

On the Complexity of Join Dependencies

MARC GYSSENS

University of Antwerp

In [10] a method is proposed for decomposing join dependencies (jds) in a relational database using the notion of a hinge. This method was subsequently studied in [11] and [12]. We show how the technique of decomposition can be used to make integrity checking more efficient. It turns out that it is important to find a decomposition that minimizes the number of edges of its largest element. We show that the decompositions obtained with the method described in [10] are optimal in this respect. This minimality criterion leads to the definition of the *degree of cyclicity*, which allows us to classify jds and leads to the notion of *n-cyclicity*, of which acyclicity is a special case for $n = 2$. We then show that, for a fixed value of n (which may be greater than 2), integrity checking can be performed in polynomial time provided we restrict ourselves to *n-cyclic* jds. Finally, we generalize a well-known characterization for acyclic jds by proving that *n-cyclicity* is equivalent to "*n-wise consistency implies global consistency*." As a consequence, consistency checking can be performed in polynomial time if we restrict ourselves to *n-cyclic* jds, for a fixed value of n , not necessarily equal to 2.

Categories and Subject Descriptors: G.2.2 [Discrete Mathematics]: Graph Theory—*graph algorithms, trees*; H.2.1 [Database Management]: Logical Design—*normal forms, schema and subschema*

General Terms: Algorithms, Design, Theory

Additional Key Words and Phrases: Decomposition, equivalence of constraints, hypergraph, integrity checking, join dependency, multivalued dependency, relational database model

1. INTRODUCTION

Functional [6], multivalued [7, 19], and the more general join dependencies [16] are fundamental in relational database theory [5]. In [9] it was postulated that in real-world databases the structure can be expressed by a set of functional dependencies together with only one join dependency. Here we are not concerned with functional dependencies, and therefore we consider relations on which a single join dependency is defined.

Unfortunately, integrity checking for join dependencies (jds), that is, checking whether the relation still satisfies the constraint after an update, turns out to be NP-complete in general. However, integrity checking can be avoided, since the presence of a jd is a necessary and sufficient condition to decompose a relation into a set of smaller relations, one for each edge of the jd. When this approach is followed (which is most common in practice), we must only verify whether the

Author's address: M. Gyssens, Department of Mathematics and Computer Science, University of Antwerp, U.I.A., Universiteitsplein 1, 2610 Antwerpen, Belgium.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1986 ACM 0730-0301/86/0300-0081 \$00.75

ACM Transactions on Database Systems, Vol. 11, No. 1, March 1986, Pages 81-108.

subrelations still represent the information of one common relation each time an update is performed on the subrelations. This is called *consistency checking*. Unfortunately, consistency checking too is NP-complete in general.

However, there exist polynomial algorithms for integrity and consistency checking in the case where only acyclic join dependencies are involved. It is well known that the so-called acyclic join dependencies are those jds that can be decomposed into a set of multivalued dependencies (mvds) [2, 3, 8, 9, 10]. An mvd is essentially a join dependency with two edges. Because of these desirable properties, we often try to design a database in such a way that the jd describing it is acyclic. In many cases, however, the initial design produces cyclic join dependencies. Several techniques have been proposed to “change” these cyclic jds into acyclic ones (e.g., attribute splitting [9]), but these techniques may have undesirable side effects (e.g., the loss of the close relationship between split attributes). In this paper we want to point out that it is not necessary to reject all cyclic jds in database design. Indeed, some cyclic jds can also be decomposed into a set of “less complicated” join dependencies, as is illustrated by the following example. Consider the join dependency:

$$J: ab \bowtie ac \bowtie bd \bowtie cd \bowtie de \bowtie df \bowtie eg \bowtie fg.$$

A possible decomposition of J is

$$\{abcde \bowtie df \bowtie eg \bowtie fg, abcd \bowtie defg, ab \bowtie ac \bowtie cd \bowtie bdefg\}.$$

Other cyclic jds are not decomposable at all in this sense, such as

$$ab \bowtie bc \bowtie cd \bowtie de \bowtie ef \bowtie fg \bowtie ga.$$

This suggests that some cyclic jds are “more cyclic” than other cyclic jds, that is, that there exist several levels of cyclicity for jds of which acyclicity is only the “lowest” one (apart from the trivial jds).

If we look at this problem from the point of view of integrity checking, we see that replacing a jd by a decomposition can make integrity checking more efficient, since the jds in a decomposition have by definition fewer edges than the original jd. It is important to make the number of edges of the join dependencies of the decomposition we wish to use to perform integrity checking as small as possible, since the time complexity of making a join is first and foremost determined by the number of relations that have to be joined. Therefore, we define the *degree of cyclicity* of a join dependency J as the minimum overall decompositions of J of the number of edges of the largest join dependency in that decomposition. In particular, we show that this minimum in each decomposition can be obtained with the method introduced in [10]. Hence the decompositions thus obtained are optimal in this respect.

The degree of cyclicity also seems a natural tool to classify join dependencies in the sense we discussed above. We therefore say that a jd is n -cyclic if its degree of cyclicity is at most n . In this way we get a hierarchical classification of jds in which acyclic jds appear for $n = 2$. We show that integrity checking can be performed in polynomial time provided only n -cyclic jds are considered for a fixed value of n , not necessarily 2.

As said before, in practical situations, the presence of a jd will be used to decompose the relation on which it is defined in a set of smaller relations. With this approach, the problem of integrity checking is replaced by the problem of consistency checking. In the case of an acyclic jd, consistency checking can be performed in polynomial time since acyclicity is equivalent with "pair-wise consistency implies global consistency" [2, 3, 8, 9]. The notion of n -cyclicity of which acyclicity is a special case for $n = 2$ suggests a generalization of this result. We show here that n -cyclicity is indeed equivalent to " n -wise consistency implies global consistency." As a consequence, checking a decomposed database for global consistency can be done in polynomial time if one allows only n -cyclic join dependencies, for a fixed level of cyclicity n , which may be greater than 2. Hence in database design we have to restrict the jds that may occur to jds of a certain level of cyclicity, but not necessarily to acyclic jds.

This paper is organized as follows. In Section 2 we give some basic notions about join dependencies and their hypergraph representations, which we use extensively throughout this paper. Essential to our decomposition methodology is the notion of a hinge. A *hinge* is a set of edges of a hypergraph that satisfies certain properties. We define this notion in Section 3, in which we also describe our decomposition methodology. In Section 4 we give an overview of the most important properties of this decomposition methodology discussed in earlier papers. In Section 5 we define the degree of cyclicity of a join dependency and characterize this notion in terms of the structure of the join dependency itself. Again the notion of a hinge turns out to be important in this characterization. We show that our decomposition minimizes the number of components in its largest element. We also introduce n -cyclicity, and we show that integrity checking can be performed in polynomial time, provided only n -cyclic jds are considered, for a fixed value of n . Finally, in Section 6, we prove that n -cyclicity is equivalent to " n -wise consistency implies global consistency" and show that, as a consequence, consistency checking can be done in polynomial time, provided only n -cyclic jds are considered, for a fixed value of n .

2. NOTATION AND TERMINOLOGY

In this paper we consider a universal relation scheme $R(\Omega)$ (or R if no ambiguity is possible) consisting of a set Ω of attributes, each associated with a domain of values. We assume that each domain is infinitely denumerable (e.g., the set of nonnegative integers). In the sequel we denote single attributes by small letters, whereas sets of attributes are denoted by capital letters. If X and Y are sets of attributes, we write XY for $X \cup Y$. If a, b, c, \dots are attributes, we write $abc \dots$ for $\{a, b, c, \dots\}$. In particular, we do not distinguish between the attribute a and the set $\{a\}$.

Let $X \subseteq \Omega$. A tuple t over X is a mapping that associates with each attribute a of Ω a value of its corresponding domain. An instance r over X is a set of tuples over X . Let $Y \subseteq X$ and let t be a tuple over X . The projection of t onto Y , denoted $t[Y]$, is obtained by restricting t to the attributes of Y . If r is an instance over X , the set obtained by projecting each tuple of r onto Y is said to be the projection of r onto Y , denoted $\pi_Y(r)$.

Let $X_1, \dots, X_k \subseteq \Omega$. We say that (r_1, \dots, r_k) is an instance over (X_1, \dots, X_k) if r_i is an instance over X_i for all $i = 1, \dots, k$.

In most cases instances must obey certain constraints to be admissible. The constraints we consider here are the join dependencies.

Definition 2.1 [1]. Let $R(\Omega)$ be a relation scheme. Let $X_1, \dots, X_k \subseteq \Omega$ and let (r_1, \dots, r_k) be an instance over (X_1, \dots, X_k) . We define the join of r_1, \dots, r_k , denoted $r_1 \bowtie \dots \bowtie r_k$ as the union of all relation instances s over $\cup_{i=1}^k X_i$ satisfying $\pi_{X_i}(s) \subseteq r_i$, for all $i = 1, \dots, k$.

The join of r_1, \dots, r_k is therefore the “largest” relation instance over $\cup_{i=1}^k X_i$, of which the projections onto X_1, \dots, X_k are contained in r_1, \dots, r_k , respectively.

Definition 2.2 [16]. Let $R(\Omega)$ be a relation scheme. A (n -embedded) join dependency (jd) J over R is an expression of the form $X_1 \bowtie \dots \bowtie X_k$ with $X_1, \dots, X_k \subseteq \Omega$. X_1, \dots, X_k are called the edges of J . Let $X = \cup_{i=1}^k X_i$. Let r be a relation instance over Ω . r satisfies the jd $X_1 \bowtie \dots \bowtie X_k$ if $\pi_X(r) = \pi_{X_1}(r) \bowtie \dots \bowtie \pi_{X_k}(r)$. If $X = \Omega$, we say that J is a full join dependency.

