

Fixed Layer Embeddings of Binary Trees

Wolfgang W. Bein * Lawrence L. Larmore † Charles Shields, Jr. ‡
I. Hal Sudborough §

Abstract

We describe total congestion 1 embeddings of complete binary trees into three dimensional grids with a fixed number of layers. More specifically, we give a one-to-one embedding of any complete binary tree into a hexahedron shaped grid such that no tree nodes or edges occupy the same grid positions.

With 7 layers, the number of nodes in the grid is at most 1.09375 times the number of nodes in the tree and with 5 layers we obtain a ratio of $\frac{75}{64} = 1.171875$. Unlike more standard embeddings these embeddings intricately weave the branches of various subtrees into each other. Finally using a standard recursive method, for 2 layers a ratio of $\frac{39}{32} = 1.21875$ can be obtained.

1 Introduction

Recent research [2] has focused on embedding binary trees into three dimensional (3D) grids. Work on such 3D embeddings has been sparse, and other than our previous work, so far as we know, such embeddings into 3D grids have not been the subject of any previous paper, although there are papers describing embeddings of complete binary trees into hypercubes [8] and butterfly networks [5]. We note that there are a number of more general papers on three-dimensional VLSI layouts (*e.g.* [1, 9].) These papers are relevant to our work without giving specific results for the case of embedding binary trees into three-dimensional grids.

*Department of Computer Science, University of Nevada, Las Vegas, Nevada 89154. Email: bein@cs.unlv.edu. Research supported by NSF grant CCR-9821009.

†Department of Computer Science, University of Nevada, Las Vegas, Nevada 89154. Email: larmore@cs.unlv.edu. Research supported by NSF grant CCR-9821009.

‡Department of Computer Science, University of Texas at Dallas, Richardson, Texas 75083. Email: chasjr@utdallas.edu. Research supported by a grant from the Girou Foundation.

§Department of Computer Science, University of Texas at Dallas, Richardson, Texas 75083. Email: hal@utdallas.edu.

For our problem then, there is a general technique based on separators in graphs that yields a two layer layout with a very large constant expansion, see [13]. We mention that unlike for the three-dimensional case, there exists a rich literature on 2D tree-to-grid embeddings. (See [3, 4, 6, 7, 10, 12].) In [2] we have given embedding schemes where the number of layers can be quite large. For practical applications one is often more interested in 3D embeddings where the number of layers is a (small) fixed number. For the sake of VLSI, or other graph drawing applications, embeddings with a small number of layers are often better. Here we describe such embeddings for 2, 5, and 7 layers.

In an embedding, one has a *guest* graph $G = (V, E)$ that represents the parallel architecture to be simulated, the computation graph to be mapped to the processors, or the circuit to be laid out. One also has a *host* graph $H = (V', E')$ that represents the parallel computer architecture on which the computation is to be performed, or the positions for gates and routing patterns on a VLSI chip or wafer, see [8, 11].

The approach described in [2] is somewhat similar to the usual techniques for embeddings into 2D grids. That is, it is an iterative procedure that obtains embeddings of the binary tree from previous embeddings of trees of smaller height. As the iteration progresses the trees are embedded into rectangularly shaped grid graphs. This makes the iteration conceptually straightforward, as during the iteration smaller brick-shaped objects are pieced together to form larger brick-shaped objects. The ease inherent to this approach comes at a price: The embeddings are not as tight as they could possibly be if one were able to intricately weave the branches of various subtrees into each other. In this paper we describe embeddings into fixed-layer three dimensional grids, using this more intricate approach, as well as a new improved version of a standard technique.

Let $K_{a,b,k}$ denote the grid graph with a rows, b columns, and k layers. We think of a and b as “big”,

