Layered Cross Product — A Technique to Construct Interconnection Networks

Shimon Even* and Ami Litman†
Computer Science Department
Technion, Israel Institute of Technology
Haifa, Israel 32000

Abstract

We introduce a *Layered Cross Product*, LCP, of layered graphs and show that several well known networks are LCP-s of simple layered graphs, such as trees. Some important properties of these networks are shown to be trivial consequences, once a network is presented as an LCP of simpler graphs.

We believe that this new tool will make the construction of new networks easier, and it will simplify the study of the properties of known and new networks.

1 Introduction

Many layered graphs are known to be useful as interconnection networks. Examples are the butterfly network, including all its different looking representations, meshes of trees, fat-trees, the Beneš network, multibutterflies. In certain cases, networks were known under different names for decades before it was discovered that they were actually isomorphic.

We believe the reason for this "blindness" was that the networks were represented in forms which concealed their structure. To be more specific, these structures are "essentially" 3-dimensional, and the representations used to describe them were 2-dimensional.

We introduce a *Layered Cross Product*, LCP, of layered graphs and show that several of the important, and well known networks, are LCP-s of simple layered graphs, such as trees.

In particular the butterfly network is an LCP of two binary trees. The mesh of trees is an LCP of two binary trees with paths attached to their leaves. The fat tree is an LCP of a binary tree and a quad-tree. These are shown in detail. One can continue this expedition to show that:

1. The Beneš network [1], which is isomorphic to the Waksman network [2], is the LCP of two structures: The first is a binary tree which shares its leaves with its mirror image, and the second is a binary tree which shares its root with its mirror image. This is not surprising, since the Beneš network is known to be layer isomorphic to the butterfly network connected to its mirror image.

2. Consider two butterfly networks connected in tandem (without reflection). (This tandem network is known not to be layer isomorphic to the Beneš network, and contrary to it, it is not known if the tandem is rearrangeable; i.e. can realize all permutations.) This network is an LCP of "bamboos"; see Fig. 5.

3. The multibutterfly, introduced by Upfal [3], is an LCP of a binary tree and a tree-like "pyramid", in which the 2^i vertices of the i'th layer are connected

[1]Supported by the Fund for the Promotion of Research at the Technion. E-address: even@cs.technion.ac.il

[2]Part of this research was conducted while the authors were at Bellcore, Morristown, New Jersey. E-address: litman@cs.technion.ac.il

SPAA '92 - 6/92/CA
©1992 ACM 0-89791-484-8/92/0006/0060 .... $1.50

to the 2^{i+1} vertices of the $(i+1)$'st layer by an expander (see [4]).

Several of the important properties of these networks are shown to be trivial consequences, once a network is presented as an LCP of simpler graphs.

We believe that this new tool will make the construction of new networks easier, and it will simplify the study of the properties of known and new networks.

2 The Layered Cross product

A *layered graph*, of $l+1$ layers, $G = (V_0, V_1, \ldots, V_l, E)$, consists of:

1. $l+1$ *layers* of vertices; V_i is the (nonempty) set of vertices in layer i.
2. E is a set of edges. Every edge $\langle u, v \rangle$ connects two vertices of two adjacent layers; i.e. if $u \in V_i$ then $v \in V_{i+1}$.

Let $G^{(1)}, G^{(2)}$ be layered graphs, each of $l+1$ layers; i.e., for $j \in \{1,2\}$, $G^{(j)} = (V^{(j)}{}_0, V^{(j)}{}_1, \ldots, V^{(j)}{}_l, E^{(j)})$. Their *Layered Cross Product, LCP*, $G^{(1)} \times G^{(2)}$ is a layered graph, $G' = (V'_0, V'_1, \ldots, V'_l, E')$, where:

1. For every $0 \le i \le l$, $V'_i = V^{(1)}{}_i \times V^{(2)}{}_i$.
2. There is an edge $\langle u', v' \rangle$ in G', connecting vertices $u' = (u^{(1)}, u^{(2)})$ and $v' = (v^{(1)}, v^{(2)})$, if and only if $\langle u^{(1)}, v^{(1)} \rangle$ and $\langle u^{(2)}, v^{(2)} \rangle$ are edges in $G^{(1)}$ and $G^{(2)}$, respectively.