A jd that consists of only two edges is called a (n -embedded) multivalued dependency (mvd) [7, 19].

Although we are mainly concerned with full jds and decompositions of full jds into full jds, we do not exclude embedded jds from our discussions wherever this restriction is not necessary.

Join dependencies can be described in hypergraphs [2, 3, 8, 9] in a very elegant way. We use this formalism very intensively further on in this paper.

Definition 2.3. A hypergraph $\mathcal{H}(\mathcal{N}, \mathcal{E})$ is a pair consisting of a set of nodes \mathcal{N} and a set of edges \mathcal{E} satisfying $\mathcal{E} \subseteq 2^{\mathcal{N}}$.

In the following definitions we give some basic notions and terminology about hypergraphs.

Definition 2.4. Let $\mathcal{H}(\mathcal{N}, \mathcal{E})$ be a hypergraph and let $\mathcal{E}' \subseteq \mathcal{E}$.

Suppose $F, G \in \mathcal{E} \setminus \mathcal{E}'$ (where “ \setminus ” stands for set difference).

– A sequence E_1, \dots, E_q of edges is a path from F to G with respect to \mathcal{E}' if

- (1) $E_1 = F$
- (2) $E_q = G$
- (3) $E_i \cap E_{i+1} \not\subseteq \cup \mathcal{E}'$ for all $i = 1, \dots, q - 1$.

A path from F to G is a path from F to G with respect to \emptyset .

– F and G are connected (with respect to \mathcal{E}') if there exists a path from F to G (with respect to \mathcal{E}').

Let $\mathcal{F} \subseteq \mathcal{E} \setminus \mathcal{E}'$, $\mathcal{F} \neq \emptyset$.

- \mathcal{F} is connected (with respect to \mathcal{E}') if every two edges of \mathcal{F} are connected (with respect to \mathcal{E}').
- \mathcal{F} is called disconnected (with respect to \mathcal{E}') if it is not connected (w.r.t \mathcal{E}').
- \mathcal{H} is called a connected hypergraph if \mathcal{E} is connected.
- \mathcal{F} is a connected component (with respect to \mathcal{E}') if it is connected, and for any $\mathcal{G} \subseteq \mathcal{E} \setminus (\mathcal{F} \cup \mathcal{E}')$ with $\mathcal{F} \not\subseteq \mathcal{G}$, \mathcal{G} is disconnected (with respect to \mathcal{E}').

Clearly, the connected components (with respect to \mathcal{E}') of a hypergraph $\mathcal{H}(\mathcal{N}, \mathcal{E})$ form a partition of $\mathcal{E} \setminus \mathcal{E}'$, since connectedness of edges (with respect to \mathcal{E}') is an equivalence relation.

Definition 2.5. Let $\mathcal{H}(\mathcal{N}, \mathcal{E})$ be a hypergraph. The reduction of \mathcal{H} is the hypergraph obtained from \mathcal{H} by deleting every edge that is properly contained in another edge. A hypergraph is called reduced if it equals its reduction.

With a jd we can associate a hypergraph, and vice-versa, in the following way:

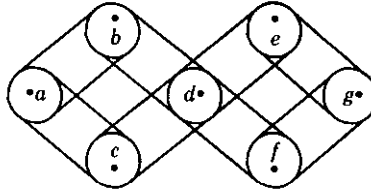
Definition 2.6. Let $R(\Omega)$ be a relation scheme.

Let $J: X_1 \bowtie \dots \bowtie X_k$ be a jd over R . The hypergraph $\mathcal{H}_J(\mathcal{N}, \mathcal{E})$ associated with J is defined by

$$\begin{aligned}\mathcal{N} &= \Omega \\ \mathcal{E} &= \{X_1, \dots, X_k\}.\end{aligned}$$

Let $\mathcal{H}(\mathcal{N}, \mathcal{E})$ be a hypergraph and suppose that \mathcal{N} is a set of attributes, $\mathcal{N} \subseteq \Omega$. Let $\mathcal{E} = \{E_1, \dots, E_k\}$. The jd $J_{\mathcal{H}}$ over R associated with \mathcal{H} is $E_1 \bowtie \dots \bowtie E_k$.

Example 2.1. The jd $ab \bowtie ac \bowtie bd \bowtie cd \bowtie de \bowtie df \bowtie eg \bowtie fg$, already mentioned in the Introduction, can be represented by the following hypergraph (if we assume $\Omega = abcdefg$):



Since there exists a very natural relationship between hypergraphs and jds, we often use the terminology designated for hypergraphs also for jds. We say that a jd is reduced or connected if the hypergraph associated with that jd is reduced or connected. In the sequel we often assume that the jds we consider are reduced and connected. It can easily be seen that this is not a real restriction.

In most cases the presence of a number of constraints in a relation scheme automatically implies that several other constraints must be satisfied as well. We therefore introduce the following definitions.

Definition 2.7. Let R be a relation scheme and let \mathcal{P} be a set of jds over R . We define by $\text{SAT}(\mathcal{P})$ the set of all relation instances over R that satisfy all the jds of \mathcal{P} .

Definition 2.8. Let R be a relation scheme. Let \mathcal{P} and \mathcal{Q} be sets of jds over R .

- \mathcal{P} implies \mathcal{Q} , denoted $\mathcal{P} \Rightarrow \mathcal{Q}$, if $\text{SAT}(\mathcal{P}) \subseteq \text{SAT}(\mathcal{Q})$.
- \mathcal{P} is equivalent to \mathcal{Q} , denoted $\mathcal{P} \Leftrightarrow \mathcal{Q}$, if $\mathcal{P} \Rightarrow \mathcal{Q}$ and $\mathcal{Q} \Rightarrow \mathcal{P}$, that is, if $\text{SAT}(\mathcal{P}) = \text{SAT}(\mathcal{Q})$.

We can now define the following:

Definition 2.9 [2, 3, 8, 9]. A join dependency is acyclic if it is equivalent to a set of multivalued dependencies. A hypergraph is acyclic if its associated jd is acyclic.

In this paper we are primarily concerned with decompositions of join dependencies. Indeed, a jd can be equivalent to a set of other, smaller jds. The knowledge of such a set of jds can teach us much about the structure of a jd [10, 11, 12]. Essential for a set of jds to be a decomposition of a given jd is that the set be equivalent to the original jd, that it contain more than one element and that each jd of that set be “smaller” than the given jd. To avoid misunderstandings, we now define the notion of decomposition of a jd in a more formal way.

Definition 2.10. Let R be a relation scheme. Let J be a jd over R and $\mathcal{J} = \{J_1, \dots, J_s\}$ be a set of jds over R . Then we say that $\{J_1, \dots, J_s\}$ is a decomposition of J if

- $J \Leftrightarrow \{J_1, \dots, J_s\}$;
- each J_i has strictly fewer edges than J ; and
- $s > 1$.

We say that J is decomposable if there exists a decomposition of J .

Note that we do not exclude embedded jds in the above definition. Furthermore, the third condition follows from the second one if we assume J to be reduced. We now illustrate Definitions 2.9 and 2.10 with an example.

Example 2.2. It can be verified that

$$\{abcde \bowtie df \bowtie eg \bowtie fg, abcd \bowtie defg, ab \bowtie ac \bowtie cd \bowtie bdefg\}$$

is a decomposition of

$$ab \bowtie ac \bowtie bd \bowtie cd \bowtie de \bowtie df \bowtie eg \bowtie fg$$

as was claimed in the Introduction. It can also be checked that

$$ab \bowtie bc \bowtie cd \bowtie de \bowtie ef \bowtie fg \bowtie ga$$

is indeed undecomposable (see Example 4.1). As is shown later on (Examples 5.1 and 5.5), both jds above are cyclic.

Consider now the jd:

$$ab \bowtie bc \bowtie cd \bowtie de \bowtie ef \bowtie fg.$$

A decomposition of this jd is

$$\{ab \bowtie bcdefg, abc \bowtie cdefg, abcd \bowtie defg, abcde \bowtie efg, abcdef \bowtie fg\}.$$

Hence this jd is acyclic.

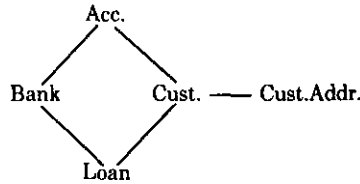
In the following section we will see how decompositions of jds can be generated in a constructive way. We conclude this section with the following observation.

THEOREM 2.1. *Suppose that $\{J_1, \dots, J_s\}$ is a decomposition of the jd J and that $\{J_{11}, \dots, J_{1s_1}\}$ is a decomposition of J_1 . Then $\{J_{11}, \dots, J_{1s_1}, J_2, \dots, J_s\}$ is a decomposition of J .*

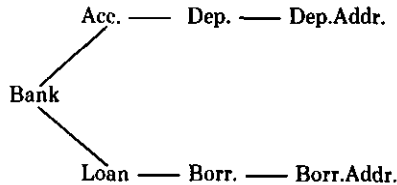
3. HINGES AND DECOMPOSITIONS OF A JD

Join dependencies are very important in relational database theory. In [9] it is conjectured that a real-world database can be described using one full jd and a

number of functional dependencies. In this paper we only consider the *jd*. The *jd* can be used to check the integrity of a current state of the database or to decompose the database into smaller relations. In the latter case, the consistency of the database must be verified each time the database is updated. Unfortunately, both problems are NP-complete in general. Therefore, it is desirable to have acyclic *jds* [2, 3, 8, 9]. Since acyclic *jds* are equivalent to a set of *mvds* (Definition 2.9), both integrity checking (in the case that the database is not decomposed) as well as consistency checking (in the other case) can be done in polynomial time. Initial design however often produces cyclic *jds*, as in the following example [9]:



Its cyclicity stems from the existence of the “cycle” *Bank—Loan—Customer—Account*. (This example suggests that cyclicity is a “local” property, i.e., that it is possible to distinguish cyclic parts in a *jd* [15].) Cyclicity of a *jd* often implies that some attributes are “overloaded” (i.e., that they have too many functions in the database scheme). In the previous example the attribute *Customer* has the meaning of both a borrower and a depositor. A way of dealing with the cyclicity of the *jd* in the previous example is to split the attribute *Customer* into two new attributes, *Borrower* and *Depositor*, which produces the following acyclic *jd*:



Attribute-splitting, however, will not always be desirable, due to the close relationship between the new attributes. In some cases it can be more interesting to keep the original (cyclic) *jd*. The fact that the desirable properties of acyclic *jds* are due to the equivalence of these *jds* to a set of *mvds*, suggests looking in the case of cyclic *jds* for more general decompositions, in order to “separate” the various parts in the *jd* that are “really” cyclic.