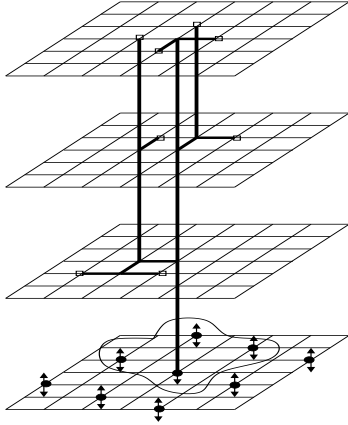


Figure 1: A Tree

and k as fixed and relatively small; in this paper we will consider $k = 2, 5,$ and 7 . Let T_h denote the complete binary tree of height h with $2^{h+1} - 1$ vertices. We consider one-to-one, congestion 1 embeddings f of T_h into $K_{a,b,k}$ for various values of h, a, b . Such an embedding is an injection assigning, to each vertex v of a tree $T = T_h$, a single vertex $f(v)$ in a grid $K = K_{a,b,k}$, and assigning to each edge uv of T a path $f(uv)$ in K between $f(u)$ and $f(v)$, such that the internal nodes in $f(uv)$ include neither $f(z)$, for any vertex z in T , nor any point in $f(st)$ for any other edge st in T . In other words, such an embedding defines a subgraph of K which is homeomorphic to T . An embedding is commonly called a *layout*, and we shall use the two terms “embedding” and “layout” interchangeably. For an embedding f of T_h into $K_{a,b,k}$ we define the *expansion ratio* r of f , to be the number of points in $K_{a,b,k}$ divided by the number of vertices in T_h , namely

$$r = \frac{abk}{2^{h+1} - 1}$$

More generally, let f be an embedding from a guest graph G to a host graph H . The *dilation* of an edge uv under f is the length of the path $f(uv)$. The *dilation* of the embedding f is the maximum dilation of any edge of G under f . The *load* of a vertex x in H under f , denoted by $load(x)$, is the number of vertices mapped by f to x . The *congestion* of a vertex x in H under f , denoted by $congestion(x)$, is the number of paths of the form $f(uv)$, for an edge uv in G , containing x as an internal vertex. The *total congestion* of a vertex x is $load(x) + congestion(x)$. In [2] we gave details for 3D embeddings with a non-fixed number of layers. That is, we gave a total congestion 1 embedding of

complete binary trees into three dimensional grids with expansion ratios of 1.25. As the height of the tree increased, the number of layers increased as well.

In this paper we describe in the next section an embedding scheme with expansion ratio $r = \frac{35}{32} = 1.09375$ in the 7-layer grid. This embedding is very tight, and it fills up all but the middle layer perfectly. In section 3 we describe another such construction for 5 layers, again, all layers except for the middle layer are used perfectly. Its expansion ratio is $\frac{75}{64} = 1.171875$. Finally, in section 4 we present a 2-layer embedding with ratio $\frac{39}{32} = 1.21875$.

2 The Texas Embedding

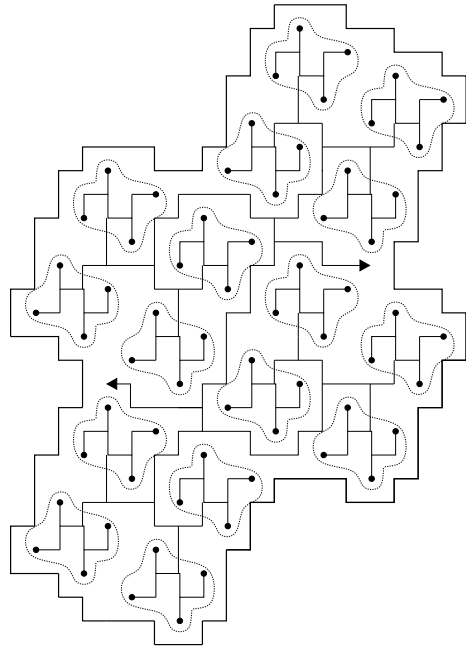


Figure 2: A Texas Tile

We consider a 7-layer grid, with layers numbered $1, \dots, 7$, top to bottom. Each layer has grid points $\{(i, j) | i, j \in N\}$. We identify base points $\mathcal{B} = \{(i, j) | 2i - j \equiv 0 \pmod{5}\}$, which are spread out over the middle layer 4 in a “knight’s-move” pattern; see Figure 1.

At the heart of the embedding, which we call the “Texas Embedding,” each base point is used as the root of a T_4 . T_4 has two subtrees T_3 ; one of those subtrees of is embedded in layers 1,2,3, as shown in Figure 1;