Let us call the layer with index 0 the *top layer*, and the layer with index l the *bottom layer*. We are interested in the design and description of interconnection networks; i.e. layered graphs in which paths of length l edges connect vertices of the top layer with vertices of the bottom layer. For this reason we say that a layered graph is *linked* if it has the following property: Every vertex is on a path of length l connecting a vertex of the top level with a vertex of the bottom level.

The LCP of two layered graphs is linked if and only if each of the multiplicands is linked.

In this paper we are not interested in the labels or names assigned to vertices; i.e. two layered graphs are considered *equal* if they are isomorphic, and the isomorphism preserves the level to which a vertex belongs. In many applications it is useful to assign labels to the vertices or edges in some specific manner; however, such a labeling is application specific and will not be considered in this paper.

Under this assumption, the LCP operation is commutative and associative. Thus, we may consider the LCP of more than two layered graphs, all with the same number of layers, without regard to the order in which they are written, or the order in which the binary operations are applied.

A simple path, of length l, with one vertex in each of the $l+1$ layers, serves as the identity element for the LCP operation.

Several examples are depicted in Figs. 1 and 2.

A layered graph is said to have the *banyan* property if for every vertex of the top layer and every vertex of the bottom layer there is a unique path of length l connecting them. (See [5].) It is a simple matter to prove the following theorem:

Theorem 2.1 The LCP operation yields a banyan layered graph if and only if each of its multiplicands is banyan.

Unfortunately, the rearrangeability property (i.e. the existence of disjoint paths for every assignment of vertices of the bottom layer to vertices of the top layer; see [4]) does not carry over from multiplicands to their LCP. (In Fig. 2(b), each of the multiplicands is rearrangeable, while their LCP is not; there are no two disjoint paths connecting vertices 0 and 2 of the top layer with vertices 0 and 1 of the bottom layer.)

3 The Butterfly Network

The *butterfly network*, $\mathcal{B}$, was invented and reinvented, independently and in various forms, since the Sixties. It has even more names than known forms, including the bidelta network, baseline network, flip network, modified data manipulator, indirect binary cube network, Omega network. See, for example [6]. One known definition is as follows:

Let l be a positive integer, and $\{0,1\}^l$ be the set of all binary words of length l. $\mathcal{B} = (V_0, V_1, \ldots, V_l, E)$, where

1. $V_0, V_1, \ldots, V_l$ are disjoint sets of vertices. Each set contains $n = 2^l$ vertices. For every $0 \leq i \leq l$, there is a one-to-one assignment of labels: $V_i \mapsto \{0,1\}^l$.
2. A vertex labeled $(x_1, x_2, \ldots, x_l)$, of layer $i-1$, is connected by an edge to vertex labeled $(y_1, y_2, \ldots, y_l)$, of layer i, if and only if for every $j \neq i$, $x_j = y_j$.

Theorem 3.1 The butterfly network, $\mathcal{B}$, is the LCP of two binary trees, one with its root up and one with its root down.

Proof: There are many ways to prove this theorem. Let us describe a graphical one.

Denote the LCP of the two trees by $\mathcal{P}$. We want to prove that $\mathcal{P} = \mathcal{B}$.

An *UF(i)* (upward fork) is a linked layered graph. It consists of two simple paths coming down from the top layer and merge into a single path at layer $i+1$.

Observe that a binary tree of height l, with its root down, is equal to

$$UF(0) \times UF(1) \times \cdots \times UF(l-1).$$

(See Fig. 1, (a) and (b).)

An *DF(i)* (downward fork) is a linked layered graph. It is the mirror image of *UF(l-i)*. By symmetry, the binary tree of height l, with its root up, is equal to

$$DF(1) \times DF(2) \times \cdots \times DF(l).$$

Thus,

$$\begin{aligned} \mathcal{P} &= UF(0) \times DF(1) \times UF(1) \times DF(2) \times \cdots \\ &\quad \times UF(l-1) \times DF(l) \end{aligned}$$

Define $SW(i)$ (switch) to be the following linked layer graph. It has two vertices in each of the $l+1$ layers. There are two disjoint paths, of length l, connecting the two vertices of the top layer with the two at the bottom layer. There are two more edges. Each connects a vertex which is on one path and on the $(i-1)$'st layer, with the vertex on the the other path and on the i'th layer.