It is therefore important to have a methodology to decompose a given *jd* as far as possible into a set of *jds* with fewer edges. Such an algorithm was introduced in [10]. In this section we give a brief description of this decomposition methodology. The crucial notion in this algorithm is that of a hinge in a hypergraph.

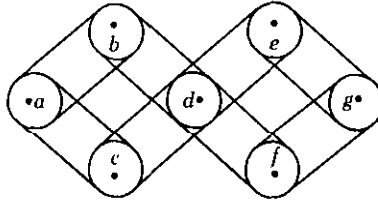
A *hinge* is a set of edges of a hypergraph that satisfies certain properties. Informally, for a set of edges to be a hinge, all the connected components with respect to that set must intersect that set within one of its edges. For instance, $\{ab, ac, bd, cd, de\}$ is a hinge of the hypergraph in Example 2.1. Hinges turn out to be very important in the theory of the decomposition of *jds*. The presence of a hinge in the hypergraph representation of a *jd*, for example, is a necessary and sufficient condition for that *jd* to be decomposable [10, 12, 13]. In [11] it is shown

that for a special class of jds, defined using the notion of hinge, there exists only one decomposition that can be characterized entirely in terms of the hinge structure of the hypergraph associated with the original jd. And in [12] it is shown that a slight generalization of the notion of hinge provides a very elegant characterization for an embedded jd to be implied by a full jd. We are now going to define a hinge more formally.

Definition 3.1. Let $\mathcal{H}(\mathcal{N}, \mathcal{E})$ be a reduced connected hypergraph. Let $\mathcal{E}' \subsetneq \mathcal{E}$, $|\mathcal{E}'| > 1$ and let $\mathcal{E}_1, \dots, \mathcal{E}_p$ be the connected components of \mathcal{H} with respect to \mathcal{E}' . \mathcal{E}' is a hinge of \mathcal{H} if for each \mathcal{E}_i , $i = 1, \dots, p$ there exists $E_i \in \mathcal{E}'$ such that $(\cup \mathcal{E}_i) \cap (\cup \mathcal{E}') \subseteq E_i$. E_i is called a separating edge corresponding to \mathcal{E}_i .

A hinge that is not contained within another hinge is called a maximal hinge. A hinge that contains no other hinges is called a minimal hinge.

Example 3.1. Consider again the jd represented by the following hypergraph $\mathcal{H}(\mathcal{N}, \mathcal{E})$:



The hinges of this hypergraph are $\{de, df, eg, fg\}$, $\{cd, de, df, eg, fg\}$, $\{bd, de, df, eg, fg\}$, $\{ab, ac, bd, cd\}$, $\{ab, ac, bd, cd, de\}$, $\{ab, ac, bd, cd, df\}$, $\{cd, de\}$, $\{cd, df\}$, $\{bd, de\}$, and $\{bd, df\}$. The second, the third, the fifth, and the sixth are maximal hinges; the others are minimal hinges.

In [13] it is shown that there exists a polynomial time algorithm to detect a hinge in a hypergraph. This observation turns out to be very important in the sequel. We now show that every hinge of a hypergraph generates a decomposition of the jd associated with that hypergraph.

THEOREM 3.1 [10]. Let $R(\Omega)$ be a relation scheme and let $\mathcal{H}(\mathcal{N}, \mathcal{E})$ be a reduced connected hypergraph with $\mathcal{N} \subseteq \Omega$ and let \mathcal{E}' be a hinge of \mathcal{H} . Let $\mathcal{E}_1, \dots, \mathcal{E}_p$ be the connected components of \mathcal{H} with respect to \mathcal{E}' and let E_1, \dots, E_p be corresponding separating edges. Let $\mathcal{N}_i = \cup \mathcal{E}_i$. We define the following hypergraphs:

$$\mathcal{H}_0\left(\mathcal{N}, (\mathcal{E}' \setminus \{E_i; i = 1, \dots, p\}) \cup \left\{E_i \cup \left(\bigcup_{E_j \in \mathcal{E}_i} \mathcal{N}_j\right); i = 1, \dots, p\right\}\right)$$

$$\mathcal{H}_i(\mathcal{N}, \mathcal{E}_i \cup \{\cup (\mathcal{E} \setminus \mathcal{E}_i)\}) \quad \text{for } i = 1, \dots, p.$$

Then $\{J_{\mathcal{H}_0}, J_{\mathcal{H}_1}, \dots, J_{\mathcal{H}_p}\}$ is a decomposition of $J_{\mathcal{H}}$.

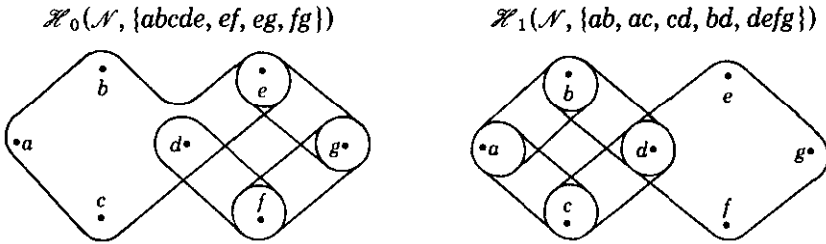
Informally, \mathcal{H}_0 is obtained from \mathcal{H} by considering every connected component of \mathcal{H} with respect to \mathcal{E}' , together with its separating edge, as one edge. \mathcal{H}_i , $1 \leq i \leq p$, is obtained by considering all the edges not in \mathcal{E}_i together as one edge. So in \mathcal{H}_0 only the structure of the hinge is preserved, whereas in \mathcal{H}_i only the structure of \mathcal{E}_i is preserved.

The Hinge Decomposition Algorithm. We propose to decompose a join-dependency $J_{\mathcal{X}}$ (using the notation introduced above):

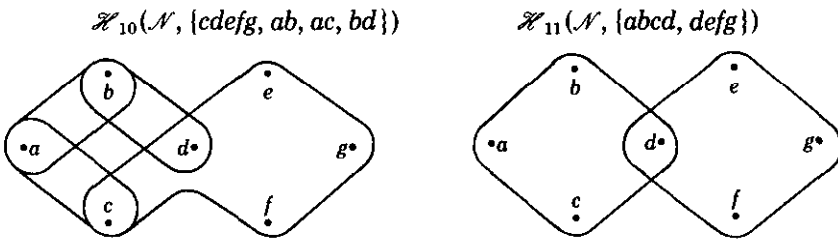
- (1°) Search for a hinge in \mathcal{X} . If there is more than one hinge, choose one (arbitrary for the time being). If there is no hinge, we cannot decompose $J_{\mathcal{X}}$ by the hinge decomposition algorithm.
- (2°) Apply Theorem 3.1 and obtain $J_{\mathcal{X}_0}, \dots, J_{\mathcal{X}_p}$.
- (3°) Apply the algorithm recursively on $J_{\mathcal{X}_0}, \dots, J_{\mathcal{X}_p}$.
Finally, use Theorem 2.1 to obtain the desired result.

Note that whenever the jd we started with is full, the decomposition we obtain also consists of full jds only.

Example 3.2. Consider the hypergraph $\mathcal{H}(\mathcal{N}, \mathcal{E})$ of Example 3.1. From Example 3.1 we know that $\{de, df, eg, fg\}$ is a hinge. There is only one connected component with respect to $\{de, df, eg, fg\}$, namely, $\{ab, ac, bd, cd\}$. de is a separating edge corresponding to it (df is another one). The second step of the hinge decomposition algorithm gives us the following hypergraphs:



Theorem 3.1 asserts that $\{J_{\mathcal{X}_0}, J_{\mathcal{X}_1}\}$ is a decomposition of $J_{\mathcal{X}}$. Since \mathcal{H}_0 does not contain any hinges, $J_{\mathcal{X}_0}$ cannot be decomposed any further with the hinge decomposition algorithm. \mathcal{H}_1 , however, contains a hinge, namely, $\{ab, ac, bd, cd\}$ with cd as a separating edge corresponding to the only connected component $\{defg\}$ with respect to this hinge. This gives rise to the hypergraphs:



Both \mathcal{H}_{10} and \mathcal{H}_{11} contain no hinges, so $J_{\mathcal{X}_{10}}$ and $J_{\mathcal{X}_{11}}$ are not decomposable any further by the hinge decomposition algorithm. $\{J_{\mathcal{X}_{10}}, J_{\mathcal{X}_{11}}\}$ is a decomposition of $J_{\mathcal{X}_1}$. Finally, Theorem 2.1 states that $\{J_{\mathcal{X}_0}, J_{\mathcal{X}_{10}}, J_{\mathcal{X}_{11}}\}$ is a decomposition of $J_{\mathcal{X}}$.

In the following section we give an overview of some properties of the hinge decomposition algorithm, most of which were proved in earlier work.

