INFORMATION AND COMPUTATION 114, 315–328 (1994)

# Combinatorial Properties of Dependence Graphs

A. EHRENFEUCHT

*Department of Computer Science, University of Colorado at Boulder, Boulder, Colorado 80309*

H. J. HOOGEBOOM

*Department of Computer Science, Leiden University, P.O. Box 9512, 2300 RA Leiden, the Netherlands*

AND

G. ROZENBERG

*Department of Computer Science, University of Colorado at Boulder, Boulder, Colorado 80309; and Department of Computer Science, Leiden University, P.O. Box 9512, 2300 RA Leiden, the Netherlands*

Combinatorial properties of dependence graphs are considered. In particular, a characterization is given of sets of unlabelled transitive graphs that can be labelled according to a dependence alphabet. 

## INTRODUCTION

The *theory of traces* (introduced in [Maz77]; see also [AalRoz88, Die90]) is a successful mathematical framework for modelling and studying concurrent systems. Within this theory the notion of a *dependence graph* formalizes the notion of a process in a concurrent system. For this reason the investigation of dependence graphs constitutes an important research line in the theory of traces (see [EhrRoz87, AalRoz88, EhrHooRoz90]).

In trace theory the description of a concurrent system is given by a *trace system*, which consists of a finite alphabet $\Sigma$ (giving the atomic events of the system), a binary relation $D$ over $\Sigma$ (representing the *dependence* between events), and a language $K$ over $\Sigma$ (consisting of the *sequential observations* of the behavior of the system); the pair $(\Sigma, D)$ is usually called the *dependence alphabet* of the system. A pair of events (i.e., letters



0890-5401/94 $6.00

from $\Sigma$) is called *dependent* if it is an element of $D$, and *independent* otherwise ($D$ is assumed to be a symmetric relation). Two sequential observations in $K$ are *equivalent*, i.e., considered to be observations of the same "concurrent run" of the system, if one of them can be obtained from the other by (repeatedly) interchanging two adjacent letters that are independent. This leads to the notion of a *trace*, a set of equivalent strings over $\Sigma$. The dependence of events within such a trace can be more adequately represented by a graph, the *dependence graph* of the trace. It is a directed acyclic graph in which the nodes are labelled by the events and the edges represent the ordering between *dependent* events. Consequently, the concurrent behaviour of a trace system is more adequately represented by the *set* of dependence graphs, corresponding to sequential observations from the language $K$.

In other words, given a trace system $M$, its set of dependence graphs $\mathscr{D}(M)$ consists of the set of graphs resulting from the set of sequential observations of $M$ by breaking it down through the dependence relation of $M$. A way to understand the *structure* of $\mathscr{D}(M)$ is to investigate the properties of the set of unlabelled graphs obtained from $\mathscr{D}(M)$ by erasing the labels of its nodes. A natural question here is: Is a given set $\mathscr{G}$ of unlabelled (acyclic directed) graphs "dependence graph consistent"? That is, does there exist a dependence alphabet $\Gamma$ which can be used to label $\mathscr{G}$ so that the resulting set of labelled graphs is a set of dependence graphs? (Note that in general $\mathscr{G}$ may be an infinite set, whereas $\Gamma$ has to be finite.) If the answer to this question is positive, then the set $\mathscr{G}$ is indeed consistent in the sense that if from one graph in $\mathscr{G}$ one deduces that the pair of labels $(a, b)$ should be dependent in a dependence alphabet $\Gamma$ labelling $\mathscr{G}$, then one will not deduce from another graph in $\mathscr{G}$ that $(a, b)$ should *not* be dependent in $\Gamma$.

In this paper we demonstrate that one can obtain a combinatorial answer to the above question: one can obtain a necessary and sufficient condition $\mathscr{C}$ which is of a combinatorial nature. In proving this result we introduce the notion of a spine graph for a given graph, capturing the essential information based on which nodes of the graph can be given the same label; we think this notion is of independent combinatorial interest.

## 1. Preliminaries

We assume the reader to be familiar with basic notions of graph theory—the aim of this section is to settle the specific notation and terminology concerning graphs that we will use in this paper.

*Graphs.* This paper will be concerned with (finite nonempty) *directed acyclic* graphs only; hence a directed acyclic graph is referred to simply as a graph. A graph $g$ is specified in the form $(V, E)$, where $V$ is the set of nodes of $g$ and $E$ is the set of (directed) edges of $g$; for a given graph $g$ we will sometimes use $V_g$ to denote $V$ and $E_g$ to denote $E$.

Let $g = (V, E)$ be a graph. The *Hasse graph* of $g$, denoted hasse($g$), is the graph $(V, E_{\text{hasse}})$, where $(x, y) \in E_{\text{hasse}}$ if and only if $(x, y) \in E$ and there exist no path in $g$ from $x$ to $y$ containing more than one edge. The *transitive closure* of $g$, denoted tr($g$), equals $(V, E_{\text{tr}})$, where $(x, y) \in E_{\text{tr}}$ if and only if there exists a path in $g$ from $x$ to $y$.

