We will now formally define **normalized polish expressions of length** k which corresponds to postorder traversal of a skewed generating tree of order k. A normalized polish expression of length k is a sequence $\alpha = \alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_m$ of elements from $\{1, 2, 3, \ldots, n, \{[\pi] | \pi \in S_j, j \leq k\}\}$ satisfying the following conditions. Let x_{ji} represent the number of operators which are permutations of length j enclosed within [] brackets in the sequence $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_i$. And y_i represents the number of operands in the sequence $\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_i$.

- for each $j \leq n$ there exists a unique index k such that $\alpha_k = j$.
- $\sum_{j=2}^{k}(j-1)x_{ji} < y_i$, for all $i = 1, 2, 3, \ldots, m$.

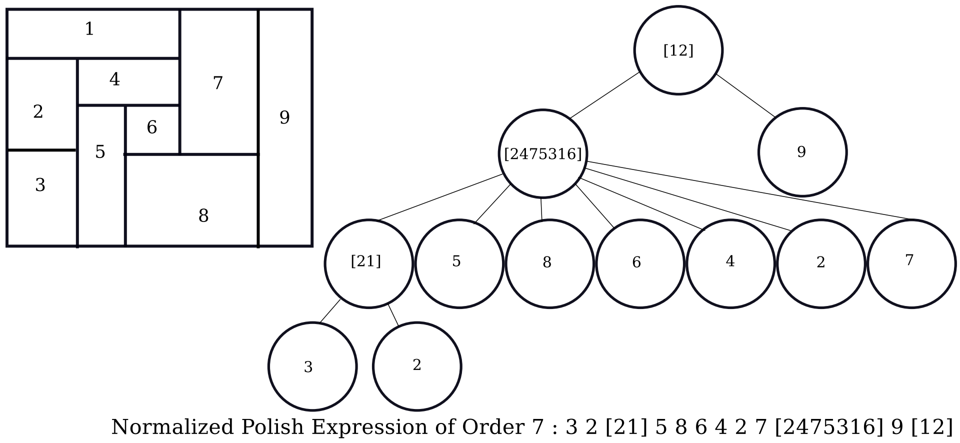


Figure 3.3: Example normalized polish expression of order k and corresponding floorplan

- $\alpha_i\alpha_{i+1} \neq [12][12]$ and $\alpha_i\alpha_{i+1} \neq [21][21]$ for each i in $1, 2, 3, \ldots, m-1$.

Figure 3.3 shows the normalized $2-5$ polish expression corresponding to the HFO$_7$ floorplan in the figure.

Now we will define moves on the Normalized Polish Expressions of order k to define the neighbourhood relationship amongst HFO$_k$ floorplans.

1. **M1**: *Swap two adjacent elements*
   (a) operand $\longleftrightarrow$ operand
   e.g. $45312[21]2[12]6\mathbf{87}[41352][21] \rightarrow 4531[21]26\mathbf{78}[41352][21]$
   Two elements are *adjacent* if they are adjacent in the sequence obtained by removing all operators from the normalized $2-5$ expression.
   (b) operand $\longleftrightarrow$ operator
   e.g. $45312[21]2[12]68\mathbf{7}[\mathbf{41352}][21] \rightarrow 4531[21]268[\mathbf{41352}]\mathbf{7}[21]$
   (c) operator $\longleftrightarrow$ operator
   e.g. $45312[21]2[12]687[41352][21] \rightarrow 4531[21]2687[41352][21]$
   In this case it is required that at most one operator is slicing($[12]/[21]$).

2. **M2**: *Complimenting*
   (a) Complement a maximal chain
   e.g. $67812345[25314][12][21][12] \rightarrow 67812345[25314][21][12][21]$
   A **maximal chain** is a sequence of slicing operators $\alpha_i, \alpha_{i+1}, \alpha_{i+2}, \ldots, \alpha_j$ in the given normalized expression α such that α_{i-1} and α_{j+1} if they exists should not be slicing operators.
   (b) *Complement an HFO$_j$ operator*
   In this operation take Uniquely HFO$_j$ operator in the given normalized polish expression and replace it with another Uniquely HFO$_j$ operator.
   e.g. $67812345[\mathbf{41352}][12][21][12] \rightarrow 67812345[\mathbf{25314}][12][21][12]$

3. **M3**: *Create an HFO$_j$ operator*

   e.g. 678145[12]3[12]2[12][21][12][21][12][21] $\to$ 67814532[41352][12][21][12]

   Select a composite rectangle that can be partitioned into j basic/composite rectangles which are not arranged into a Uniquely HFO$_j$ floorplan and re-arrange the j components into an Uniquely HFO$_j$ floorplan.