4. PROPERTIES OF THE HINGE DECOMPOSITION ALGORITHM

In this section we give an overview of some properties of the hinge decomposition algorithm, most of which were shown in earlier work. Some of these properties

are included to illustrate the strength of the hinge decomposition algorithm, whereas others will be needed in Sections 5 and 6, which contain the main results of this paper.

The first result we want to mention characterizes the presence of a hinge and, at the same time, demonstrates the importance of this notion:

THEOREM 4.1 [10, 12]. *Let R be a relation scheme and J be a reduced and connected jd. Then the following properties are equivalent:*

- (1°) J is decomposable;
- (2°) \mathcal{H}_J contains a hinge;
- (3°) There exists $E \in \mathcal{E}_J$ such that $\mathcal{E}_J \setminus \{E\}$ is not a connected subset with respect to $\{E\}$.

Example 4.1. Clearly, a jd that contains hinges is decomposable (by means of the hinge decomposition algorithm). Theorem 4.1 also claims the converse. This allows to check for decomposability by looking at the hinges. Consider the jd:

$$ab \bowtie bc \bowtie cd \bowtie de \bowtie ef \bowtie fg \bowtie ga.$$

It can easily be seen that this jd does not contain a hinge, either by a straightforward verification or by using the implication (2°) \Rightarrow (3°) of the previous theorem. Hence the above jd is undecomposable, which proves what was already claimed in the Introduction and in Example 2.2

In order to phrase some important properties of decompositions generated by the hinge decomposition algorithm, we need to introduce some additional terminology concerning decompositions.

Definition 4.1. Let J be a jd and suppose that $\{J_1, \dots, J_s\}$ is a decomposition of J . This decomposition is said to be final if none of the J_i , $i = 1, \dots, s$ is decomposable. This decomposition is said to be nonredundant if for each $i = 1, \dots, s$, $\{J_1, \dots, J_{i-1}, J_{i+1}, \dots, J_s\} \not\models J_i$.

Clearly, getting a final and nonredundant decomposition is an important property of a decomposition methodology. We have the following:

THEOREM 4.2 [10]. *Every decomposable reduced and connected jd can be decomposed with the hinge decomposition algorithm into a final and nonredundant decomposition.*

Example 4.2. Consider again the jds and hypergraphs of Example 3.2. We know already that $\{J_{\mathcal{X}_0}, J_{\mathcal{X}_{10}}, J_{\mathcal{X}_{11}}\}$ is a decomposition of $J_{\mathcal{X}}$. By Theorem 4.2, this decomposition is also final and nonredundant. The first claim already follows from Theorem 4.1; the second one implies that none of the three jds in the decomposition can be omitted.

Furthermore, it is possible to generate decompositions with the hinge decomposition algorithm in a time that is polynomial in the size of the input:

THEOREM 4.3 [13]. *Let $R(\Omega)$ be a relation scheme with $|\Omega| = m$ and let J be a jd over R with k edges. Let $n = km$. Then it is possible to generate a final decomposition of J with the hinge decomposition algorithm in $O(n^3)$ time.*

Furthermore, a final decomposition obtained with the hinge decomposition algorithm contains at most $k - 1$ jds.

PROOF. The proof of the first claim in Theorem 4.3 uses characterization (3°) of Theorem 4.1 for the presence of hinge. Since the proof is too long to include in this paper, we refer to [13] for further details. We now show the second part of Theorem 4.3. Let $\{J_1, \dots, J_s\}$ be a decomposition (not necessarily final) obtained after a number of steps of the hinge decomposition algorithm. Let σ be the sum of the numbers of edges of J_1, \dots, J_s . Then it can be easily shown by induction on the number of steps needed to obtain the decomposition that

$$\sigma = k + s - 1. \quad (1)$$

Since each jd in the decomposition contains at least two edges, we have that

$$\sigma \geq 2s. \quad (2)$$

Substitution of (2) in (1) yields

$$s \leq k - 1.$$

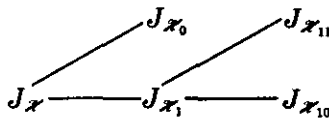
In particular, this equation also holds for a final decomposition obtained with the hinge decomposition algorithm. \square

It follows from Theorem 4.3 that an acyclic jd with k edges is equivalent to a set of $k - 1$ mvds, since the inequality (2) becomes an equality in that case.

Since hinges play a crucial role in the hinge decomposition algorithm, we may expect that there exists a close relationship between the jds of a decomposition obtained with the hinge decomposition algorithm and the "hinge-structure" of the jd that was decomposed. This relationship is expressed in Theorem 4.4, below. In order to prove this theorem, we need some lemmas. The first lemma was proved in [10].

LEMMA 4.1. *Every final decomposition of a jd obtained with the hinge decomposition algorithm can also be obtained by using only maximal hinges. In this way we obtain in each step of the decomposition process exactly two jds of which at least one is decomposable.*

Example 4.3. Consider again the decomposition obtained in Example 3.2. We show that this decomposition can also be obtained by using only maximal hinges. We know from Example 3.1 that $\{ab, ac, bd, cd, de\}$ is a maximal hinge of \mathcal{H} . An application of Theorem 3.1 with this hinge gives us also the hypergraphs \mathcal{H}_0 and \mathcal{H}_1 , but in the opposite order. Furthermore, $\{ab, ac, bd, cd\}$ is a maximal hinge of \mathcal{H}_1 . We often represent a decomposition process using only maximal hinges by a binary tree. The decomposition process in this example is represented by



LEMMA 4.2. *Let R be a relation scheme and let J be a jd over R . Let \mathcal{E}' be a maximal hinge of \mathcal{H}_J . Then $\mathcal{E}_J \setminus \mathcal{E}'$ is connected. Let E be a separating edge of \mathcal{E}'*

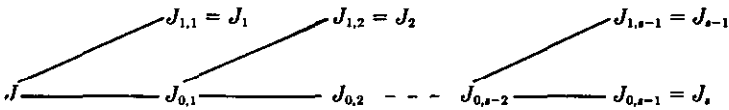
corresponding to \mathcal{E}_J . Then $(\mathcal{E}_J \setminus \mathcal{E}') \cup \{E\}$ is also a hinge of \mathcal{H}_J and E is a separating edge of this hinge corresponding to every connected component with respect to that hinge. Finally, if J_0 and J_1 are the jds in the decomposition defined by \mathcal{E}' (according to Theorem 3.1), then J_1 and J_0 can be obtained from J by replacing the edges of \mathcal{E}' respectively, $(\mathcal{E}_J \setminus \mathcal{E}') \cup \{E\}$ by their union.

PROOF. A straightforward verification shows that $\mathcal{E}_J \setminus \mathcal{E}'$ is connected. (Otherwise, \mathcal{E}' would not be a maximal hinge.) We show that $(\mathcal{E}_J \setminus \mathcal{E}') \cup \{E\}$ is a hinge of \mathcal{E}_J of which E is a separating edge corresponding to every connected component with respect to it. This follows from the fact that each edge not in $(\mathcal{E}_J \setminus \mathcal{E}') \cup \{E\}$ necessarily belongs to \mathcal{E}' and that E is a separating edge of \mathcal{E}' corresponding to $\mathcal{E}_J \setminus \mathcal{E}'$. The remaining part of the lemma follows from a simple look at the hypergraphs in Theorem 3.1. \square

Example 4.4. In Example 4.3 we pointed out that $J_{\mathcal{H}_1}$ and $J_{\mathcal{H}_0}$ can be obtained from $J_{\mathcal{H}}$ by a decomposition using the maximal hinge $\{ab, ac, bd, cd, de\}$. Clearly, de is the only separating edge of this hinge corresponding to $\{df, eg, fg\}$, which is connected to $\{ab, ac, bd, cd, de\}$. A straightforward verification shows that $\{df, eg, fg, de\}$ is also a hinge and that de is a separating edge of this hinge corresponding to the (only) connected component $\{ab, ac, bd, cd\}$ with respect to it. Clearly, $J_{\mathcal{H}_0}$ is obtained from $J_{\mathcal{H}}$ by replacing $\{ab, ac, bd, cd, de\}$ by its union, whereas $J_{\mathcal{H}_1}$ is obtained from $J_{\mathcal{H}}$ by replacing $\{df, eg, fg, de\}$ by its union.

LEMMA 4.3. Let R be a relation scheme and $J: X_1 \bowtie \dots \bowtie X_k$ a reduced and connected jd over R . Let $\{J_1, \dots, J_s\}$ be a final decomposition of J obtained by using hinges. Let, for $i = 1, \dots, s$, J_i be the jd $Y_{i1} \bowtie \dots \bowtie Y_{ik_i}$. Then, for all $i = 1, \dots, s$ and for all $j = 1, \dots, k_i$, there exists $1 \leq t_{ij} \leq k$ with $X_{t_{ij}} \subseteq Y_{ij}$ and $Y_{ij} \cap (\cup \{Y_{il}; 1 \leq l \leq k_i \text{ and } l \neq j\}) \subseteq X_{t_{ij}}$.

PROOF. We can assume that the above decomposition is obtain by using maximal hinges only (Lemma 4.1). Let



be a tree representing the decomposition of J into $\{J_1, \dots, J_s\}$ as in Example 4.3. We prove by induction on their depth in the tree that all the jds in that tree satisfy the property of Lemma 4.3.