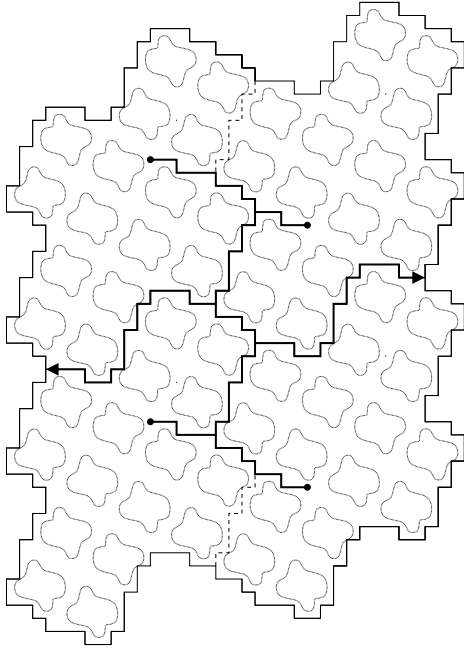


Figure 3: Four Texas Tiles: A Block

the other T_3 is embedded in an identical manner in layers 5,6,7.

It is not immediately clear that all the T_3 's fit into the layers without running into each other. It turns out that there is no overlap, and, in fact, that all points are used. To see this, we define the points used in more precise terms: Let (k, l) be a base point, then the points in layer 3 used are $\mathcal{B}_{k,l}^3 = \{(k, l), (k-1, l), (k-1, l-1), (k-2, l-1), (k, l-1)\}$. For layer 2, the set is $\mathcal{B}_{k,l}^2 = \{(k, l), (k-1, l), (k-1, l+1), (k, l+1), (k+1, l+1)\}$ and for layer 1 the set is $\mathcal{B}_{k,l}^1 = \{(k, l), (k, l-1), (k-1, l), (k+1, l), (k, l+1)\}$. It is a routine exercise to show that for any $i = 1, 2, 3$, the family of sets $\{\mathcal{B}_{k,l}^i | (k, l) \in \mathcal{B}\}$ form a partition of the grid \mathbf{N}^2 .

We continue our description of the Texas Embedding by focusing on Figure 2. This figure conveys best the choice of the name "Texas Embedding": Four base points at a time are connected to form a T_6 . Such a piece is called a "Texas piece." Figure 2 shows how 16 Texas pieces can be placed to form a T_{10} . The cyclic path around those 16 pieces is used for further routing, and it is connected to the T_{10} inside its perimeter by one of the two escape-routes indicated by arrows. Altogether, the object described in Figure 2 with one of the two escape routes (right or left) is called a "Texas tile." In Figure 3 one takes four tiles and connects these into

a regular larger pattern called a "Texas block." These Texas blocks can now be used to continue constructing larger trees indefinitely using the usual H-construction. (This standard technique is described in [13].)

To obtain the expansion ratio we can carefully subdivide the 7-layer grid into rectangular blocks of size 16 by 20 by 7, so that each block approximates the position of one of the Texas tiles. There are 2240 points in each block and 2048 nodes in T_{10} (including the one escape point,) yielding a ratio of $r = \frac{2240}{2048} = \frac{35}{32} = 1.09375$.

3 The Twin Tree Embedding

Consider a 5-layer grid, with layers numbered 1, ..., 5, top to bottom. We describe an embedding of T_h in this grid, such that the expansion ratio converges to $\frac{75}{64} \approx 1.172$. We call this a "Twin Tree Embedding."

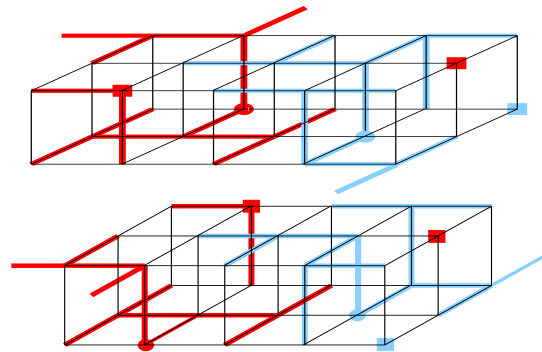


Figure 4: A Twin Block (top); A Twin Mirror Block (bottom)

The overall technique is similar to that of the Texas Embedding given in Section 2. Layer 3 contains a pattern of *base points* (as shown in Figure 5), each one of which will be the root of a copy of T_4 in the final construction. The "outer" layers, namely layers 1,2,4,5, are exactly filled with copies of T_3 . Each copy of T_3 uses just 15 nodes of the outer layers, and, in the limit, no outer layer node is unused. Each base point in Layer 3 is connected to the roots of two copies of T_3 , one root in Layer 2 and another in Layer 4.