It is well known that the hasse graph and the transitive closure of a (directed acyclic) graph contain the same amount of information concerning the partial order defined by the graph; i.e., for graphs $g_1$ and $g_2$, $\text{hasse}(g_1) = \text{hasse}(g_2)$ if and only if $\text{tr}(g_1) = \text{tr}(g_2)$. Since these notions are defined in terms of edges in a graph, and do not change the nodes of a graph, they carry over to node-labelled acyclic graphs through their underlying graphs.

A *topological ordering* of a graph $g$ is a linear ordering $x_1, ..., x_n$ of the nodes of $g$ which satisfies the partial ordering defined by $g$; i.e., $(x_i, x_j) \in E_g$ implies $i < j$.

Let $g_1 = (V_1, E_1)$, $g_2 = (V_2, E_2)$ be graphs. A function $\psi: V_1 \to V_2$ is a *homomorphism of* $g_1$ *into* $g_2$ if, for all $u, v \in V_1$, $(u, v) \in E_1$ if and only if $(\psi(u), \psi(v)) \in E_2$; $\psi$ is an *isomorphism* if $\psi$ is also bijective. If $g_1$ and $g_2$ are *isomorphic*, then we write $g_1 =_{\text{is}} g_2$.

A *graph language* is a set of graphs.

A *(node-) labelling* of a graph $g$ is a (total) function on $V_g$. A *(node-) labelled graph* is a triple $(V, E, \varphi)$ where $(V, E)$ is a graph and $\varphi$ is a labelling of $(V, E)$. For a node-labelled graph $h = (V, E, \varphi)$ we use $V_h$, $E_h$, and $\varphi_h$ to denote $V$, $E$, and $\varphi$ respectively. The graph $(V, E)$, called the *underlying graph of h*, is denoted by und($h$).

A *(node-) labelled graph language* is a set of node-labelled graphs. For a node-labelled graph language $K$, $\text{und}(K) = \{\text{und}(h) : h \in K\}$.

*Traces and Dependence Graphs.* Given an alphabet $\Sigma$, $\Gamma = (\Sigma, D)$ is called a *dependence alphabet* (*over* $\Sigma$) if $D$ is a symmetric and reflexive relation over $\Sigma$; $D$ is called the *dependence relation of* $\Gamma$. If $(a, b) \in D$, then $a$ and $b$ are called *dependent*. Let $\equiv_\Gamma$ be the least congruence over $\Sigma^*$ that satisfies $ab \equiv_\Gamma ba$ for $a, b \in \Sigma$ and $(a, b) \notin D$, i.e., two words in $\Sigma^*$ are equal modulo $\equiv_\Gamma$ if one can be obtained from the other by interchanging repeatedly a pair of adjacent non-dependent letters. Elements of the quotient monoid $\Sigma^*/\equiv_\Gamma$ are called *traces*.

For a word $w = a_1 \cdots a_n$ over $\Sigma$, with $n \geqslant 1$ and $a_i \in \Sigma$ for $i \in \{1, ..., n\}$, the *canonical $\Gamma$-dependence graph of* $w$, denoted $\langle w \rangle_\Gamma$, is the node-labelled graph $g = (V, E, \varphi)$ with $V = \{1, ..., n\}$, such that $\varphi(i) = a_i$ for all $i \in V$, and for all $i, j \in V$, $(i, j) \in E$ if and only if $i < j$ and $(a_i, a_j) \in D$. For $w \in \Sigma^+$, any

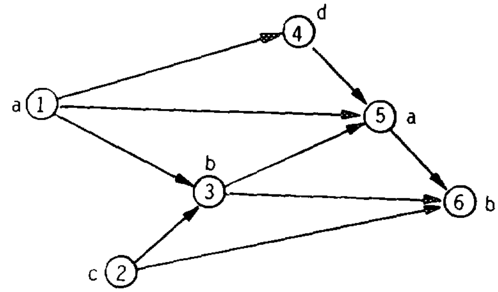


FIGURE 1

node-labelled graph isomorphic with $\langle w \rangle_\Gamma$ is called a *$\Gamma$-dependence graph (of $w$)*.

It is well known (see, e.g., [AalRoz88]) that dependence graphs faithfully represent traces in the sense that two words are equal modulo $\equiv_\Gamma$ if and only if their $\Gamma$-dependence graphs are isomorphic.

A node-labelled graph is a *dependence graph* if it is a $\Gamma$-dependence graph for some dependence alphabet $\Gamma$. A *naked dependence graph* is an unlabelled graph $g$ such that $g = \text{und}(h)$ for a dependence graph $h$.

1.1. EXAMPLE. Let $\Gamma = (\Sigma, D)$ be the dependence alphabet where $\Sigma = \{a, b, c, d\}$, and where $D$ contains the pairs $(a, b)$, $(a, d)$ and $(b, c)$ (as well as the obvious pairs needed to make $D$ a symmetric and reflexive relation).