On depth 0, we have only J itself, and, for J , Lemma 4.3 is trivially satisfied. Suppose now that all jds in the above tree until depth m satisfy the property of Lemma 4.3. Let $J': Y_1 \bowtie \dots \bowtie Y_p$ be a jd in this tree on depth $m+1$. For notational convenience, we prove that there exists $1 \leq t \leq p$ with $X_t \subseteq Y_1$ and $Y_1 \cap (Y_2 \cup \dots \cup Y_p) \subseteq X_t$. Let $J'': Y'_1 \bowtie \dots \bowtie Y'_{p'}$ ($p' > p$) be the father of J' . We distinguish two possible cases:

Case 1. Y_1 is an edge of J'' . Without loss of generality we can assume that $Y_1 = Y'_1$. By induction there exists $1 \leq t \leq s$ with $X_t \subseteq Y'_1$ and $Y'_1 \cap (Y'_2 \cup \dots \cup Y'_{p'}) \subseteq X_t$. Clearly, $X_t \subseteq Y_1$, and

$$Y_1 \cap (Y_2 \cup \dots \cup Y_p) = Y'_1 \cap (Y'_2 \cup \dots \cup Y'_{p'}) \subseteq X_t.$$

Case 2. Y_1 is not an edge of J'' . Then, according to Lemma 4.2, there exists a hinge of $\mathcal{H}_{J''}$ such that Y_1 is the union of that hinge. Without loss of generality, let $\{Y'_1, \dots, Y'_q\}$, $q < p'$ be that hinge. Also, this hinge contains a separating edge corresponding to all the connected components with respect to that hinge. We can assume that this separating edge is Y'_1 . Now let X_t be as in Case 1. Then

$$\begin{aligned} Y_1 \cap (Y_2 \cup \dots \cup Y_p) &= (Y'_1 \cup \dots \cup Y'_q) \cap (Y'_{t+1} \cup \dots \cup Y'_{p'}) \\ &\subseteq Y'_1 \cap (Y'_{t+1} \cup \dots \cup Y'_{p'}) \\ &\subseteq Y'_1 \cap (Y'_2 \cup \dots \cup Y'_{p'}) \\ &\subseteq X_t. \end{aligned}$$

This completes the proof. \square

Example 4.5. We know from Example 3.2 that

$$\begin{aligned} J_{\mathcal{H}_0}: abcde \bowtie df \bowtie eg \bowtie fg \\ J_{\mathcal{H}_{10}}: ab \bowtie ac \bowtie bd \bowtie cdefg \\ J_{\mathcal{H}_{11}}: abcd \bowtie defg \end{aligned}$$

is a decomposition of

$$J_{\mathcal{H}}: ab \bowtie ac \bowtie bd \bowtie cd \bowtie de \bowtie df \bowtie eg \bowtie fg.$$

The edges of $J_{\mathcal{H}}$ satisfying the conditions of Lemma 4.3 are, as is easily verified, (in the same order as above)

$$\begin{aligned} \text{for } J_{\mathcal{H}_0}: & de, df, eg \quad \text{and} \quad fg; \\ \text{for } J_{\mathcal{H}_{10}}: & ab, ac, bd \quad \text{and} \quad cd; \\ \text{for } J_{\mathcal{H}_{11}}: & bd \text{ (or } cd) \quad \text{and} \quad de \text{ (or } df). \end{aligned}$$

We can now prove the actual theorem.

THEOREM 4.4. *Let R be a relation scheme and $J: X_1 \bowtie \dots \bowtie X_k$ be a reduced and connected jd over R . Let $\{J_1, \dots, J_s\}$ be a final decomposition of J obtained with the hinge decomposition algorithm. Suppose, for $i = 1, \dots, s$, that J_i is the jd $Y_{i1} \bowtie \dots \bowtie Y_{ik_i}$. Then there exist $1 \leq t_{i1}, \dots, t_{ik_i} \leq k$ such that*

- for all $j = 1, \dots, k_i$, $X_{t_{ij}} \subseteq Y_{ij}$ (and hence all the $X_{t_{ij}}$ are different);
- $\{X_{t_{i1}}, \dots, X_{t_{ik_i}}\}$ is a minimal hinge of \mathcal{H}_{J_i} ;
- for all $j = 1, \dots, k_i$, Y_{ij} is the union of $X_{t_{ij}}$ and all the edges of some connected components \mathcal{H}_J with respect to $\{X_{t_{i1}}, \dots, X_{t_{ik_i}}\}$ for which $X_{t_{ij}}$ is a corresponding separating edge.

PROOF. Choose $X_{t_{i1}}, \dots, X_{t_{ik_i}}$ as in Lemma 4.3. Then, for $1 \leq l \neq m \leq k_i$, we have that $Y_{il} \cap Y_{im} \subseteq X_{t_{il}} \cap X_{t_{im}}$. Hence Y_{il} and Y_{im} are disconnected with respect to $\{X_{t_{i1}}, \dots, X_{t_{ik_i}}\}$. Furthermore, we have that

$$\begin{aligned} Y_{i1} \cap (X_{t_{i1}} \cup \dots \cup X_{t_{ik_i}}) &\subseteq Y_{i1} \cap (X_{t_{i1}} \cup Y_{i2} \cup \dots \cup Y_{ik_i}) \\ &\subseteq X_{t_{i1}} \cup (Y_{i1} \cap (Y_{i2} \cup \dots \cup Y_{ik_i})) \\ &\subseteq X_{t_{i1}} \end{aligned}$$

and *mutatis mutandis* for the other indices,

$$Y_{ij} \cap (X_{t_{i1}} \cup \dots \cup X_{t_{ik_i}}) \subseteq X_{t_{ij}}.$$

Hence $\{X_{t_{i1}}, \dots, X_{t_{ik}}\}$ is a hinge satisfying the first and third conditions of Theorem 4.4. It remains to show that this hinge is minimal. Suppose this would not be the case. Then, after a possible renaming of indices, let $\{X_{t_{i1}}, \dots, X_{t_{ip}}\}$, $p < k_i$ be a maximal hinge of $\{X_{t_{i1}}, \dots, X_{t_{ik}}\}$ with $X_{t_{i1}}$ as separating edge corresponding to $\{X_{t_{i(p+1)}}, \dots, X_{t_{ik}}\}$. Then

$$\begin{aligned}
 (Y_{i1} \cup \dots \cup Y_{ip}) \cap (Y_{i(p+1)} \cup \dots \cup Y_{ik_i}) &= \cup \{Y_{il} \cap Y_{im}; 1 \leq l \leq p, p+1 \leq m \leq k_i\} \\
 &\subseteq \cup \{X_{t_{il}} \cap X_{t_{im}}; 1 \leq l \leq p, p+1 \leq m \leq k_i\} \\
 &= (X_{t_{i1}} \cup \dots \cup X_{t_{ip}}) \cap (X_{t_{i(p+1)}} \cup \dots \cup X_{t_{ik_i}}) \\
 &\subseteq X_{t_{i1}} \\
 &\subseteq Y_{i1}.
 \end{aligned}$$

Consequently, $\{Y_{i1}, \dots, Y_{ip}\}$ is a hinge of \mathcal{H}_{J_i} , in contradiction with our decomposition being final (cf. Theorem 4.2). So $\{X_{t_{i1}}, \dots, X_{t_{ik}}\}$ is indeed a minimal hinge of \mathcal{H}_J . \square

We conclude this section with an illustration of the previous theorem.

Example 4.6. Consider again the hypergraph \mathcal{H} of Example 3.1 and the decomposition of $J_{\mathcal{H}}$ obtained in Example 3.2. Minimal hinges of \mathcal{H} satisfying Theorem 4.4 for the various jds of this decomposition are

- for \mathcal{H}_0 : $\{de, df, eg, fg\}$;
- for \mathcal{H}_{10} : $\{ab, ac, bd, cd\}$;
- for \mathcal{H}_{11} : $\{cd, de\}$.

For the last jd, we could also have taken $\{cd, df\}$, $\{bd, de\}$ or $\{bd, df\}$.

5. THE COMPLEXITY OF INTEGRITY CHECKING

An important reason for replacing a join dependency by a decomposition is making integrity checking less time consuming. Indeed, performing a join requires a time exponential in the number of edges that are to be joined. It therefore is important to minimize the size of the largest jd to be taken into account. This leads us to the definition of the *degree of cyclicity* of a jd, which is stated in Definition 5.2. (A very similar definition has also been stated independently in [18].) We then characterize the degree of cyclicity of a jd in terms of the jd itself and show that our decomposition methodology generates decompositions that satisfy the above minimality criterion. Finally, we define *n-cyclic* jds as jds whose degree of cyclicity is at most n and we show that for a fixed value of n , integrity checking can be performed in polynomial time, provided only n -cyclic jds are considered.

We first establish some notations.

Definition 5.1. Let R be a relation scheme and J be a reduced and connected jd over R . Then

- $\text{DEC}(J)$ is the set of all decompositions of J augmented with J itself;
- $\text{MINHIN}(J)$ is the set of all minimal hinges of \mathcal{H}_J ; and
- $|J|$ is the number of edges of J .

Definition 5.2. Let R be a relation scheme and J be a reduced and connected jd over R . We define the degree of cyclicity of J , denoted $\text{dgr}(J)$, as

$$\text{dgr}(J) = \min\{\max\{|J'|, J' \in \mathcal{J}\}, \mathcal{J} \in \text{DEC}(J)\}.$$

In words, the degree of cyclicity of a jd is the minimum over $\text{DEC}(J)$ of the number of edges of the largest jd in an element of $\text{DEC}(J)$. From the definition of “decomposition of a jd” (Definition 2.10), it immediately follows:

LEMMA 5.1. Let R be a relation scheme and J a jd over R . Then $\text{dgr}(J) \leq |J|$. Furthermore, $\text{dgr}(J) = |J|$ if and only if J is undecomposable.

Observe that a (nontrivial) jd is acyclic if and only if its degree of cyclicity equals 2. Indeed, a jd is acyclic if and only if it is equivalent with a set of multivalued dependencies. Obviously, it is impossible to find a “better” decomposition.

Example 5.1. Consider the jd $ab \bowtie bc \bowtie cd \bowtie de \bowtie ef \bowtie fg \bowtie ga$. In Example 4.1 we proved that this jd is undecomposable. Hence its degree of cyclicity equals 7. In particular, this jd is cyclic.