4. **M4**: *Destroy an HFO$_j$ operator*

   e.g. 32[21]586427[2475316]9[12] $\to$ 32[21]586427[41352][12][21]9[12]

   Here we differ slightly from Wong and The (1989). Since j can vary from 2 to k, the replacement policy is uniform. We replace a wheel operator like they do. But for destroying an HFO$_j$ operator for $j > 5$, we replace it with a wheel operator in the beginning and a chain of alternating slicing operators such that the normalization property is not violated.

The moves **M1**, **M1(b)** and **M2(c)** might produce a sequence that violates condition 2 in the set of conditions for normalized polish expressions expression of order k. But here also, checking whether resulting expression is normalized can be done efficiently.

Given a normalized polish expression of order k, it neighbours are all valid normalized polish expression which can be obtained by a single move from the list of moves above. It can be proved that the diameter of the solution space, that is the maximum distance between two valid normalized polish expressions of order k of length n, is $O(n^2)$. We prove this by observing that within $O(n)$-destroy HFO$_j$ operator moves, all the operators in the given expression can be made slicing operators. For each operator, with $O(n)$ operand-operator swap moves, it can be moved to the end of the expression. Hence within $O(n^2)$ steps any normalized expression of order k of length n can be transformed into an expression where all the operands are at the beginning and all the operators are at the end, and are slicing operators. The moves are defined such that if an expression can be obtained from another using a single move, there exists another moves which returns it back to the original. Hence we have proved existence of a normalized polish expression of order k which is a distance of $O(n^2)$ from any other expression. Hence between two normalized polish expressions of order k, there is a path of length $O(n^2)$ through this special node.

CHAPTER 4

Special Families - HFO$_5$

4.1 Introduction

Hierarchical Floorplans of Order 5 is the only HFO$_k$ other than slicing floorplans which have been studied in the literature to the best of our knowledge. Wong and The (1989) proved that they are in bijective correspondence with skewed order 5 trees and provided moves for simulated annealing based search on normalized $2-5$ polish expressions which are expressions which correspond to the post-order traversal of skewed order 5 trees. Similar to the way Shen and Chu (2003) obtained bounds on number of slicing floorplan by counting the number of skewed slicing trees we obtained a recurrence relation for the number of distinct HFO$_5$ floorplans with n rooms. Based on our result which proved that HFO$_k$ floorplans are in one-one correspondence with skewed generating trees of order k we claim that such a recurrence can be easily extended to HFO$_k$ floorplans for an arbitrary k.

4.2 Recurrence relation for the number HFO$_5$ floorplans

Since we have proved that the number of distinct HFO$_5$ floorplans with n rooms is equal to the number of distinct skewed generating trees of order 5 with n leaves(also proved by Wong and The (1989)) it suffices to count such trees . Let t_n denote the number of distinct skewed generating trees of order k with n leaves and $t_1 = 1$ representing a tree with a single node. Let a_n denote such trees whose root is labeled 12, b_n denote trees whose root is labeled 21, c_n denote trees whose root is labeled 41352 and d_n denote the trees whose root is labeled 25314. Since these are the only Uniquely HFO$_k$ permutations for $k \leq 5$ the root has to labeled by one of these. Hence

$$t_n = a_n + b_n + c_n + d_n$$

Since it is a skewed tree if the root is labeled 12, its left child cannot be 12 but it can be 12,41352 ,25314 or a leaf node. Similarly if the root is labeled 21 its left child cannot be 21 but it can be 12,41352 ,25314 or a leaf node. But for trees whose roots are labeled 41352/25314 can have any label for any of the five children. Hence we get,

$$\begin{aligned}
a_n &= t_{n-1}.1 + \Sigma_{i=2}^{n-1} t_{n-i}(b_i + c_i + d_i) \\
b_n &= t_{n-1}.1 + \Sigma_{i=2}^{n-1} t_{n-i}(a_i + c_i + d_i) \\
c_n &= \Sigma_{\{i,j,k,l,m \geq 1 | i+j+k+l+m=n\}} t_i t_j t_k t_l t_m \\
d_n &= \Sigma_{\{i,j,k,l,m \geq 1 | i+j+k+l+m=n\}} t_i t_j t_k t_l t_m
\end{aligned}$$