Observe that $SW(i) = UF(i-1) \times DF(i)$. Thus,

$$\mathcal{P} = SW(1) \times SW(2) \times \cdots \times SW(l). \qquad (1)$$

Next, let us label the vertices in each of the $l+1$ layers of $\mathcal{P}$ with binary words of length l, according to the following scheme. In each SW(j), label one of the down going paths by 0 and the other by 1. Consider vertex $u = (u_1, u_2, \ldots, u_l)$ of the LCP yielding $\mathcal{P}$, as in Equation 1. Let the label of u be $(x_1, x_2, \ldots, x_l)$, where $x(j) = 0$ if u_j is on the path labeled 0 of $SW(j)$ and $x(j) = 1$ if u_j is on the path labeled 1 of $SW(j)$.

Now let us study the edges connecting a vertex $u = (u_1, u_2, \ldots, u_l)$ of $\mathcal{P}$, which is on layer $i-1$ and whose label is $(x_1, x_2, \ldots, x_l)$, to vertices on layer i. If $j \neq i$, then in $SW(j)$, u_j has only one edge going down. Thus, if there is an edge going down from u to v, in $\mathcal{P}$, then u_j and v_j are on the same down path of $SW(j)$, and therefore have the same bit as their label. However, if $j = i$, then in $SW(j)$, u_j has two edges going down, one to the path labeled 0 and one to the path labeled 1. These two edges cause u to have two down going edges in $\mathcal{P}$, in one of them the j'th bit of the label is 0, and in the other it is 1.

Clearly, this is the structure of $\mathcal{B}$, proving the equality. □

Certain properties of $\mathcal{B}$ are simple consequences of Theorem 3.1. First, since a binary tree has the banyan property, by Theorem 2.1, $\mathcal{B}$ has the banyan property. More interesting is the following immediate corollary:

Corollary 3.2 There is a layer preserving isomorphism between the butterfly network and its mirror image, resulting from flipping top to bottom.

Additional properties which become transparent by Theorem 3.1 are:

1. A layered subgraph of $\mathcal{B}$, which consists of $0 < k \leq l+1$ consecutive layers and the edges connecting them in $\mathcal{B}$, (while all other vertices and edges are removed), consists of 2^{l+1-k} components, each of which is a butterfly network of k layers.
2. For every two vertices in any of the layers of $\mathcal{B}$, there is a layer preserving automorphism of $\mathcal{B}$ which maps each of these vertices to the other.

4 Mesh of Trees

The *mesh of trees* was first introduced by Leighton in 1983, [7]. It can be described as follows.

Let $n = 2^l$. Consider n^2 vertices arranged in a $n \times n$ matrix (but so far no edges are introduced). For every row of the matrix, construct a binary tree of height l, whose leaves are the n vertices of the row, and the remaining $n-1$ vertices are new. Repeat for every column. Now put the roots of the row-trees on layer 0, the roots of the column-trees on layer $2l$; the remaining vertices are assigned to layers (in a unique way) to maintain the requirement that all edges connect vertices of adjacent layers. The mesh of trees, for $l = 2$ is shown in Fig. 3(a).

Clearly, the number of vertices is $3n^2 - 2n$ and the number of edges is $4n^2 - 4n$.

For a given l, define a *stalactite* to be a layered graph with $2l + 1$ layers, the top layer is labeled 0 and the bottom one is labeled $2l$. In its $l + 1$ lower layers it has a binary tree whose root is on layer $2l$ and its n leaves are on layer l. In addition, there are n disjoint paths; each leads from the top level down to one of the tree-leaves. (An example, for $l = 2$ is shown on the l.h.s. of Fig. 3(b).)

A *stalagmite* is a mirror image of a stalactite.

Theorem 4.1 A mesh of trees is the LCP of a stalactite and a stalagmite.

For example, the LCP of the stalactite and stalagmite, shown in Fig. 3(b), yields the mesh of trees shown in Fig. 3(a).

Proof: Layer l of the stalactite consists of n vertices. Name them $r_1, r_2, \ldots, r_n$, from left to right. Similarly, name the vertices of layer l of the stalagmite, $c_1, c_2, \ldots, c_n$. Layer l of their LCP consists of n^2 vertices:

$$\{(r_i, c_j) \mid 1 \leq i, j \leq n\}.$$

For a fixed i, in the stalactite, there is a simple path from the top layer down to r_i. If we restrict our attention to the top $l+1$ layers, each such path acts as the identity element of the LCP, producing a copy of the binary tree of the stalagmite. The leaves of this binary tree are the vertices on the i'th row of the matrix. A similar observation follows for the columns. □

Obviously, the mesh of trees has the banyan property. Furthermore, given any set of (disjoint) paths of length $2l$, connecting vertices of the top level with vertices of the bottom level, and any two "free" vertices, u on top and v at the bottom, there is a unique path of length l connecting them, and it is disjoint from any other of the above mentioned paths. i.e. the mesh of trees is nonblocking (in the strict sense; for definitions see Pippenger [4]), by having a "dedicated" path for every pair of free vertices.