Now consider again the jd $J_{\mathcal{H}}$ of Example 3.1. Using one of the many characterizations for acyclicity (see, e.g., [2, 3, 8, 9]), it can be shown that $J_{\mathcal{H}}$ is cyclic (we prove this in Example 5.5). Taking into account the above observation, it follows that the degree of cyclicity of $J_{\mathcal{H}}$ is at least 3. Since the largest jd in the decomposition of $J_{\mathcal{H}}$ constructed in Example 3.2 has size 4, it follows that $\text{dgr}(J_{\mathcal{H}})$ is at most 4. Later on, we shall see how we can calculate the precise degree of cyclicity of a jd.

Although Definition 5.2 seems a very natural way to define the complexity of a jd, it does not provide an efficient way to calculate the degree of cyclicity of a jd, as was pointed out in the previous example. Therefore, we are now going to show that the degree of cyclicity of a decomposable jd equals the size of the largest minimal hinge of its associated hypergraph. Hence we characterize the notion “degree of cyclicity of a jd” in terms of the jd itself.

We first prove that the degree of cyclicity of a decomposable jd is at least the size of the largest minimal hinge of its associated hypergraph. We therefore need some lemmas.

We first show that equivalence of jds is “inherited” if we “restrict” each jd to a subset of the set of attributes that is involved. We therefore define the “trace” of a jd.

Definition 5.3. Let $R(\Omega)$ be a relation scheme and let $J: X_1 \bowtie \dots \bowtie X_k$ be a jd over R and let $X \subseteq \Omega$. The trace of J with respect to X , denoted $\text{tr}_X(J)$, is the reduction of $X_1 \cap X \bowtie \dots \bowtie X_k \cap X$.

Example 5.2. Consider again the jd $J_{\mathcal{H}}$ of Example 3.1. Let $X = abcde$. Then $\text{tr}_X(J_{\mathcal{H}}) = ab \bowtie ac \bowtie bd \bowtie cd \bowtie de$.

LEMMA 5.2. Let R be a relation scheme and let J be a reduced and connected jd over R . Suppose that $\mathcal{E}' = \{E_1, \dots, E_l\}$ is a hinge of \mathcal{H}_J . Then $\text{tr}_{\cup \mathcal{E}'}(J) = E_1 \bowtie \dots \bowtie E_l$.

PROOF. Let E be an arbitrary edge of J not in \mathcal{E}' . Since, by definition, there exists $1 \leq j \leq l$ such that $E \cap (\cup \mathcal{E}') \subseteq E_j$, Lemma 5.2 follows.

Example 5.3. Consider again the jd $J_{\mathcal{H}}$ of Example 3.1. In that example we saw that $\{ab, ac, bd, cd, de\}$ is a hinge \mathcal{H} . From Example 5.2 it follows that $\text{tr}_{abde}(J_{\mathcal{H}}) = ab \bowtie ac \bowtie bd \bowtie cd \bowtie de$, in accordance with Lemma 5.2. Note that the condition in Lemma 5.2 that \mathcal{E}' is a hinge cannot be arbitrarily removed (though it can be weakened). Indeed, consider the set $\{ab, ac, cd, de\}$. This set is not a hinge of \mathcal{H} , but has the same set of nodes as $\{ab, ac, bd, cd, de\}$. Hence $\text{tr}_{abde}(J_{\mathcal{H}}) = ab \bowtie ac \bowtie bd \bowtie cd \bowtie de \neq ab \bowtie ac \bowtie cd \bowtie de$.

LEMMA 5.3. Let $R(\Omega)$ be a relation scheme and let J, J_1, \dots, J_s be jds over R and suppose $J \Leftarrow (J_1, \dots, J_s)$. Let $X \subseteq \Omega$. Then $\text{tr}_X(J) \Leftarrow \{\text{tr}_X(J_1), \dots, \text{tr}_X(J_s)\}$.

PROOF. The proof uses the chase technique [14]. Consider the implication " \Leftarrow ". Consider tableaux for J and for $\text{tr}_X(J)$. Each time a jd J_i , $1 \leq i \leq s$ is applied to the tableau for J , we can apply $\text{tr}_X(J_i)$ in a similar way to the tableau for $\text{tr}_X(J)$. Since the chase for $\{J_1, \dots, J_s\} \Rightarrow J$ is successful, it follows that the constructed chase for $\{\text{tr}_X(J_1), \dots, \text{tr}_X(J_s)\} \Rightarrow \text{tr}_X(J)$ is successful too. The implication " \Rightarrow " can be shown in the same way. \square

Example 5.4. Consider again the jd $J_{\mathcal{H}}$ of Example 3.1 and its decomposition $\{J_{\mathcal{H}_0}, J_{\mathcal{H}_{10}}, J_{\mathcal{H}_{11}}\}$ constructed in Example 3.2. Let $X = abdeg$. Then

$$\begin{aligned} \text{tr}_X(J_{\mathcal{H}}) &= ab \bowtie bd \bowtie de \bowtie eg; \\ \text{tr}_X(J_{\mathcal{H}_0}) &= abde \bowtie eg; \\ \text{tr}_X(J_{\mathcal{H}_{10}}) &= ab \bowtie bd \bowtie deg; \\ \text{tr}_X(J_{\mathcal{H}_{11}}) &= abd \bowtie deg. \end{aligned}$$

According to Lemma 5.3, we have that $ab \bowtie bd \bowtie de \bowtie eg \Leftarrow \{abd \bowtie de \bowtie eg, ab \bowtie bd \bowtie deg, abd \bowtie deg\}$. Note that the right-hand side of this equivalence is neither final nor nonredundant.

We can now prove

LEMMA 5.4. Let R be a relation scheme and let J, J_1, \dots, J_s be jds over R with J reduced and connected. Suppose $J \Leftarrow \{J_1, \dots, J_s\}$. Let $\mathcal{E}' = \{E_1, \dots, E_l\}$ be a minimal hinge of \mathcal{H}_J . Then there exists $1 \leq i \leq s$ with $|J_i| \geq |\mathcal{E}'|$.

PROOF. Let J' be $E_1 \bowtie \dots \bowtie E_l$. From Lemma 5.2 and Lemma 5.3 we know that $J' \Leftarrow \{\text{tr}_{\cup \mathcal{E}'}(J_1), \dots, \text{tr}_{\cup \mathcal{E}'}(J_s)\}$. Since $\mathcal{H}_{J'}$ does not contain hinges (because \mathcal{E}' is minimal), it follows from Theorem 4.1 that J' is undecomposable. Hence there exists $1 \leq i \leq s$ with $|\text{tr}_{\cup \mathcal{E}'}(J_i)| \geq |J'|$. Clearly, $|J'| = |\mathcal{E}'|$ and $|J_i| \geq |\text{tr}_{\cup \mathcal{E}'}(J_i)|$, from which the desired inequality follows. \square

As an immediate corollary of Lemma 5.4, we obtain

COROLLARY 5.1. Let R be a relation scheme and let J be a decomposable reduced and connected jd over R . Then $\text{dgr}(J) \geq \max\{|\mathcal{E}'|, \mathcal{E}' \in \text{MINHIN}(J)\}$.

Example 5.5. Consider again the jd $J_{\mathcal{H}}$ of Example 3.1. From Example 3.1 it follows that the size of the largest minimal hinge of $J_{\mathcal{H}}$ equals 4. Hence, by Corollary 5.1, $\text{dgr}(J) \geq 4$. In particular, this result proves that $J_{\mathcal{H}}$ is cyclic. Since we already know (Example 5.1) that $\text{dgr}(J) \leq 4$, it follows that $\text{dgr}(J) = 4$.

To prove that the inequality in Corollary 5.1 is actually an equality, it suffices to construct for a given decomposable jd a decomposition of which the largest element contains at most as many edges as there are edges in the largest minimal hinge of the hypergraph associated with the given jd. It turns out that the decompositions obtained with the hinge decomposition algorithm satisfy this criterion:

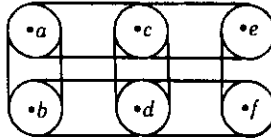
LEMMA 5.5. *Let R be a relation scheme and let J be a decomposable reduced and connected jd over R . Then $\text{dgr}(J) \leq \max\{|\mathcal{E}'|, \mathcal{E}' \in \text{MINHIN}(J)\}$.*

PROOF. According to Theorem 4.4, the number of edges of a final decomposition of a given decomposable jd obtained with the hinge decomposition algorithm equals the number of edges of some minimal hinge of the original jd. The statement of the lemma immediately follows from this observation. \square

If we combine Corollary 5.1, Lemma 5.5, and Theorem 4.3, we immediately get:

THEOREM 5.1. *Let R be a relation scheme and let J be a decomposable reduced and connected jd over R . Then $\text{dgr}(J) = \max\{|\mathcal{E}'|, \mathcal{E}' \in \text{MINHIN}(J)\}$. Furthermore, for each decomposition \mathcal{J} of J obtained by using hinges we have that $\text{dgr}(J) = \max\{|\mathcal{J}'|; \mathcal{J}' \in \mathcal{J}\}$. Hence each decomposition obtained by our method minimizes the number of edges of its largest component.*

Example 5.6. We already showed for the jd $J_{\mathcal{A}}$ of Example 3.1 that $\text{dgr}(J_{\mathcal{A}}) = 4$. We therefore look at another example. Consider the jd J' : $ab \bowtie ac \bowtie bd \bowtie cd \bowtie ce \bowtie df \bowtie ef$. This jd can be represented by the hypergraph:



The above hypergraph contains two hinges, $\{ab, ac, bd, cd\}$ and $\{cd, ce, df, ef\}$, which are both minimal and maximal. Hence $\text{dgr}(J') = 4$. Note that this result is obtained without looking at decompositions of J' .