We now concentrate on how the copies of T_3 are embedded in Layers 1 and 2. (Layers 4 and 5 use the same pattern.) Call the rectangles bounded by dotted lines (which are between grid lines) in Figure 5 *domains*. Each domain contains a 5×3 rectangular grid, which contains two base points, one "red" (dark) and one

“blue” (light). Figure 4 shows two $3 \times 5 \times 2$ blocks, the bottom one a mirror image of the top one. Each block contains one red (dark) root and one blue (light) root, shown as large round dots, both on the bottom level. Each block contains red and blue edges, as shown. One of those two blocks will be placed atop each domain, so that each root in the block will be placed above a base point of the same color in Layer 3.

Figure 8 shows how blocks over adjacent domains fit together, so that each root in Level 2 is the root of one copy of T_3 of the same color. We note that each red tree extends over three blocks, while each blue tree extends over two blocks; yet, there is an “average” of one tree of each color in each block.

Finally, the base points in Layer 3 are connected to each other using the H-construction, as shown in Figure 5. Lemma 1 summarizes the result.

Lemma 1 T_{5+2t} embeds in $M[5(2^t) + 1, 3(2^t) + 2, 5]$ and T_{6+2t} embeds in $M[5(2^t) + 1, 6(2^t) + 2, 5]$, for all $t \geq 0$.

The limiting expansion ratio is thus $r = \frac{75}{64} = 1.171875$.

4 Embeddings in Two Layers

In this section, we use a recursive method to obtain an embedding of T_h in a rectangular 2-layer grid with expansion ratio $\frac{39}{32} = 1.21875$, for all $h \geq 7$.

For each $t \geq 0$, we give four embeddings of T_{7+4t} into the rectangular grid $M[12(2^t), 13(2^t), 2]$. We call these embeddings Type A, Type B, Type C, and Type D. Each embedding has *reserved points*, which are points which are not used for the embedding of T_h , but are left free to make higher level connections. We summarize the rules, which we call the *reserve conditions*, for each type of embedding below.

- The lower row and the top half of the left column of level 2 of each embedding of Type A are reserved.
- The right column and the left half of the top row of level 2 of each embedding of Type B are reserved.
- The left column and the lower half of the right column of level 2 of each embedding of Type C are reserved.
- The lower row and the right half of the top row of level 2 of each embedding of Type D are reserved.

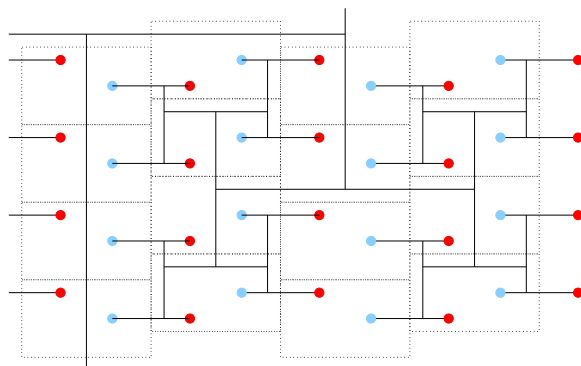


Figure 5: H-Construction, Middle Layer

- The middle one or two grid points of the top row of level 2 of each embedding are reserved for the *escape path* from the root of the tree, which is always located approximately in the middle of the grid.

The base step of the recursion consists of the four embeddings of T_7 into $M[12, 13, 2]$ shown in Figures 6 and 7. For $t > 0$, each type of embedding of T_{7+4t} into $M[12(2^t), 13(2^t), 2]$ is obtained by combining 16 “small” embeddings, *i.e.*, embeddings of T_{3+4t} , to form the “large” embedding of T_{7+4t} as shown in Figure 9. (Note that, for $t > 0$, the Type A and the Type D embeddings are identical, as are the Type B and Type C embeddings.) The reserved points of the 16 small embeddings are used to make the connections (which are all in level 2) and also to provide the reserved points of the large embedding. The connections shown in those figures are all in level 2. If $h > 7$ is not of the form $7 + 4t$, then the final step of process will use only 2, 4, or 8 of the embeddings from the previous step.

5 Concluding Remarks

We note that the 2-layer construction in the previous section does not follow the weaving-of-branches approach of the Texas Embedding or the Twin Tree Embedding. As much as we are convinced that for the latter embeddings the expansions ratios are rather tight, we do expect that the expansion ratio for 2-layers can be further improved. We mention that the dilation of our embeddings are (naturally) relatively high, we give precise calculations in the full length journal version. We note that there is an embedding into 6-layers with an expansion ratio equal to our 5-layer embed-

ding, see [2]. Finally we suspect that if there exists an embedding with a certain expansion ratio for k layers, then there exists an embedding with the same or better expansion ratios for $\ell > k$; however such a reduction has not been shown.