It is interesting to mention that n^2 vertices and n^2 edges can be removed, by putting edges which short-cut the paths of length 2 passing through layer l. The end result maintains the property mentioned above and its structure can be described as the LCP of a stalactite and a stalagmite of $2l$ layers, in which each of the paths is shortened by one edge. The case of $l = 2$ is shown in Fig. 4(a).

5 Fat-Trees

The concept of fat-trees was first introduced by Leiserson, [8]. We will consider here a special type of a fat-tree, layer isomorphic to one described by Leighton, et.al. [9].

The definition of a fat-tree makes use of a layered tree, called *quad-tree*: Its root, on layer 0, has 4 sons on level 1, and each of the 4^i vertices on layer i, $0 \leq i < l$, has 4 sons on level $i+1$. To get the *fat-tree*, each vertex u on layer i, $0 \leq i \leq l$, of the quad-tree, is duplicated 2^{l-i} times. The duplicates are labeled $(u, 0), (u, 1), \ldots, (u, 2^{l-i} - 1)$. If in the quad-tree there is an edge $\langle u, v \rangle$, where u is on layer i and v is on layer $i+1$, then for every j, $0 \leq j \leq 2^{l-i-1} - 1$, (v, j) is connected by edges to $(u, 2j)$ and $(u, 2j+1)$. The case of $l = 2$ is shown in Fig. 4(b).

Theorem 5.1 A fat-tree is the LCP of a quad-tree, with its root up, and a binary tree, with its root down.

Proof: In a binary tree with its root down, if we label the vertices on the i'th layer by the integers $0, 1, \ldots, 2^{l-i} - 1$, then it follows that the sons of the vertex labeled j on the $(i + 1)$'st layer, are the vertices on the i'th layer labeled $2j$ and $2j + 1$. It is now a simple matter to see that the LCP of a quad-tree and a binary tree is a fat-tree. □

Again, it follows immediately from Theorem 2.1 that a fat-tree has the banyan property.

Acknowledgement

The authors would like to thank Bill Aiello for his advice and encouragement.

References

[1] V. Beneš, "Permutation Groups, Complexes, and Rearrangeable Multistage Connecting Network", *Bell Sys. Tech. J.*, Vol. 43, 1964, pp. 1619-1640.

[2] A. Waksman, "A Permutation Network", *J. of the ACM*, Vol. 15, No. 1, 1968, pp. 159-163.

[3] E. Upfal, "An $O(\log N)$ Deterministic Packet-Routing Scheme", *J. of the ACM*, Vol. 29, No. 1, 1992, pp. 55-70.

[4] N. Pippenger, "Telephone Switching Networks", *Proceedings of Symposia in Applied Mathematics*, Vol. 26, American Mathematical Society, 1982, pp. 101-133.

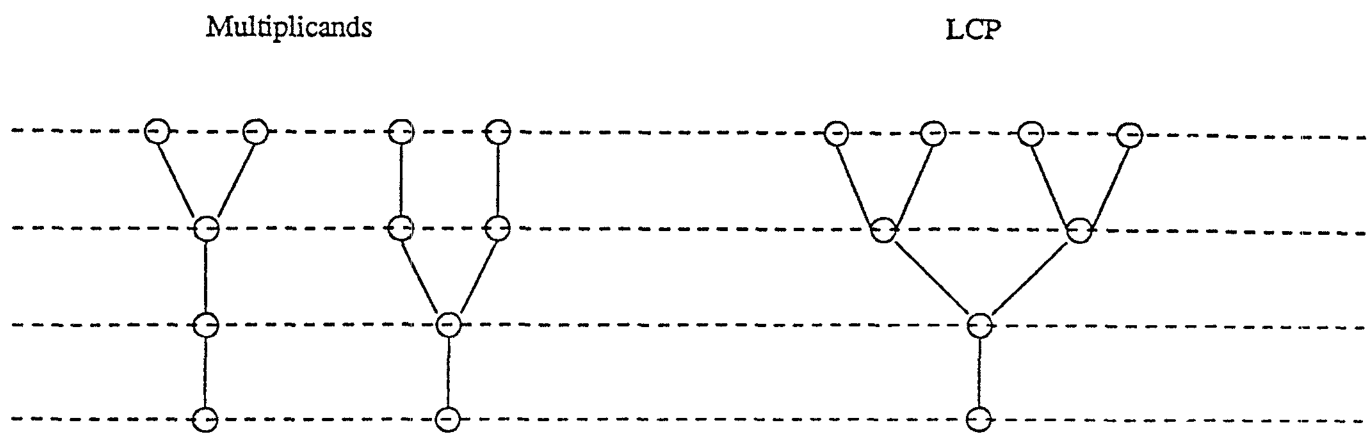
[5] G.R. Goke and G.J. Lipovski, "Banyan Networks for Partitioning Multiprocessor Systems", *1st Ann. Symp. on Computer Architecture*, Florida, U.S.A., 1973, pp. 21-28.