So $c_n = d_n$. Also note that since a node labeled 41352/25314 ought to have five children, $c_{n,} d_n = 0$ for $n < 5$. Summing up a_n and b_n and using the identity $t_i = a_i + b_i + c_i + d_i$ we get

$$\begin{aligned}
a_n + b_n &= t_{n-1} + t_{n-1}t_1 + \Sigma_{i=2}^{n-1} t_{n-i}(a_i + b_i + c_i + d_i + c_i + d_i) \\
&= t_{n-1} + \Sigma_{i=1}^{n-1} t_{n-i} t_i + 2\Sigma_{i=2}^{n-1} t_{n-i} c_i
\end{aligned}$$

If we substitute for c_i in $\Sigma_{g=1}^{n-1} t_{n-g} c_g$, we will get

$$\Sigma_{\{h,i,j,k,l,m \geq 1 | h+i+j+k+l+m=n\}} t_h t_i t_j t_k t_l t_m$$

because t_{n-g} runs from 1 to $n-1$ and i, j, k, l, m in the expansion of c_i sums up to g, hence if we let $h = n - g$ then we get $h + i + j + k + l + m = n$. Thus we get the following recurrence for t_n

$$\begin{aligned}
t_n = \quad & t_{n-1} + \Sigma_{i=1}^{n-1} t_{n-i} t_i + \\
& 2\Sigma_{\{h,i,j,k,l,m \geq 1 | h+i+j+k+l+m=n\}} t_h t_i t_j t_k t_l t_m + \\
& 2\Sigma_{\{i,j,k,l,m \geq 1 | i+j+k+l+m=n\}} t_i t_j t_k t_l t_m
\end{aligned}$$

We were not able to solve the recurrence using the ordinary generating function $T(z)$ associated with the sequence t_n defined below.

$$T(z) = \Sigma_{n=1}^{\infty} t_n z^{n-1}$$

We multiplied the recurrence with $\Sigma_{n=1}^{\infty} z^{n-1}$, to get,

$$T(z) = zT(z) + zT^2(z) + z^4T^5(z) + z^5T^6(z) + t_1$$

Substituting $t_1 = 1$, we get the following polynomial equation in $T(z)$

$$z^5T^6(z) + z^4T^5(z) + zT^2(z) + (z-1)T(z) + 1 = 0$$

Unfortunately this is a polynomial of sixth degree. Hence no general solution is available for its roots, which are needed to obtain the closed form expression for the above recurrence relation.

Note that in a similar way recurrence relation for any HFO$_k$ can be constructed by counting the skewed generating trees of order k where the roots can be any Uniquely HFO$_l$ permutation for $l \leq k$. With our characterization of HFO$_l$ and the algorithm for checking whether a permutation is HFO$_l$, we can easily find out the number of Uniquely HFO$_l$ permutations for any l and easily get the recurrence for any HFO$_k$ using the above mentioned method.

4.3 Poly-time Algorithm for counting HFO$_5$ permutations

Note that the recurrence obtained above can be used to construct a polynomial time algorithm for finding t_n thus the number of distinct HFO$_5$ floorplans with n rooms. We are going to use dynamic programming to compute the value of t_n using the above recurrence relation. The algorithm is fairly straight forward.