The $\Gamma$-dependence graph $\langle w \rangle_\Gamma$ of the word $w = acbdab$ is given in Fig. 1.

## 2. $\Gamma$-LABELLINGS AND SPINES

Assume we are given a dependence alphabet $\Gamma$, a transitive graph $g$, and a labelling of $g$. If we want to check whether the labelling makes $g$ the transitive closure of a $\Gamma$-dependence graph, we have to verify at least the following two properties. First, every pair of nodes with dependent labels should be connected by an edge. Second, if two nodes are connected by an edge in hasse($g$), i.e., this edge is not the shortcut of some path in $g$, then this edge should connect two nodes labelled by dependent letters.

This leads to the following notion.

2.1. DEFINITION. Let $g = (V, E)$ be a transitive graph, and let $\Gamma = (\Sigma, D)$ be a dependence alphabet. A mapping $\varphi: V \to \Sigma$ is called a *$\Gamma$-labelling* of $g$, if

i. $(\varphi(x), \varphi(y)) \in D$, $x \neq y$, implies that either $(x, y) \in E$ or $(y, x) \in E$, and

ii. $(x, y) \in E_{\text{hasse}(g)}$ implies that $(\varphi(x), \varphi(y)) \in D$.

It is clear that labelling functions of the transitive closures of $\Gamma$-dependence graphs are examples of $\Gamma$-labellings. In fact, the two requirements in the definition above turn out to be sufficient to characterize transitive closures of dependence graphs in terms of their labellings. This is shown in the next result.

2.2. LEMMA. *Let $g = (V, E)$ be a transitive graph, and let $\Gamma = (\Sigma, D)$ be a dependence alphabet. The mapping $\varphi$ is a $\Gamma$-labelling of $g$ if and only if $(V, E, \varphi)$ is the transitive closure of a $\Gamma$-dependence graph.*

*Proof.* (1) Let $h = (V, E, \varphi)$ be the transitive closure of a dependence graph. We verify the two requirements from the definition of a $\Gamma$-labelling.

i. If the labels of two nodes are dependent, then according to the definition of dependence graphs, there should be an edge between these nodes (in one direction or the other). Obviously this edge remains present in the transitive closure of the graph.

ii. Edges present in $h$ where either present in the original dependence graph or added when taking the transitive closure. The edges present in hasse($h$) cannot have been added and are contained in the dependence graph. Hence they connect nodes with dependent labels.

(2) Let $\varphi$ be a $\Gamma$-labelling of the transitive graph $g$, and consider a topological ordering $x_1, ..., x_n$ of the nodes of $g$. We show that $h = (V, E, \varphi)$ is the transitive closure of the dependence graph $\delta = (V, E_\delta, \varphi)$ of $w = \varphi(x_1) \cdot \varphi(x_2) \cdot \cdots \cdot \varphi(x_n)$; i.e., we show that $h = \text{tr}(\delta)$. Of course $\delta$ can be seen to be isomorphic to the canonical dependence graph $\langle w \rangle_\Gamma$ using the isomorphism which maps node $i$ of $\langle w \rangle_\Gamma$ into node $x_i$ of $\delta$. It suffices to demonstrate that $E = E_{\text{tr}(\delta)}$.

i. Due to the given isomorphism between $\delta$ and $\langle w \rangle_\Gamma$, $(x_i, x_j) \in E_\delta$ implies that $(\varphi(x_i), \varphi(x_j)) \in D$ and $i < j$. Hence either $(x_i, x_j) \in E$ or $(x_j, x_i) \in E$; the latter edge, however, is not present, because we have assumed that the sequence $x_1, ..., x_n$ forms a topological ordering of the nodes of $g$. This of course shows $E_\delta \subseteq E$, which implies that $E_{\text{tr}(\delta)} \subseteq E$.

ii. It remains to be demonstrated that $E \subseteq E_{\text{tr}(\delta)}$. Clearly, since both $h$ and tr($\delta$) are transitive, we may as well show that the hasse graph of $h$ is included in $\delta$. By definition, $(x_i, x_j) \in E_{\text{hasse}(h)} = E_{\text{hasse}(g)}$ implies that $(\varphi(x_i), \varphi(x_j)) \in D$ (and $i < j$), and consequently $(x_i, x_j) \in E_\delta$. Hence $E_{\text{hasse}(h)} \subseteq E_\delta$. ∎

Now assume that we are given an unlabelled transitive graph $g$, and ask ourselves the question whether two specific nodes of $g$ can be labelled with the same letter by some $\Gamma$-labelling of $g$, where $\Gamma$ is an unspecified dependence alphabet. First, of course, the two nodes—having dependent labels—should be connected by an edge. Second, any node which is connected to one of the nodes by an edge in hasse($g$) must have a label that is dependent on the common label of the two nodes. Consequently, it should also be connected to the other node (not necessarily in hasse($g$)).