We now use the degree of cyclicity as a tool to classify join dependencies according to their complexity.

Definition 5.4. Let R be a relation scheme and let J be a reduced and connected jd over R . Let n be a positive integer. J is said to be n -cyclic if its degree of cyclicity is at most n .

In this way we obtain a hierarchical classification of join dependencies. According to a previous remark, 2-cyclicity is the same as acyclicity. It is well known [2, 3, 8, 9] that integrity checking for acyclic jds can be performed in polynomial time. We now extend this result.

THEOREM 5.2. *Let $n > 1$ be a fixed integer. Let $R(\Omega)$ be a relation scheme and let J be an n -cyclic reduced and connected jd over R with k edges. Let m be the number of attributes in Ω . Then checking an instance r over R for integrity can be done in a time that is polynomial in m , k , and l , where $l = |r|$.*

PROOF. By Theorem 4.3, we can obtain a decomposition of J with the hinge decomposition algorithm on $O(k^3m^3)$ time. Also by Theorem 4.3, this decomposition consists of at most $k - 1$ jds. By Theorem 5.1, each of these jds has at most n edges. Hence checking whether such a jd holds can be done in $O(l^{n+1}m)$ time, as can be shown by straightforward calculation. This means that, using the decomposition, checking whether J holds can be done in $O(k^3m^3 + kml^{n+1})$ time. \square

Hence integrity checking can be done in polynomial time provided only n -cyclic jds are considered for a fixed value of n .

6. THE COMPLEXITY OF CONSISTENCY CHECKING

Instead of using a jd to check a relation instance for consistency after each update, we could also decompose the relation scheme according to the jd, and thus work with smaller relations. If this approach is followed—which is usually the case—the problem of integrity checking is eliminated, since the join of the various “subrelations” by definition satisfies the jd. Nevertheless, we have to make sure that all our subrelations are the projections of a common relation instance over the original relation scheme. In other words, we may not lose tuples by first making the join of all the subrelations and then projecting back onto the edges of the jd. So we have to check our decomposed database for *consistency* after each update of a subrelation.

A very important advantage of a jd describing the structure of a decomposed database being acyclic is that checking a database instance for consistency can be performed in polynomial time, because acyclicity is equivalent with “pair-wise consistency implies global consistency.” Pair-wise consistency means that every two subrelations are the projections of a common relation instance, and this property can easily be verified in polynomial time. In this section we generalize this result by proving that n -cyclicity is equivalent with “ n -wise consistency implies global consistency.” Thus checking for (global) consistency remains solvable in polynomial time if we restrict ourselves to n -cyclic jds for a fixed value of n .

We first recall some well-known basic facts about consistency and introduce the necessary notation. We then prove the two lemmas we need to show that “ n -wise consistency implies global consistency” is a sufficient condition for n -cyclicity. We then prove that it is also a necessary condition. Finally, we show that checking for (global) consistency can be performed in polynomial time, provided we restrict ourselves to n -cyclic jds for a fixed value of n .

Definition 6.1. Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$. Let (r_1, \dots, r_k) be an instance over (X_1, \dots, X_k) . Let n be a positive integer.

- If $n \leq k$, then (r_1, \dots, r_k) is called n -wise consistent if for each subsequence i_1, \dots, i_n of $1, \dots, k$ there exists an instance r over $X_{i_1} \cup \dots \cup X_{i_n}$ satisfying $\pi_{X_{i_t}}(r) = r_{i_t}$ for all $t = 1, \dots, n$.
- (r_1, \dots, r_k) is called globally consistent if it is k -wise consistent.
- If $n > k$, then (r_1, \dots, r_k) is called n -wise consistent if it is globally consistent.

We can now make some straightforward observations:

LEMMA 6.1. Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$. Let (r_1, \dots, r_k) be an instance over (X_1, \dots, X_k) . Let n be a positive integer. Let

$m \leq n$. If (r_1, \dots, r_k) is n -wise consistent, then it is also m -wise consistent. Suppose $n \leq k$. Then (r_1, \dots, r_k) is n -wise consistent if and only if for each subsequence i_1, \dots, i_n of $1, \dots, k$ and for all $t = 1, \dots, n$ we have that $\pi_{X_{i_t}}(r_{i_1} \bowtie \dots \bowtie r_{i_n}) = r_{i_t}$.

LEMMA 6.2. Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$, $k > 2$. Let $1 < l < k$. (r_1, \dots, r_k) is a globally consistent instance over (X_1, \dots, X_k) if and only if (r_1, \dots, r_l) is globally consistent over (X_1, \dots, X_l) and $(r_1 \bowtie \dots \bowtie r_l, r_{l+1}, \dots, r_k)$ is globally consistent over $(\cup_{i=1}^l X_i, X_{l+1}, \dots, X_k)$.

The last observation concerning Definition 6.1 we want to make here deals with 2-wise consistency, which is usually called *pair-wise consistency*. It can be easily seen that pair-wise consistency can be checked in the following way:

LEMMA 6.3. Let R be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$. Let (r_1, \dots, r_k) be an instance over (X_1, \dots, X_k) . (r_1, \dots, r_k) is pair-wise consistent if

$$\forall 1 \leq i \neq j \leq k: \pi_{X_i \cap X_j}(r_i) = \pi_{X_i \cap X_j}(r_j).$$

We illustrate the notion of consistency with the following example:

Example 6.1. Let $R(\Omega)$ be a relation scheme and let $\Omega = abc$. Let (r_1, r_2, r_3) be the following instance over (ab, bc, ac) :

$\frac{a}{1}$	$\frac{b}{1}$	$\frac{b}{1}$	$\frac{c}{3}$	$\frac{a}{4}$	$\frac{c}{3}$
4	1	5	2	1	2
1	5				

Clearly, the projections of r_1 and r_2 onto $b = ab \cap bc$ both equal

$$\frac{b}{1} \\ 5$$

The projections of r_2 and r_3 onto $c = bc \cap ac$ both equal

$$\frac{c}{3} \\ 2$$

and the projections of r_1 and r_3 onto $a = ab \cap ac$ both equal

$$\frac{a}{1} \\ 4$$

Hence, by Lemma 6.3, (r_1, r_2, r_3) is pair-wise consistent. Calculating $r_1 \bowtie r_2 \bowtie r_3$ gives

$\frac{a}{4}$	$\frac{b}{1}$	$\frac{c}{3}$
1	5	2

Clearly, $\pi_{ab}(r_1 \bowtie r_2 \bowtie r_3) \neq r_1$. Thus, by Lemma 6.1, (r_1, r_2, r_3) is not globally consistent.

We are now going to prove that a n -cyclicity is implied by “ n -wise consistency implies global consistency.” We therefore need two technical lemmas (Lemmas 6.5 and 6.6). In view of the assumptions we made in Section 2, we can assume without loss of generality that all the domains involved contain the set of all nonnegative integers and all pairs of nonnegative integers. We introduce the following notation, which is needed in Lemma 6.5.

Definition 6.2. Let $R(\Omega)$ be a relation scheme and let $X \subseteq \Omega$. Let $\{Y_1, \dots, Y_l\}$ be a partition of X and let i_1, \dots, i_l be nonnegative integers or pairs of nonnegative integers. Then $i_{1Y_1} \dots i_{lY_l}$ denotes the tuple over X that associates the value i_j with each attribute of Y_j for each $j = 1, \dots, l$.

Example 6.2. Let $\Omega = X = abcdefg$. Let $Y_1 = abc$, $Y_2 = de$, and $Y_3 = fg$. Then $0_{Y_1}2_{Y_2}(2, 3)_{Y_3}$ denotes the tuple

$$\begin{array}{ccccccc} a & b & c & d & e & f & g \\ \hline 0 & 0 & 0 & 2 & 2 & (2, 3) & (2, 3) \end{array}.$$

Using this notation we can easily calculate the projection of a tuple:

LEMMA 6.4. Let $R(\Omega)$ be a relation scheme and let $Z \subseteq X \subseteq \Omega$. Let $\{Y_1, \dots, Y_l\}$ be a partition of X and let i_1, \dots, i_l be nonnegative integers or pairs of nonnegative integers. Let

$$i_{1Y_1} \dots i_{lY_l}$$

be a tuple over X . Then the projection of this tuple onto Z is

$$i_{1Y_1 \cap Z} \dots i_{lY_l \cap Z}.$$

The proofs of both Lemma 6.5 and Lemma 6.6 are notationally rather involved. Therefore, we give here only the main constructions of these proofs. The reader can find further details in the Appendix. The constructions made in these lemmas are subsequently illustrated by examples (Examples 6.3 and 6.4). We advise the reader to consult these examples while examining the lemmas and their proofs.

LEMMA 6.5. Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$, $k > 2$. Let $X = \bigcup_{i=1}^k X_i$. Suppose $J : X_1 \bowtie \dots \bowtie X_k$ is a reduced and connected undecomposable jd without attributes occurring in only one edge. Define the instance (r_1, \dots, r_k) over (X_1, \dots, X_k) as follows:

$$\begin{aligned} r_1 &= \{0_{X_1}\} \cup \{0_{X_1 \cap X_j}(t, j)_{X_1 \setminus X_j}; 2 \leq j \neq t \leq k\}; \\ r_i &= \{0_{X_1 \cap X_i \cap X_j}(t, j)_{X_i \setminus X_j}; 2 \leq j \neq t \leq k\}, \quad i \neq 1. \end{aligned}$$

Then (r_1, r_k) is $k - 1$ -wise consistent, but not globally consistent.