The table T is used to store the values of $t_i, 1 \leq i \leq n$. The **for loop** of lines 3-29, computes successive values of t_i using the recurrence relation we obtained earlier. The algorithm runs in time $O(n^6)$. In general the algorithm for HFO$_k$ based on a recurrence obtained using the above method will run in time $O(n^{k+1})$.

```
T ←new Array(n);
T [1] ← 1;
for m = 2 to n do
    x ←0,y ←0,z ←0;
    for i = 1 to m −1 do
        x ←x +T [i][m − i];
    end
    for i = 1 to m − 4 do
        for j = 1 to min(m-i,m −4) do
            for k = 1 to min(m-(i + j),m −4) do
                for l = 1 to min(m-(i + j + k),m-4) do
                    y ← y + T[i] ∗ T[j] ∗ T[k] ∗ T[l] ∗ T[m − (i + j + k + l)];
                end
            end
        end
    end
    for h = 1 to m − 5 do
        for i = 1 to min(m − h,m − 5) do
            for j = 1 to min(m − (h + i),m − 5) do
                for k = 1 to min(m − (h + i + j),m − 5) do
                    for l = 1 to min(m − (h + i + j + k,m − 5) do
                        z ← z+T[h]∗T[i]∗T[j]∗T[k]∗T[l]∗T[m−(h+i+j+k+l)];
                    end
                end
            end
        end
    end
    T [m] ← x + 2y + 2z + T [m − 1];
end
Output T [n];
```

Algorithm 4: Algorithm for producing the count of number of distinct HFO_5 floorplans of n rooms

CHAPTER 5

Properties of Baxter Permutations

Baxter permutations are an interesting family of permutations combinatorially. They were first introduced to solve a conjecture about fixed points of commutative functions by G. BaxterBaxter (1964). They are interesting from the VLSI perspective because of their bijective correspondence to mosaic floorplans. In this chapter we explore some properties of Baxter permutation which can be easily associated with properties of corresponding mosaic floorplans. The first such property is that Baxter permutations are closed under inverse. We give a direct proof for this by the method of contradiction. Then we prove that the mosaic floorplan corresponding to the inverse is obtained by taking a mirror image of the floorplan corresponding to the permutation about the horizontal axis. Another such result is that reverse of a Baxter permutation is also a Baxter permutation. This is a straight forward observation from the definition of Baxter permutations itself. But this result becomes interesting when the connection to geometry is made. The geometric operation on a mosaic floorplan corresponding to reverse on a Baxter permutation, is rotating the mosaic floorplan first by $90°$s clockwise and then taking a mirror image along the horizontal axis.

5.1 Closure Under Inverse

Theorem 9. *If a permutation $\pi \in S_n$ is Baxter then so is π^{-1}.*

We prove this by giving a direct prove using the method of contradiction.

Proof. Suppose it is not, then there is a text matching $3142/2413$ with absolute difference of first and last being exactly one. Let this text be at locations i, j, k, l with $i < j < k < l$. Suppose $\pi^{-1}[i], \pi^{-1}[j], \pi^{-1}[k], \pi^{-1}[l]$ forms the pattern 2413, the we know that $\pi^{-1}[i]+1 = \pi^{-1}[l]$ and $\pi^{-1}[k] < \pi^{-1}[i] < \pi^{-1}[l] < \pi^{-1}[j]$. Hence $\{i, j, k, l\}$ appears in the order (k, i, l, j) in π with i, l appearing in consecutive locations so they form the pattern 3142. If $k = j+1$ then this violates the assumption that π is Baxter. If $k > j+1$ then $j+1$ has to appear before i or after l in π as i and l appear in consecutive positions. If $j+1$ appears before i in π then $j+1, i, l, j$ forms the pattern 3142 with absolute difference of first and last being one thus violating the assumption that π is Baxter. So the only place $j+1$ could be is after l, now consider $k, i, l, j+1$, this forms the pattern 3142 and $|k-(j+1)| < |k-j|$, so if still $|k-(j+1)| > 1$ then we could apply the same argument as above and include $j+2$. This process cannot go on for ever as each time $|k-(j+i)|$ is decreasing in value. So after $|k-j|-1$ steps we get a text matching the pattern 3142 with absolute difference of first and last being one thus contradicting the assumption that π is Baxter. Since we have exhausted all the cases and arrived at a contradiction in each one our assumption that π^{-1} contained a text matching 2413 with absolute difference of first and last being one is wrong. Similarly it can be proved that π^{-1} does not contain any text matching 3142 with absolute difference of first and last being one. Hence the theorem. □

We know prove that equivalent operation on a mosaic floorplan corresponding to the inverse, is taking the mirror image about vertical axis.

Theorem 10. *Let f_π denote the mosaic floorplan corresponding to a Baxter permutation. For any given Baxter permutation π, the floorplan corresponding to inverse, $f_{\pi^{-1}}$ can be obtained from f_π by taking a mirror image about the horizontal axis.*

Proof. Let π be a Baxter permutation of length n. Let us take two indices i and j such that $i < j$. Consider $\pi[i]$ and $\pi[j]$, either $\pi[i] < \pi[j]$ or $\pi[i] > \pi[j]$.