We formalize these two observations in the notion of the spine of a graph.

2.3. DEFINITION. Let $g = (V, E)$ be a transitive graph. The *spine* of $g$, denoted by sp($g$), is the graph $(V, E_{sp})$, where $(x, y) \in E_{sp}$ if and only if

a. $(x, y) \in E$,

b. For each $z \in V$, $(x, z) \in E_{hasse(g)}$ implies $z = y$ or $(z, y) \in E$, and

c. For each $z \in V$, $(z, y) \in E_{hasse(g)}$ implies $x = z$ or $(x, z) \in E$.

As for previous notions, this notion carries over to node-labelled transitive graphs through their underlying unlabelled graphs.

2.4. EXAMPLE. Let $g$ be the graph from Fig. 2.1, which is the transitive closure of the dependence graph from Fig. 1, Example 1.1. The hasse graph and the spine of $g$ are given in Fig. 2.2 and Fig. 2.3, respectively.

As an illustration of the construction of sp($g$), note that (2, 5) is not an edge in sp($g$) because (4, 5) is an edge in hasse($g$), while (2, 4) is not present in $g$ (cf. c from the above definition). ∎

Note that in b and c of the definition of the spine of a graph only nodes $z$ are considered that are either direct successors of $x$, or direct predecessors of y. It is not necessary to consider the direct predecessors of $x$ or the direct successors of $y$, since—by transitivity—these nodes are also connected to $y$ and $x$, respectively.

The spine of a transitive graph is again transitive. This will simplify both the construction and the representation of spines, since we can represent a spine by its hasse diagram.

2.5. LEMMA. *Let* $g = (V, E)$ *be a transitive graph. Then* sp($g$) *is transitive.*

*Proof.* Elementary, using the transitivity of $g$. ∎

As the following result shows, the spine at least partially fulfills its purpose: whenever two nodes of a transitive graph can be given the same

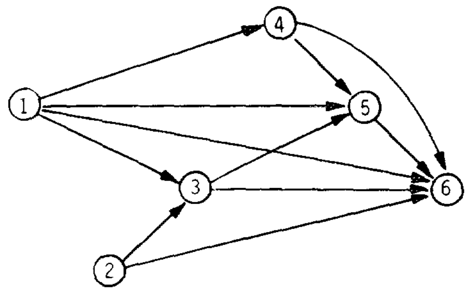


FIGURE 2.1

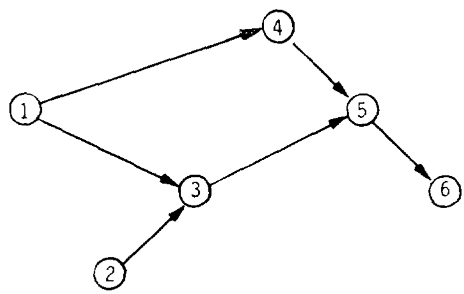


FIGURE 2.2

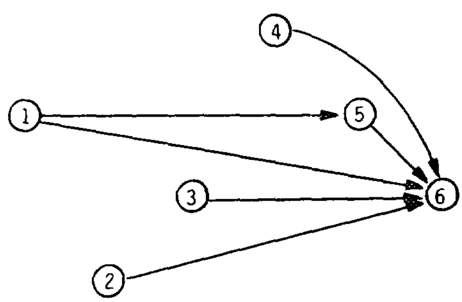


FIGURE 2.3

label by some $\Gamma$-labelling, then they are connected in the spine of the graph. This property of course motivated the introduction of the notion of a spine.

2.6. LEMMA. *Let $g=(V, E)$ be a transitive graph, let $\Gamma$ be a dependence alphabet, and let $\varphi$ be a $\Gamma$-labelling of $g$. If $\varphi(x)=\varphi(y)$ and $x \neq y$, then $x$ and $y$ are connected by an edge in* $\mathrm{sp}(g)$.

*Proof.* We check the requirements from the definition of the spine of a graph.

a. By the definition of $\Gamma$-labelling, nodes that get the same label are connected by an edge in $g$. Assume that $(x, y) \in E$ (the case that $(y, x) \in E$ is treated analogously).

b. Choose $z \in V$ such that $(x, z) \in E_{\mathrm{hasse}(g)}$. By the definition of a $\Gamma$-labelling this implies that $(\varphi(x), \varphi(z)) \in D$. Consequently, we conclude that $(\varphi(y), \varphi(z)) \in D$, which implies that either $(y, z) \in E$ or $(z, y) \in E$. The first of these alternatives is impossible because we have assumed that $(x, y) \in E$; $(y, z) \in E$ would contradict $(x, z) \in E_{\mathrm{hasse}(g)}$.

c. Symmetric to b above. ▌

Alternatively, this result shows that if two nodes are unconnected in the spine of a graph, then these nodes should get different labels if we want to construct a $\Gamma$-labelling of the graph, making it the transitive closure of a $\Gamma$-dependence graph. Hence the maximal number of nodes in a completely unordered (unconnected) subgraph of the spine gives a lower bound on the size of the dependence alphabet $\Gamma$. We use the following terminology related to this number.