References

- [1] A. Aggarwal, M. Klawe, and P. Shor. Multilayer grid embeddings in vlsi. *Algorithmica*, 6:129–151, 1991.
- [2] W. W. Bein, L. L. Larmore, C. Shields, and I. H. Sudborough. Three-dimensional embedding of binary trees. In *Proceedings of the 5th International Symposium on Parallel Architectures*, pages 140 – 147. IEEE Computer Society, 2000.
- [3] S. N. Bhatt and S. S. Cosmadakis. The Complexity of minimizing wire lengths in VLSI layouts. *Information Processing Letters*, 25:263–267, 1987.
- [4] A. Gregori. Unit-length embedding of binary trees in a square grid. *Information Processing Letters*, 31:167–173, 1989.
- [5] Ajay K. Gupta and Susanne E. Hambrusch. Embedding complete binary trees into butterfly networks. *Transactions on Computers*, 40(7):853–863, 1991.
- [6] R. Heckmann, R. Klasing, B. Monien, and W. Unger. Optimal embedding of complete binary trees into lines and grids. In *Proceedings of the 17th Graph Theoretic Concepts in Computer Science*, volume 570 of *Lecture Notes in Computer Science*. Springer, 1991.
- [7] M.-C. Heydemann, J. Opatrny, and D. Sotteau. Embedding complete binary trees into extended grids with edge congestion 1. In *Parallel Algorithms and Applications*, volume 805 of *Lecture Notes in Computer Science*. Springer, 1994.
- [8] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, and Hypercubes*. Morgan Kaufmann, San Mateo, CA, 1992.
- [9] F.T. Leighton and A.L. Rosenberg. Three-dimensional circuit layouts. *SIAM Journal on Computing*, 15:793–813, 1986.
- [10] Z. Miller, M. Perkel, D. Pritikin, and I. H. Sudborough. Expansions of layouts of complete binary trees into grids I - upper bounds. Submitted, 2000.
- [11] B. Monien and I. H. Sudborough. Embedding one interconnection network in another. *Computing Supplement*, 7:257–282, 1990.
- [12] M. S. Patterson, W. Ruzzo, and L. Snyder. Bounds on minimax edge length for complete binary trees. In *Proc. 23rd Symp. Theory of Computing*, 1991.
- [13] J. D. Ullman. *Computational Aspects of VLSI*. Computer Science Press, Rockville, Maryland, 1984.

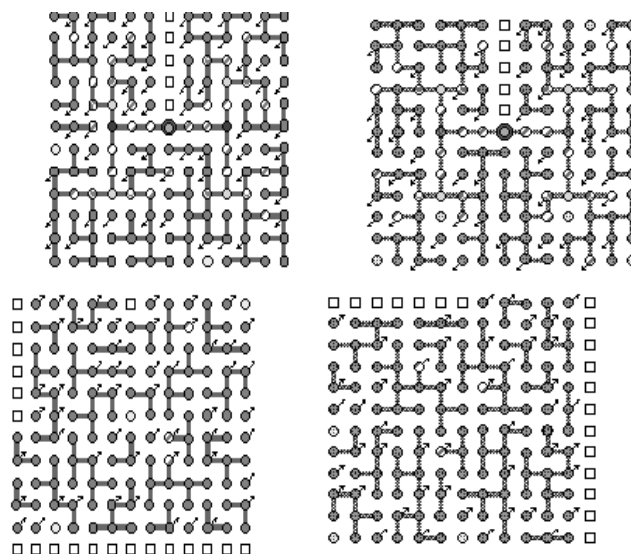


Figure 6: Layout A (left); Layout B (right)