Case I:$\pi[i] < \pi[j]$

Since $\pi[i] < \pi[j]$ and $\pi[i]$ appears before $\pi[j]$ by Lemma 1, $\pi[i]$ is to the left of $\pi[j]$ in the mosaic floorplan corresponding to π, denoted by f_π. In the inverse of π, π^{-1} indices $\pi[i]$ and $\pi[j]$ will be mapped to i and j respectively. Hence in $f_{\pi^{-1}}$, the basic rectangles labeled by i and j will be such that i precedes j in the top-left deletion ordering(as $i < j$) and also in bottom left deletion ordering(as $\pi[i] < \pi[j]$). Hence i is to the left of j in $f_{\pi^{-1}}$.

Case II:$\pi[i] > \pi[j]$

Since $\pi[i] > \pi[j]$ and $\pi[i]$ appears before $\pi[j]$ by Lemma 2, $\pi[i]$ is below $\pi[j]$ in

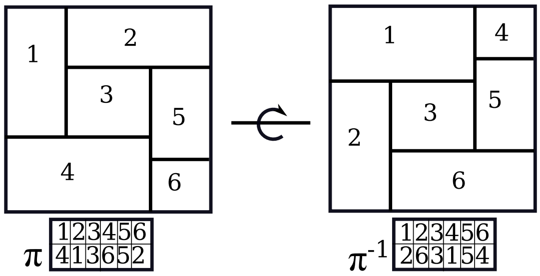


Figure 5.1: Obtaining a mosaic floorplan corresponding to the inverse of a Baxter permutation

f_π. In the inverse of π, π^{-1} indices $\pi[i]$ and $\pi[j]$ will be mapped to i and j respectively. Hence in $f_{\pi^{-1}}$, the basic rectangles labeled by i and j will be such that i precedes j in the top-left deletion ordering(as $i < j$) but in bottom left deletion ordering j precedes i(as $\pi[i] < \pi[j]$). Hence i is above j in $f_{\pi^{-1}}$.

Hence we get a mapping between the basic rectangles of f_π and basic rectangles of $f_{\pi^{-1}}$, such that whenever $\pi[i]$ is below $\pi[j]$, their images i, j will be such that i will be above j. And whenever $\pi[i]$ is to the left of $\pi[j]$ so is i and j. The geometrical operation which flips the above/below relation but does not affect the left/right relation is flipping the object about the horizontal axis. Hence the theorem. Figure 5.1 illustrates the above mentioned link between inverse and the geometry. □

5.2 Closure under reverse

Theorem 11. *If π is a Baxter permutation then so is its reverse, more over the mosaic floorplan corresponding to reverse of π can be obtained from mosaic floorplan corresponding to π by applying clockwise rotation by $90°$s and then applying reflection about the vertical axis.*

Proof. By definition Baxter permutations itself it is clear that the reverse of a Baxter permutation is also a Baxter permutation. But let us find out what is the equivalent operation on the mosaic floorplan corresponding to the Baxter permutation which produces the mosaic floorplan corresponding to the reverse of the given Baxter permutation. Let π be a Baxter permutation and as above let f_π represent the mosaic floorplan corresponding to π. Let us take two indices i and j in π such that $i < j$. If $\pi[i] < \pi[j]$ in π we know from the above analysis that $\pi[i]$ is to the left of $\pi[j]$ in f_π. Let π^r represent the

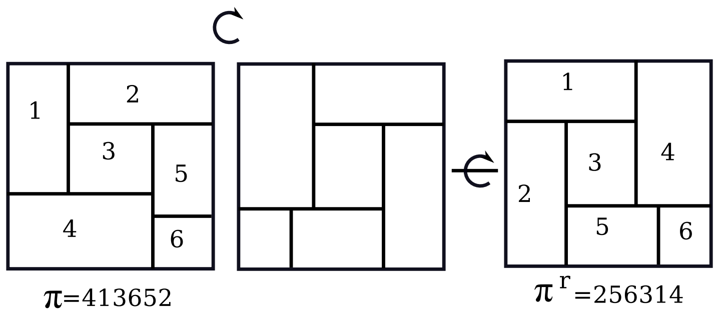


Figure 5.2: Obtaining a mosaic floorplan corresponding to the reverse of a Baxter permutation