2.7. DEFINITION. Let $g=(V, E)$ be a transitive graph.

(1) $\mathrm{maxco}(g)=\max\{\#U \mid U \subseteq V \text{ and } U \times U \cap E=\varnothing\}$.

(2) $g$ is called a *line graph* if $E$ can be written as $\{(x_i, x_j) \mid 1 \leqslant i<j \leqslant n\}$ for some ordering $x_1, \ldots, x_n$ of $V$.

Obviously, a line graph corresponds to a totally ordered subset of the nodes of $g$. The value $\mathrm{maxco}(g)$ on the other hand equals the maximal number of nodes of $g$ in a totally unordered subset. It is well known that $\mathrm{maxco}(g) \leqslant n$ if and only if the (nodes of the) graph $g$ can be covered by (the nodes of) $n$ (not necessarily disjoint) line graphs (see, e.g., [Bog70]). Note that such a covering contains no edges other than those in $g$, but may very well leave some of the edges of $g$ uncovered.

As discussed above, the information present in the spine of a graph will help in constructing a dependence alphabet $\Gamma$ and a $\Gamma$-labelling of a given graph. We will show that this construction can be done in a uniform way for any set of graphs for which the maxco-values of the spines can be bounded by some constant.

2.8. THEOREM. *Let $n \geqslant 1$. There exists a dependence alphabet $\Gamma$ such that, for each transitive graph $g$ with* maxco$(sp(g)) \leqslant n$, *there is a $\Gamma$-labelling of $g$.*

*Proof.* Fix an arbitrary alphabet $\Sigma$ with cardinality $n$, and let $\Sigma_1$ be the alphabet $\{(a, A) \in \Sigma \times 2^{\Sigma} \mid a \in A\}$. Consider the dependence alphabet $\Gamma = (\Sigma_1, D)$, where $(a, A)$ and $(b, B)$ in $\Sigma_1$ are dependent (with respect to $\Gamma$) if and only if $a \in B$ or $b \in A$.

Let $g = (V, E)$ be a transitive graph with maxco$(\text{sp}(g)) \leqslant n$. Because of the bound on maxco, the nodes of sp$(g)$ can be covered by $n$ (not necessarily disjoint) line graphs. Using such a covering, the nodes of sp$(g)$ can be partitioned into $n$ (disjoint) sets, such that the nodes in each set induce a line graph as subgraph of $g$. From this partition we construct a labelling $\psi: V \to \Sigma$ such that each pair of nodes with the same label is connected by an edge in sp$(g)$.

For each node $x$ we add to the label $\psi(x)$ the labels of the immediate neighbours of $x$ in hasse$(g)$:

$$\varphi(x) = (\psi(x), \{\psi(x)\} \cup \{\psi(y) \mid (y, x) \in E_{\text{hasse}(g)} \text{ or } (x, y) \in E_{\text{hasse}(g)}\}) \in \Sigma_1.$$

We verify that $\varphi$ is a $\Gamma$-labelling of $g$.

i. First, we prove that every pair of dependent nodes is conected by an edge. Let $x$ and $y$ be nodes of $g$, and let $\varphi(x) = (a, A)$ and $\varphi(y) = (b, B)$ such that $((a, A), (b, B)) \in D$. Without loss of generality we may assume that $a \in B$. Then, by construction of $\varphi$, there exists a node $z$ with label $(a, A_1)$ such that $y$ and $z$ are connected by an edge in hasse$(g)$. The nodes $x$ and $z$ are connected by an edge in sp$(g)$, since $\psi$ labels both these nodes with the same label. If $z = x$, then it is clear that $x$ and $y$ are connected by an edge in $g$: the edges of hasse$(g)$ form a subset of those of $g$. Otherwise, one considers the different possibilities for the directions of the edge between $x$ and $z$, and the edge between $y$ and $z$. In all four cases it is easily verified that $x$ and $z$ are connected by an edge in $g$. This follows either by transitivity or by the definition of the spine.

ii. Second, if two nodes are connected by an edge in hasse$(g)$, then their labels are dependent. This is an immediate consequence of the construction of the labelling $\varphi$. ∎

We give an example to illustrate the construction used in the proof.

2.9. EXAMPLE. Let $g$ be the transitive closure of the naked version of the dependence graph from Example 1.1, as represented in Fig. 1, and consider its spine, given in Fig. 2.3, Example 2.4. Clearly $\text{maxco}(\text{sp}(g)) = 4$, so a dependence alphabet $\Gamma$ which can be used to find a $\Gamma$-labelling for $g$ can be based on an alphabet with four letters, say $\Sigma = \{a, b, c, d\}$.