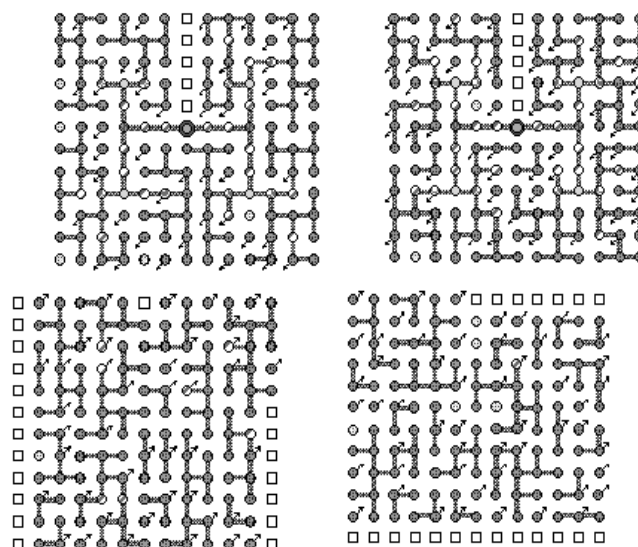


Figure 7: Layout C (left); Layout D (right)

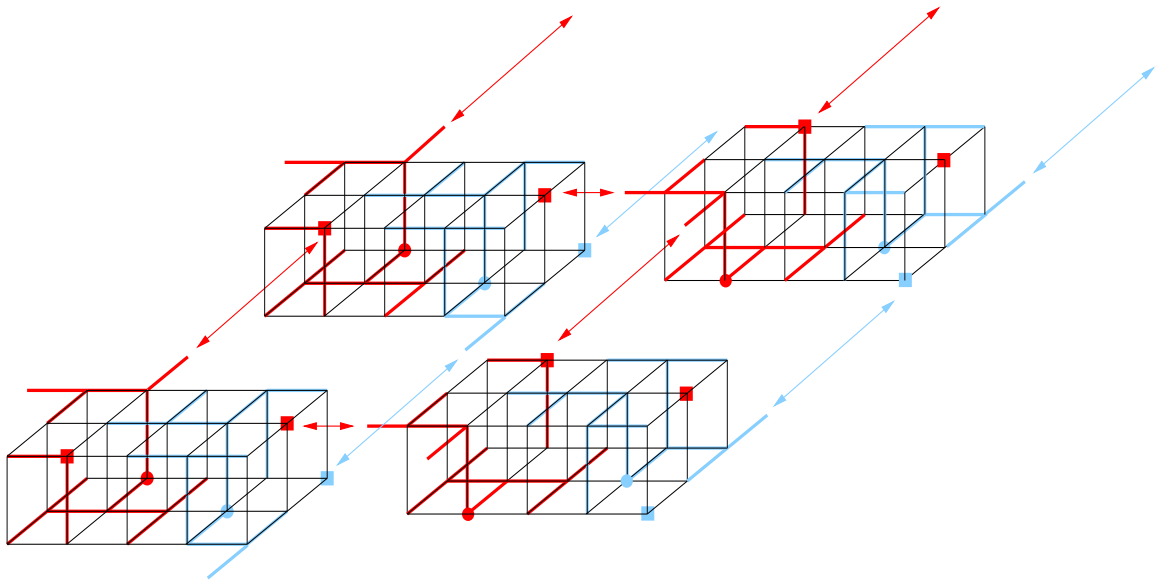


Figure 8: Topmost Layers

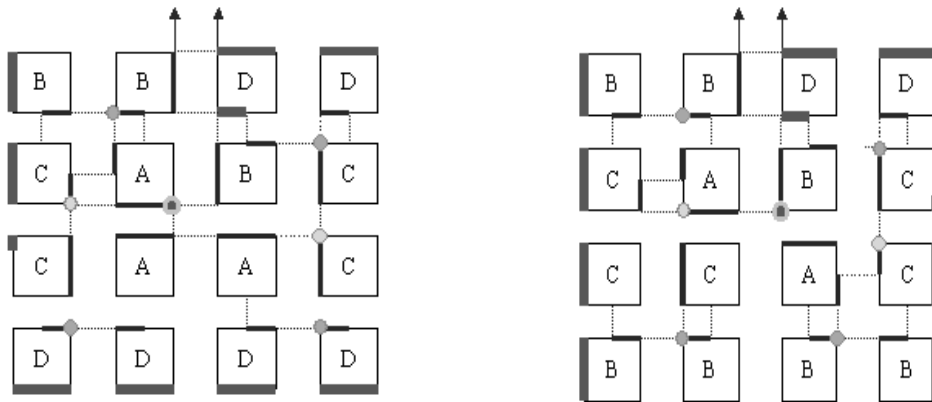


Figure 9: Procedural Diagram for A and D (left); Procedural Diagram for B and C (right)