reverse of the permutation π and let f_{π^r} represent the mosaic floorplan corresponding to the reverse of π, π^r. In π^r the order of numbers $\pi[i]$ and $\pi[j]$ will be reversed, and they will be at locations $k = n-i+1$ and $l = n-j+1$ respectively. That is $l < k$ and $\pi^r[l] > \pi^r[k]$ $\pi[j] > \pi[i]$ we get that in f_{π^r} $\pi[j]$ is below $\pi[i]$. If $\pi[i] > \pi[j]$ in π then we know from the proof of the earlier theorem that $\pi[i]$ is below $\pi[j]$. In the reverse π^r the order of numbers $\pi[i]$ and $\pi[j]$ will be reversed. Hence they will be at locations $k = n - i + 1$ and $l = n - j + 1$ respectively. That is $l < k$ and $\pi^r[l] < \pi^r[k]$ as $\pi[j] < \pi[i]$. This implies that $\pi[j]$ is to the left of $\pi[i]$ in f_{π^r}. Summarizing this, if the room labeled $\pi[i]$ is to the left of $\pi[j]$ in f_π then the operation which obtains the floorplan corresponding to the reverse of π will change the relative ordering of these blocks such that $\pi[i]$ will be below $\pi[j]$. And similarly if the room labeled $\pi[i]$ is below $\pi[j]$ in f_π then the operation corresponding to the reverse will change their relative ordering such that $\pi[i]$ is to the left of $\pi[j]$. This corresponds to the rotation by $90°$ clock-wise and then taking mirror image along the horizontal axis. Figure 5.2 illustrates the above mentioned link between reverse and the geometry.

$\square$

CHAPTER 6

Discussions and Open Questions

6.1 Summary of Results

We characterized permutations corresponding to the Abe-label of HFO_k floorplans. We also proved that HFO_k floorplans are in bijective correspondence with skewed generating trees of Order k. This gave us a recurrence relation for the exact number of HFO_k floorplans with n rooms and thus a polynomial time algorithm for generating the count for any given n. We obtained a linear time algorithm for checking if a given permutation is HFO_k for a particular value of k. The same algorithm can be used to check whether a permutation is HFO_k for some unknown k in $O(n^2 \log n)$ time. We extended the neighbourhood moves on HFO_k floorplans for stochastic search methods like simulated annealing on these family of floorplans. We also proved that Baxter permutations are closed under inverse and reverse.

6.2 Open Questions

Even though we were able to obtain a recurrence relation for the exact number of HFO_k floorplans with n rooms and thus a polynomial time algorithm for generating the count for any given n, we were not able to find a closed form expression for the number of distinct HFO_k floorplans with n rooms. Even for a particular value of k (especially 5) it would be interesting to see a closed form expression for the number of distinct HFO_k floorplans. Another open question arising from our research is the number of distinct Uniquely HFO_k floorplans. We were able to obtain some trivial lower bounds based on the construction method described in the proof of Infinite hierarchy. But no closed form expression for the number of Uniquely HFO_k floorplans were obtained.

REFERENCES

1. **Ackerman, E.**, **G. Barequet**, and **R. Y. Pinter** (2006). A bijection between permutations and floorplans, and its applications. *Discrete Applied Mathematics*, **154**(12), 1674 – 1684. ISSN 0166-218X. URL http://www.sciencedirect.com/science/article/pii/S0166218X0600117X.

2. **Albert, M.** and **M. Atkinson** (2005). Simple permutations and pattern restricted permutations. *Discrete Mathematics*, **300**(1-3), 1 – 15. ISSN 0012-365X. URL http://www.sciencedirect.com/science/article/pii/S0012365X05003377.

3. **Baxter, G.**, On fixed points of the composite of commuting functions. *In Proc. Amer. Math. Soc*, volume 15. 1964.

4. **Chung, F. R. K.**, **R. L. Graham**, **V. E. Hoggatt**, and **M. Kleiman** (1978). The number of baxter permutations. *Journal of Combinatorial Theory, Series A*, **24**(3), 382 – 394. ISSN 0097-3165. URL http://www.sciencedirect.com/science/article/pii/0097316578900687.