(1) $\text{sp}(g)$ can be covered by the "lines" $\{1, 5, 6\}$, $\{3\}$, $\{2\}$, and $\{4\}$, inducing the labelling $\psi(1) = \psi(5) = \psi(6) = a$, $\psi(3) = b$, $\psi(2) = c$, and $\psi(4) = d$. Adding context information, we get the labels $\varphi(1) = (a, \{a, b, d\})$, $\varphi(5) = (a, \{a, b, d\})$, $\varphi(6) = (a, \{a\})$, $\varphi(3) = (b, \{a, b, c\})$, $\varphi(2) = (c, \{b, c\})$, and $\varphi(4) = (d, \{a, d\})$. In this way we find a dependence alphabet with five distinct symbols.

(2) As an alternative, consider the covering by $\{1, 5\}$, $\{3\}$, $\{2\}$, and $\{4, 6\}$. After naming these lines by $a$, $b$, $c$, and $d$, respectively, we get the labels $\varphi(1) = \varphi(5) = (a, \{a, b, d\})$, $\varphi(3) = (b, \{a, b, c\})$, $\varphi(2) = (c, \{b, c\})$, and $\varphi(4) = \varphi(6) = (d, \{a, d\})$. Note that this reduces to the labelling $\varphi_1(1) = \varphi_1(5) = a$, $\varphi_1(3) = b$, $\varphi_1(2) = c$, and $\varphi_1(4) = \varphi_1(6) = d$ over the dependence alphabet $\Gamma = (\{a, b, c, d\}, D)$, where $D$ is the symmetric and reflexive relation containing the pairs $(a, b)$, $(a, d)$ and $(b, c)$. Although this is the relation we started with in Example 1.1, the labelling we have obtained differs from the original labelling. ∎

Combining Lemma 2.6 and Theorem 2.8 yields the following result.

2.10. COROLLARY. *Let $K$ be a graph language consisting of unlabelled transitive graphs. The following conditions are equivalent.*

(1) *There exists a dependence alphabet $\Gamma$ such that for each $g \in K$ there exists a $\Gamma$-labelling of $g$.*

(2) *There exists a constant $c$ such that, for each $g \in K$, $\text{maxco}(\text{sp}(g)) \leqslant c$.* ∎

## 3. SPINES AND SQUARES

The last result in the previous section states that if the maxco-values of the set of spines of a transitive unlabelled graph language can be bounded by some constant, then one can find a dependence alphabet $\Gamma$ such that all graphs in the language can be labelled to become transitive closures of $\Gamma$-dependence graphs. This is a characterization in terms of a graph-theoretical property of the spines of the graphs involved. In this section we establish a more direct characterization in terms of the graphs themselves.

We need the following notions.

3.1. DEFINITION. Let $g=(V, E)$ be a transitive graph.

(1) The nodes $x_1, ..., x_n, y_1, ..., y_n$ of $V$ *induce a square graph* (of size $n$) if

— $(x_i, x_j) \in E$ for $1 \leqslant i < j \leqslant n$,
— $(y_i, y_j) \in E$ for $1 \leqslant i < j \leqslant n$,
— $(x_i, y_i) \in E_{\text{hasse}(g)}$ for $1 \leqslant i \leqslant n$, and
— $(y_i, x_j) \notin E$ for $1 \leqslant i, j \leqslant n$.

(2) $\text{maxsq}(g) = \max\{n \mid g \text{ contains an induced square of size } n\}$. ∎

Hence, when nodes $x_1, ..., x_n, y_1, ..., y_n$ induce a square graph, then we get a situation as in Fig. 3.1, where edges required to be in the hasse graph are given as solid arcs, other edges as dashed ones. Note that we have omitted arcs that would follow by transitivity.

3.2. *Remark.* Let $g(V, E)$, with $V = \{x_1, ..., x_n, y_1, ..., y_n\}$ be a square graph; i.e., the nodes $x_1, ..., x_n, y_1, ..., y_n$ satisfy the conditions of the above definition. Then the nodes $x_1, ..., x_n$ form a totally unordered set in the spine of $g$. The edge $(x_i, x_j)$, $i<j$, is not present in $\text{sp}(g)$ because $(x_i, y_i) \in E_{\text{hasse}(g)}$, whereas $(y_i, x_j) \notin E$.

From this it is clear that $\text{maxco}(\text{sp}(g)) \geqslant n$, while $\text{maxco}(g) = 2$.

This gives an example of a class of graphs for which the maxco-values are bounded, while these values are not bounded for the corresponding spines. ∎

It is easily seen that the maxco-value and the maxsq-value of a given graph are bounded by the maxco-value of the spine of the graph.

3.3. LEMMA. *For each transitive graph* $g$, $\text{maxco}(g) \leqslant \text{maxco}(\text{sp}(g))$, *and* $\text{maxsq}(g) \leqslant \text{maxco}(\text{sp}(g))$.