[6] C.P. Kruskal and M. Snir, "A Unified Theory of Interconnection Network Structure", *Theoretical Comp. Sci.*, Vol. 48, 1986, pp. 75-94.

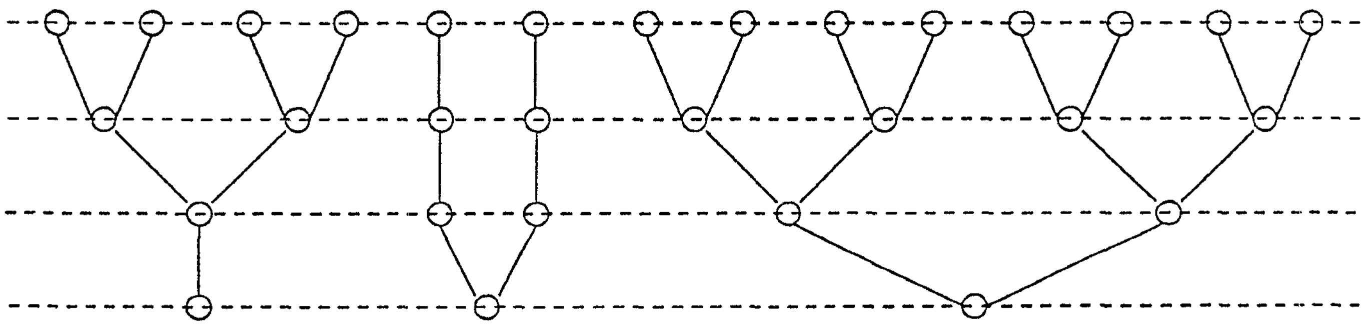
[7] T. Leighton, "Parallel Computation Using Meshes of Trees", *Proceedings of the WG'83, International Workshop on Graphtheoretic Concepts in Computer Science*, edited by M. Nagl and J. Perl, Trauner Verlag, 1984, pp. 200-218.

[8] C.E. Leiserson, "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing", *IEEE Trans. on Comp.*, Vol. c-34, No. 10, 1985, pp. 892-901.

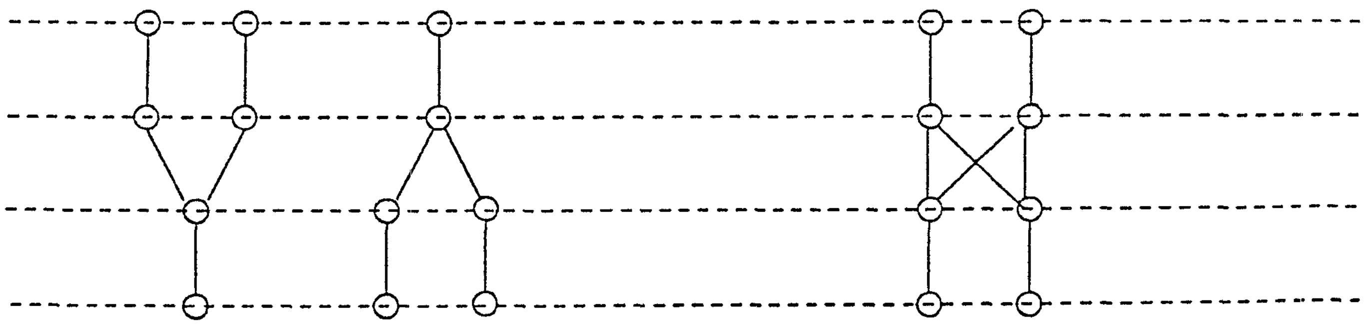
[9] T. Leighton, B. Maggs and S. Rao, "Universal Packet Routing Algorithms", *29th Annual Symp. on Foundations of Comp. Sci.*, IEEE Computer Soc. Press, 1988, pp. 256-269.