5. **Dulucq, S.** and **O. Guibert** (1998). Baxter permutations. *Discrete Mathematics*, **180**(1-3), 143–156.

6. **Hong, X.**, **G. Huang**, **Y. Cai**, **J. Gu**, **S. Dong**, **C.-K. Cheng**, and **J. Gu** (2000). Corner block list: An effective and efficient topological representation of non-slicing floorplan. *International Conference on Computer-Aided Design*, 8. ISSN 1092-3152.

7. **Murata, H.**, **K. Fujiyoshi**, **S. Nakatake**, and **Y. Kajitani**, Rectangle-packing-based module placement. *In Proceedings of the 1995 IEEE/ACM international conference on Computer-aided design*, ICCAD '95. IEEE Computer Society, Washington, DC, USA, 1995. ISBN 0-8186-7213-7. URL http://portal.acm.org/citation.cfm?id=224841.225094.

8. **Murata, H.**, **K. Fujiyoshi**, **S. Nakatake**, and **Y. Kajitani** (1996). Vlsi module placement based on rectangle-packing by the sequence-pair. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, **15**(12), 1518 –1524. ISSN 0278-0070.

9. **Murata, H.**, **K. Fujiyoshi**, **T. Watanabe**, and **Y. Kajitani**, A mapping from sequence-pair to rectangular dissection. *In Design Automation Conference 1997. Proceedings of the ASP-DAC '97*. 1997.

10. **Pan, P.** and **C. Liu** (1995). Area minimization for floorplans. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, **14**(1), 123 –132. ISSN 0278-0070.

11. **Sait, S. M.** and **H. Youssef**, *VLSI physical design automation: theory and practice*. World Scientific, 1999. ISBN 9789810238834.

12. **Sakanushi, K.**, **Y. Kajitani**, and **D. Mehta** (2003). The quarter-state-sequence floorplan representation. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, **50**(3), 376 – 386. ISSN 1057-7122.

13. **Schmerl, J. H.** and **W. T. Trotter** (1993). Critically indecomposable partially ordered sets, graphs, tournaments and other binary relational structures. *Discrete Math.*, **113**, 191–205. ISSN 0012-365X. URL http://dx.doi.org/10.1016/0012-365X(93)90516-V.

14. **Shen, Z.** and **C. Chu** (2003). Bounds on the number of slicing, mosaic, and general floorplans. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **22**(10).

15. **Wang, T.-C.** and **M. D. F. Wong** (1992). Optimal floorplan area optimization. *IEEE Trans. on CAD of Integrated Circuits and Systems*, **11**(8), 992–1002.

16. **Wong, D.** and **C. Liu**, A new algorithm for floorplan design. *In 23rd Conference on Design Automation, 1986*. 1986.

17. **Wong, D.** and **K.-S. The**, An algorithm for hierarchical floorplan design. *In Computer-Aided Design, 1989. ICCAD-89. Digest of Technical Papers., 1989 IEEE International Conference on*. 1989.

18. **Wong, D. F.** and **P. S. Sakhamuri**, Efficient floorplan area optimization. *In DAC*. 1989.

19. **Yao, B.**, **H. Chen**, **C.-K. Cheng**, and **R. Graham** (2003). Floorplan representations: Complexity and connections. *ACM Trans. Des. Autom. Electron. Syst.*, **8**(1), 55–80. ISSN 1084-4309.

20. **Young, E.**, **C. C. N. Chu**, and **Z. C. Shen** (2002). Twin binary sequences: A non-redundant representation for general non-slicing floorplan. *IEEE TCAD*, **22**, 196–201.

21. **Zhou, S.**, **S. Dong**, **C.-K. Cheng**, and **J. Gu**, Ecbl: an extended corner block list with solution space including optimum placement. *In Proceedings of the 2001 international symposium on Physical design*, ISPD '01. ACM, New York, NY, USA, 2001. ISBN 1-58113-347-2. URL http://doi.acm.org/10.1145/369691.369756.

LIST OF PAPERS BASED ON THESIS

1. Shankar Balachandran, Sajin Koroth - A study on the number of Hierarchical Rectangular Partitions of Order k *Journal* of Integer Sequences, **Under review.**