*Proof.* The inequality $\text{maxco}(g) \leqslant \text{maxco}(\text{sp}(g))$ is clear since nodes that are unconnected in $g$ remain unconnected in $\text{sp}(g)$.

The inequality $\text{maxsq}(g) \leqslant \text{maxco}(\text{sp}(g))$ follows from the above remark. If the graph $g$ contains an induced square of size $n$, then one can find at least $n$ unconnected nodes in $\text{sp}(g)$. ∎

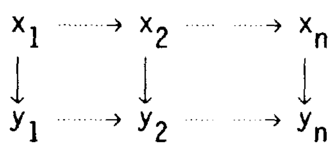


FIG. 3.1. Diagram of a square graph.

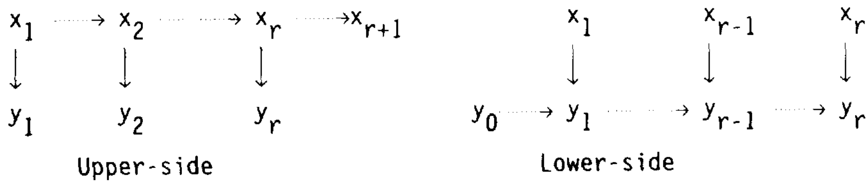


FIGURE 3.2

We will need the following notions of sequences that form "half of a square graph."

3.4. DEFINITION. Let $g=(V, E)$ be a transitive graph. A sequence $x_1, ..., x_r, x_{r+1}$ of nodes is an *upper-side of length* $r$, if

i. $(x_i, x_j) \in E$ for all $1 \leqslant i < j \leqslant r+1$, and

ii. there exist nodes $y_1, y_2, ..., y_r$ such that $(x_i, y_i) \in E_{\text{hasse}(g)}$ for each $1 \leqslant i \leqslant r$ and $(y_i, x_j) \notin E$ for all $1 \leqslant i < j \leqslant r+1$.

A sequence $y_0, y_1, ..., x_r$, of nodes is a *lower-side of length* $r$, if

i. $(y_i, y_j) \in E$ for all $0 \leqslant i < j \leqslant r$, and

ii. there exist nodes $x_1, x_2, ..., x_r$ such that $(x_i, y_i) \in E_{\text{hasse}(g)}$ for each $1 \leqslant i \leqslant r$ and $(y_i, x_j) \notin E$ for all $0 \leqslant i < j \leqslant r$. ▌

Hence, an upper-side and a lower-side correspond to situations as depicted in Fig. 3.2.

If we want to find lower-sides or upper sides in a given graph, then we have to look for lines in the graph that are completely unconnected in the spine of the graph.

3.5. LEMMA. *Let* $g=(V, E)$ *be a transitive graph and let* $r, s \in \mathbb{N}$, $r+s \geqslant 1$. *Let* $\rho = z_1, z_2, ..., z_n$, *where* $n \geqslant r+s$, *be a sequence of nodes such that, for* $1 \leqslant i < j \leqslant n$, $(z_i, z_j) \in E$ *and* $(z_i, z_j) \notin E_{\text{sp}(g)}$.

*Then either there is a subsequence* $x_1, ..., x_r, x_{r+1}$ *of* $\rho$ *that forms an upper-side (of length* $r$*), or there is a subsequence* $y_0, y_1, ..., y_s$ *of* $\rho$ *that forms a lower-side (of length* $s$*).*

*Proof.* By induction on $r+s$. When either $r=0$ or $s=0$ then the result is obvious: any single node forms an upper-side and a lower-side.

We proceed with the induction step. The proof will recursively yield a structure of the form in Fig. 3.3, where in each step either $r$ or $s$ is increased by 1.

According to the assumption of the lemma, $(z_1, z_n) \notin E_{\text{sp}(g)}$, whereas $(z_1, z_n) \in E$. This implies there is a node $z$ such that either $(z_1, z) \in E_{\text{hasse}(g)}$ and $(z, z_n) \notin E$, or $(z, z_n) \in E_{\text{hasse}(g)}$ and $(z_1, z) \notin E$.

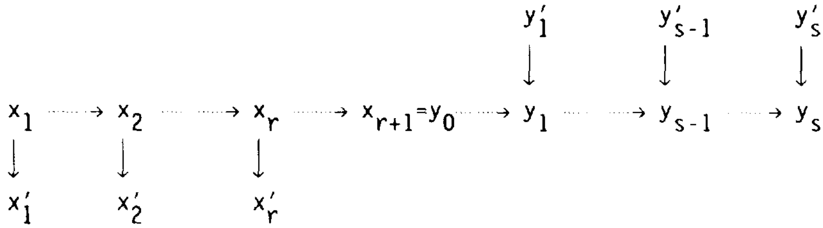


FIGURE 3.3

In the former case consider $z_2, \ldots, z_n$. By inductive assumption it contains either a subsequence of length $r-1$ that forms an upper-side, or a subsequence of length $s$ that forms a lower-side. In case of an upper-side, it can be combined with $z_1$ to form an upper-side of length $r$ since, by transitivity, $(z, z_n) \notin E$ implies that $(z, z_j) \notin E$ for $2 \leqslant j \leqslant n$. Hence we have found either an upper-side of length $r$, or a lower-side of length $s$.