(a)

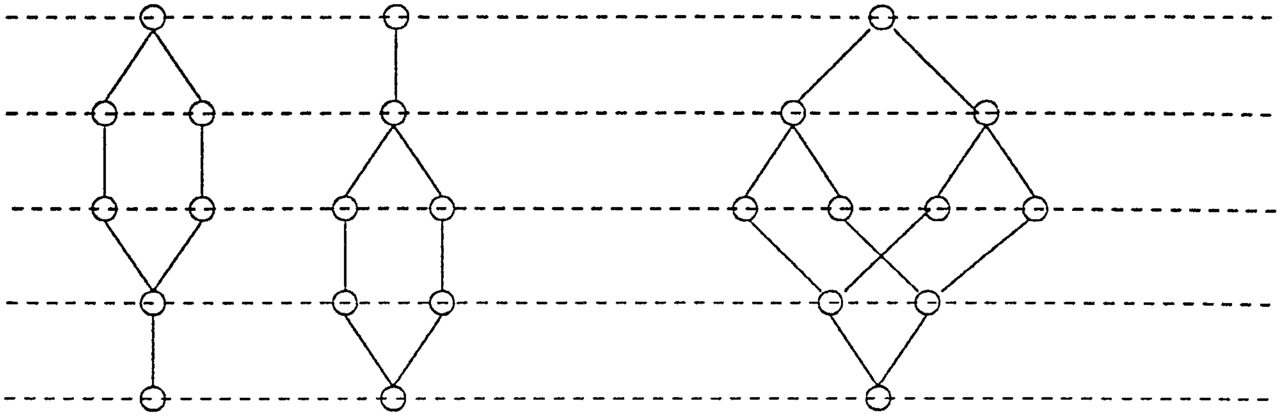


(b)

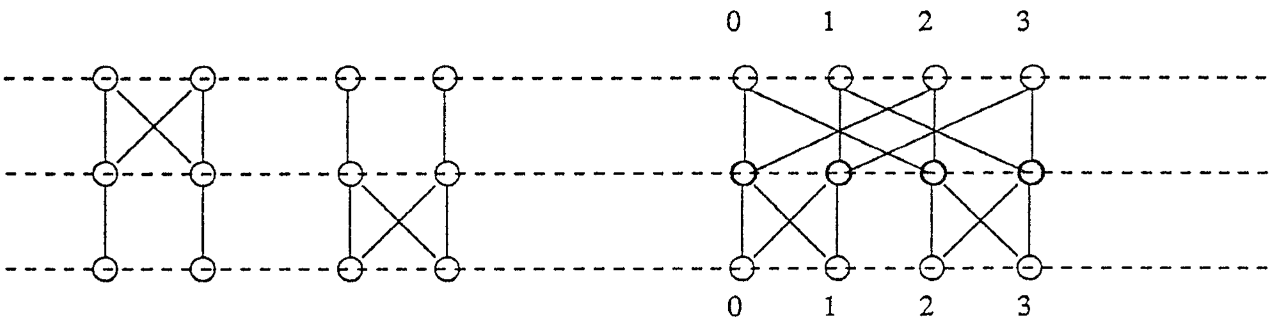


(c)