The latter case is dealt with in an analogous way by considering $z_1, \ldots, z_{n-1}$. ∎

We are now able to prove some sort of converse of Lemma 3.3: the maxco-value of the spine of a graph can be bounded by a polynomial in the maxco- and maxsq-value of the graph.

3.6. THEOREM. *For each transitive graph* $g$, *if* $\text{maxco}(g) \leqslant c$ *and* $\text{maxsq}(g) \leqslant d$, *then* $\text{maxco}(sp(g)) \leqslant 2c^2 \cdot d$.

*Proof.* We will show that, if $\text{maxco}(g) \leqslant c$ and $\text{maxco}(\text{sp}(g)) > 2c^2 \cdot d$, then $\text{maxsq}(g) \geqslant d$.

Consider a set $CO$ of $2c^2 \cdot d$ nodes that are not connected by edges in $\text{sp}(g)$. Since $\text{maxco}(g) \leqslant c$, (the nodes of) $g$ can be covered by at most $c$ line graphs. One of these line graphs should contain at least $2cd$ nodes from $CO$. In this way we have found a sequence $\rho$ of at least $2cd$ nodes, such that, if $x$ precedes $y$ in $\rho$, then $(x, y) \in E$, whereas $(x, y) \notin E_{\text{sp}(g)}$.

According to Lemma 3.5, either $\rho$ contains a subsequence that forms an upper-side of length $cd$, or $\rho$ contains a subsequence that forms a lower-side of length $cd$. We will assume here that $\rho$ contains an upper-side—the case where $\rho$ contains a lower-side being analogous.

So, let the sequence $x_1, \ldots, x_{cd}, x_{cd+1}$ be an upper-side of length $cd$, and let $y_1, y_2, \ldots, y_{cd}$ be nodes such that $(x_i, y_i) \in E_{\text{hasse}(g)}$ for each $1 \leqslant i \leqslant cd$ and $(y_i, x_j) \notin E$ for all $1 \leqslant i < j \leqslant cd+1$. Again each of the nodes $y_i$ is part of (at least) one of the $c$ line graphs covering $g$. Hence there is a subsequence of length $d$ of the $y_i$ that is contained in a single line graph. Clearly these $y_i$ form, together with the corresponding $x_i$, a square graph of size $d$ that is an induced subgraph of $g$. ∎

Combining our results, we obtain the main result of this paper.

3.7. THEOREM. *Let $K$ be a graph language consisting of unlabelled transitive graphs. The following three conditions are equivalent.*

(1) *There exists a dependence alphabet $\Gamma$ such that for each $g \in K$ there exists a $\Gamma$-labelling of $g$.*

(2) *There exists a constant $c$ such that, for each $g \in K$, $\operatorname{maxco}(\operatorname{sp}(g)) \leqslant c$.*

(3) *There exist constants $c$ and $d$ such that, for each $g \in K$, $\operatorname{maxco}(g) \leqslant c$, and $\operatorname{maxsq}(g) \leqslant d$.*

*Proof.* (1) if and only if (2): this is Corollary 2.10.

(2) implies (3): this follows from Lemma 3.3.

(3) implies (2): this follows from Theorem 3.6. ∎

## ACKNOWLEDGMENTS

This paper was written within the working group ASMICS, ESPRIT Basic Research Action 3166. The authors are indebted to V. Diekert and an anonymous referee for detailed comments on a previous version of this paper.

RECEIVED September 12, 1991; FINAL MANUSCRIPT RECEIVED June 18, 1992

## REFERENCES

[AalRoz88] AALBERSBERG, IJ. J. AND ROZENBERG, G. (1988), Theory of traces, *Theoret. Comput. Sci.* **60**, 1–82.

[Bog70] BOGART, K. P. (1970), Decomposing partial orderings into chains, *J. Combin. Theory* **9**, 97–99.

[Die90] DIEKERT, V. (1990), "Combinatorics on Traces," Lecture Notes in Computer Science, Vol. 454, Springer-Verlag, Berlin/New York.

[EhrHooRoz90] EHRENFEUCHT, A., HOOGEBOOM, H. J. AND ROZENBERG, G. (1993), On the structure of recognizable languages of dependence graphs, *RAIRO, Informatique théorique et Applications* **27**, 7–22.

[EhrRoz87] EHRENFEUCHT, A. AND ROZENBERG, G. (1987), On the structure of dependency graphs, *in* "Concurrency and Nets," pp. 141–170, Springer-Verlag, Berlin/New York.

[Maz77] MAZURKIEWICZ, A (1977), "Concurrent Program Schemes and Their Interpretations," DAIMI Rep. PB-78, Aarhus University, Aarhus.