Fig. 1

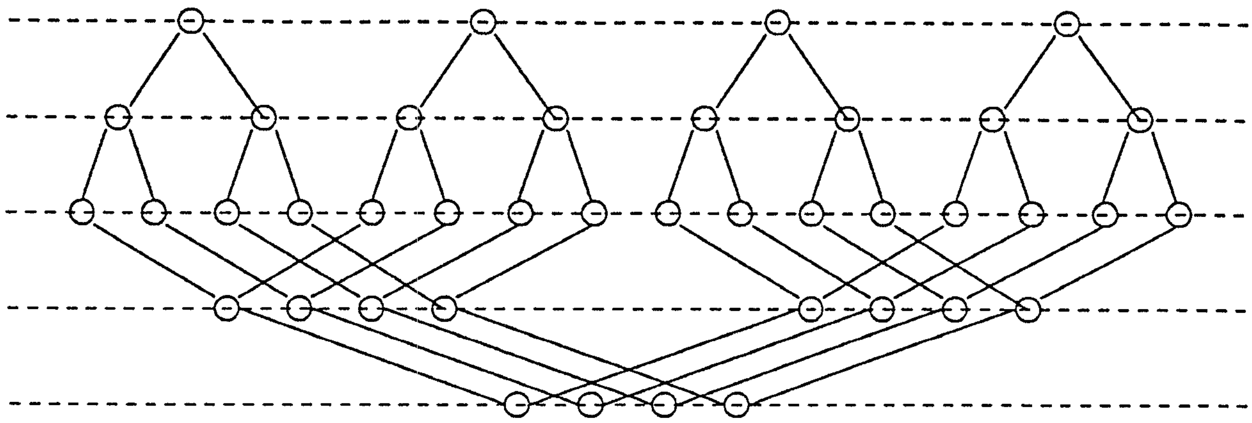


(a)

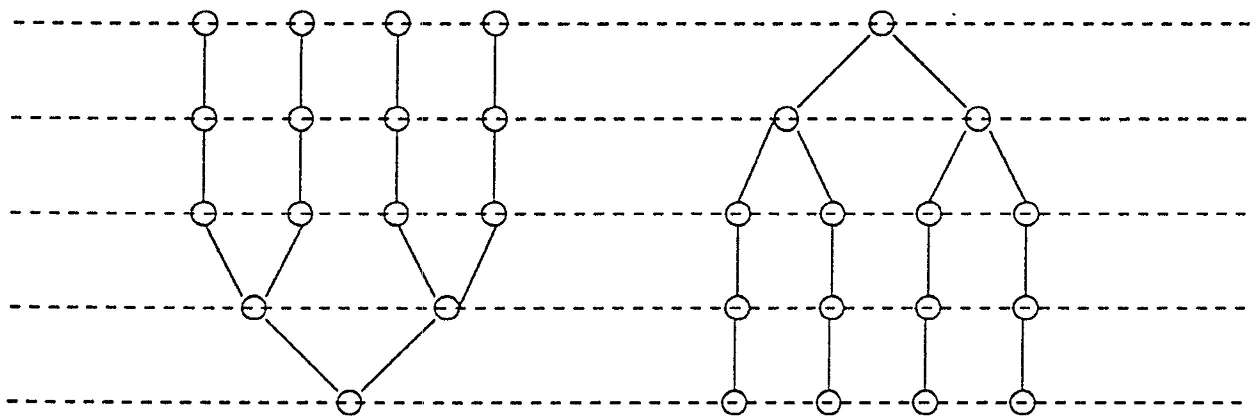


(b)

Fig. 2

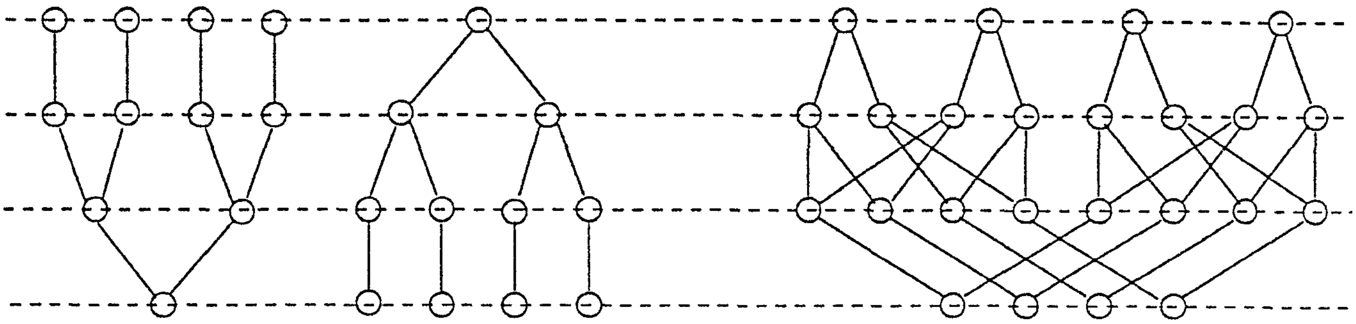


(a)

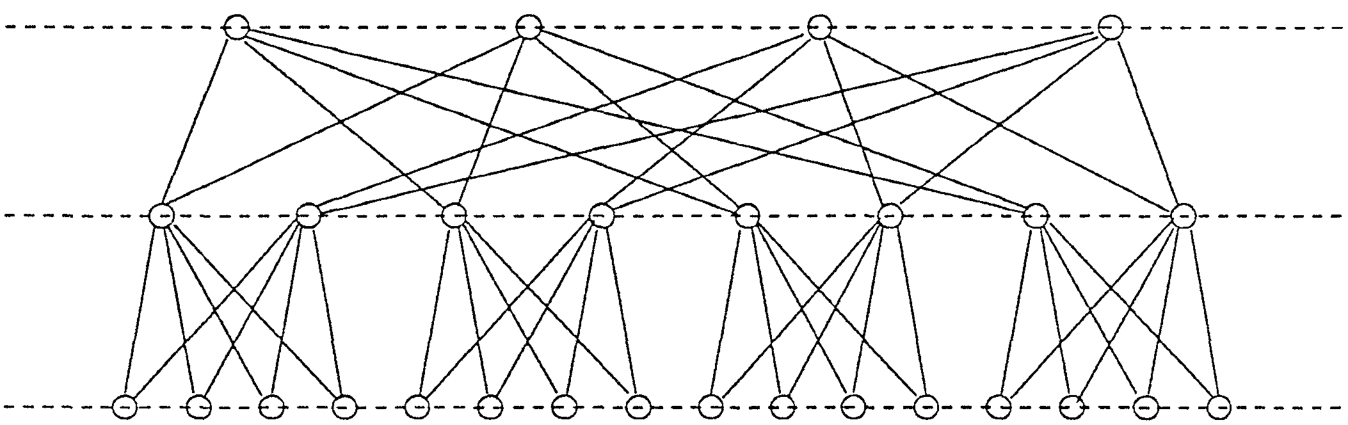


(b)

Fig. 3



(a)



(b)

Fig. 4

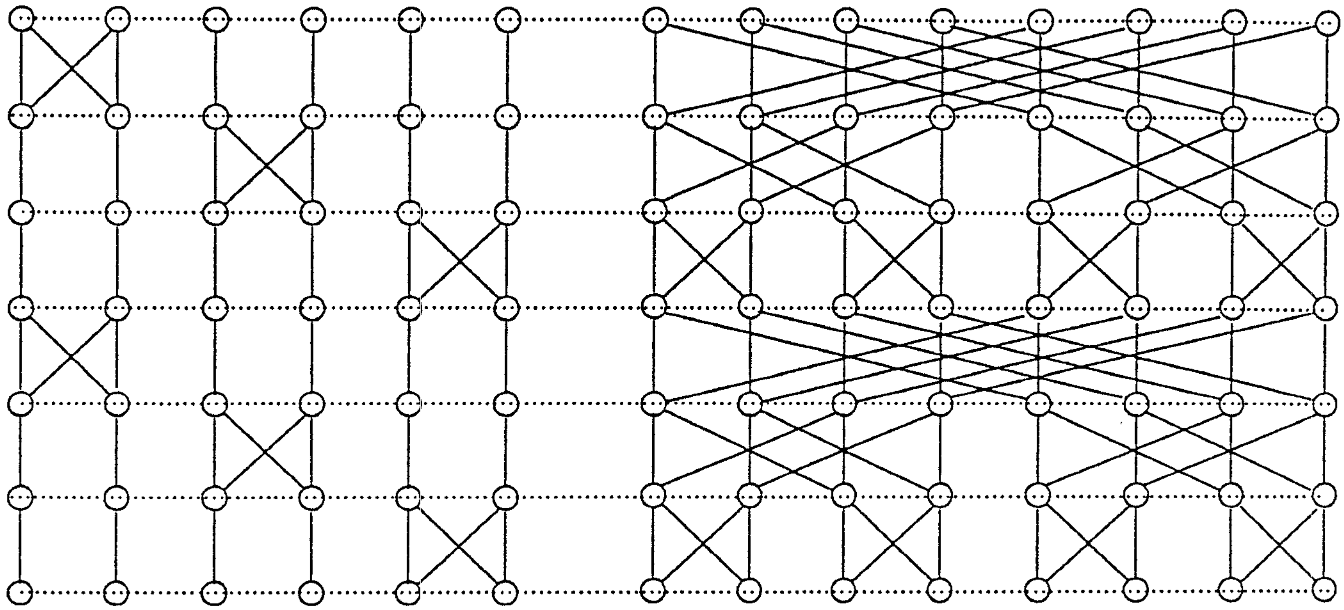


Fig. 5