PROOF. To show that (r_1, \dots, r_k) is $k - 1$ -wise consistent, it suffices to construct relation instances r_l over X , $l = 1, \dots, k$ satisfying $r_i = \pi_{X_i}(r^l)$ for $l \neq i$. Therefore, define

$$\begin{aligned} r^1 &= \{0_{X_1 \cap X_j}(t, j)_{X \setminus X_j}; 2 \leq j \neq t \leq k\}; \\ r^l &= \{0_{X_1 \setminus X_l}\} \cup r^1, \quad l \neq 1. \end{aligned}$$

A straightforward verification using Lemma 6.4 shows that for $1 \leq i \neq l \leq k$ we have it indeed that $\pi_{X_i}(r^l) = r_i$. Thus (r_1, \dots, r_k) is $k - 1$ -wise consistent. The

proof that (r_1, \dots, r_k) is not k -wise (globally) consistent can be found in the Appendix. \square

Example 6.3. Let $\Omega = X = abcd$. Let $X_1 = ab$, $X_2 = bc$, $X_3 = cd$, and $X_4 = da$. Then r_1, r_2, r_3 , and r_4 are, respectively,

a	b	b	c	
0	0	0	3	$(j = 2, t = 3)$
(3, 2)	0	0	4	$(j = 2, t = 4)$
(4, 2)	0	(2, 3)	2	$(j = 3, t = 2)$
(2, 3)	(2, 3)	(4, 3)	4	$(j = 3, t = 4)$
(4, 3)	(4, 3)	(2, 4)	(2, 4)	$(j = 4, t = 2)$
0	(2, 4)	(3, 4)	(3, 4)	$(j = 4, t = 3)$
0	(3, 4)			$(j = 4, t = 3)$

c	d	d	a	
3	(3, 2)	(3, 2)	(3, 2)	$(j = 2, t = 3)$
4	(4, 2)	(4, 2)	(4, 2)	$(j = 2, t = 4)$
2	2	2	(2, 3)	$(j = 3, t = 2)$
4	4	4	(4, 3)	$(j = 3, t = 4)$
(2, 4)	2	2	0	$(j = 4, t = 2)$
(3, 4)	3	3	0	$(j = 4, t = 3)$

The relation r^1 in this example is

a	b	c	d	
(3, 2)	0	3	(3, 2)	$(j = 2, t = 3)$
(4, 2)	0	4	(4, 2)	$(j = 2, t = 4)$
(2, 3)	(2, 3)	2	2	$(j = 3, t = 2)$
(4, 3)	(4, 3)	4	4	$(j = 3, t = 4)$
0	(2, 4)	(2, 4)	2	$(j = 4, t = 2)$
0	(3, 4)	(3, 4)	3	$(j = 4, t = 3)$

r^2 equals r^1 augmented with the tuple

a	b	c	d
0	0	2	2

r^3 equals r^1 augmented with

a	b	c	d
0	0	3	3

and r^4 equals r^1 augmented with

a	b	c	d
0	0	4	4

It is a straightforward verification that for $i, l = 1, \dots, 4$, $i \neq l$ it holds that $\pi_{X_i}(r^l) = r_i$. Hence (r_1, r_2, r_3, r_4) is 3-wise consistent. It is also easy to verify that $r_1 \bowtie r_2 \bowtie r_3 \bowtie r_4 = r^1$. Since $0_{X_1} \notin \pi_{X_1}(r^1)$, $\pi_{X_1}(r_1 \bowtie r_2 \bowtie r_3 \bowtie r_4) \neq r_1$. Hence (r_1, r_2, r_3, r_4) is not globally consistent, in accordance with Lemma 6.5.

LEMMA 6.6. *Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$. Suppose that $J: X_1 \bowtie \dots \bowtie X_k$ is a reduced and connected jd over R and let $\mathcal{E}' = \{X_1, \dots, X_l\}$, $l < k$ be a hinge of \mathcal{H}_J . Let $n \leq l$ and suppose (r_1, \dots, r_l) is an n -wise consistent instance over (X_1, \dots, X_l) . Then (r_1, \dots, r_l) can be extended to an n -wise consistent instance (r_1, \dots, r_k) over (X_1, \dots, X_k) .*

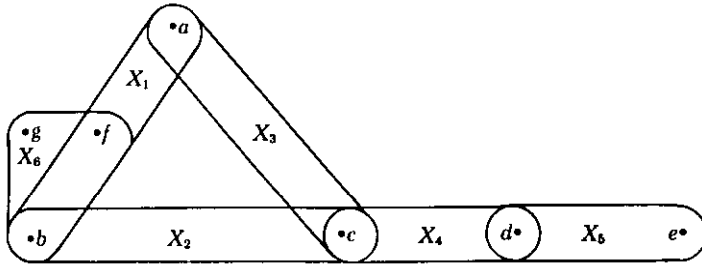
PROOF. Choose for each connected component of \mathcal{H}_J with respect to \mathcal{E}' a fixed corresponding separating edge of \mathcal{E}' . For $l+1 \leq j \leq k$, we denote with X_j' the separating edge corresponding to the connected component to which X_j belongs. Suppose $l+1 \leq j \leq k$. Let r_j' be the instance in $\{r_1, \dots, r_l\}$ defined over X_j' and let r_j be the instance over X_j defined as

$$r_j = \{t \mid \exists t' \in r_j': t[X_j \cap X_j'] = t'[X_j \cap X_j'] \ \& \ \forall a \in X_j \setminus X_j': t(a) = 0\}.$$

Then (r_1, \dots, r_k) is an n -wise consistent instance of (X_1, \dots, X_k) . (Details of the proof can be found in the Appendix. \square)

The constructions made in the proof of the previous lemma are illustrated in Example 6.4, below.

Example 6.4. Let $X_1 \bowtie \dots \bowtie X_6$ be represented by the following hypergraph \mathcal{H} :



Clearly, $\{X_1, X_2, X_3\}$ is a hinge of \mathcal{H} . Let (r_1, r_2, r_3) be the following instance over (X_1, X_2, X_3) :

a	b	f	b	c	a	c
1	1	1	1	2	1	2
2	1	2	1	1	2	1
1	1	2			1	1

Clearly, (r_1, r_2, r_3) is (3-wise) consistent because r_1, r_2 , and r_3 are the projections onto X_1, X_2 , and X_3 , respectively, of, for instance,

a	b	c	f
1	1	2	1
2	1	1	2
1	1	1	1
1	1	1	2

We are now going to extend (r_1, r_2, r_3) to a 3-wise consistent instance (r_1, \dots, r_6) over (X_1, \dots, X_6) . We first have to assign fixed separating edges to the connected components of \mathcal{H} with respect to \mathcal{E}' . Therefore, we choose X_2 as a separating edge corresponding to $\{X_4, X_5\}$. Clearly, X_1 is the only separating

edge corresponding to $\{X_6\}$. Using the notation introduced in the proof of Lemma 6.6, we have $X'_4 = X_2$, $X'_5 = X_2$, $X'_6 = X_1$, $r'_4 = r_2$, $r'_5 = r_2$, and $r'_6 = r_1$.

According to the definition given in the proof of Lemma 6.6, r_4 , r_5 , and r_6 are, respectively,

c	d	d	e	b	f	g
2	0	0	0	1	1	0
1	0			1	2	0

(r_1, \dots, r_6) is 3-wise consistent, as can be verified in a straightforward manner (e.g., by using Lemma 6.1).

We can now prove that n -cyclicity is implied by " n -wise consistency implies global consistency."

THEOREM 6.1. *Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$. Let $J: X_1 \bowtie \dots \bowtie X_k$ be a reduced and connected jd over R . Let $n \geq 2$. If each n -wise consistent instance (r_1, \dots, r_k) over (X_1, \dots, X_k) is also globally consistent, then J is n -cyclic.*

PROOF. Suppose Theorem 6.1 is false. Then let $J: X_1 \bowtie \dots \bowtie X_k$ be a counterexample to Theorem 6.1. that is minimal, first with respect to the number of edges and, second, with respect to the number of attributes. Clearly, $k > n$. We distinguish two cases.

Case 1. J is decomposable. Then, by Definition 5.4 and Theorem 5.1, \mathcal{H}_J contains a minimal hinge of size l , $n < l < k$, say $\{X_1, \dots, X_l\}$. Suppose now that (r_1, \dots, r_l) is an n -wise consistent instance over (X_1, \dots, X_l) . By Lemma 6.6, (r_1, \dots, r_l) can be extended to an n -wise consistent instance (r_1, \dots, r_k) over (X_1, \dots, X_k) . By assumption, (r_1, \dots, r_k) is also globally consistent. Hence (r_1, \dots, r_l) is globally consistent over (X_1, \dots, X_l) . Since $\{X_1, \dots, X_l\}$ is a minimal hinge of \mathcal{H}_J , it follows that the hypergraph representation of $J': X_1 \bowtie \dots \bowtie X_l$ does not contain hinges. So, by Theorem 4.1, J' is undecomposable, and hence not n -cyclic, since $l > n$ (Lemma 5.1). Thus J' is also a counterexample to Theorem 6.1, in contradiction with the minimality of J with respect to the number of edges. Therefore Case 1 cannot occur, and hence we are in Case 2.

Case 2. J is undecomposable. Clearly, the conditions of Lemma 6.5 are satisfied. Thus there exists an instance (r_1, \dots, r_k) over (X_1, \dots, X_k) that is $k-1$ -wise consistent, and hence n -wise consistent (Lemma 6.1), but not globally consistent, in contradiction with the assumption of J being a counterexample to Theorem 6.1. Hence our initial assumption is false and Theorem 6.1 holds. \square

The converse of Theorem 6.1 also holds.

THEOREM 6.2. *Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$. Let $J: X_1 \bowtie \dots \bowtie X_k$ be a reduced and connected jd over R . Let $n > 1$ and suppose J is n -cyclic. Let (r_1, \dots, r_k) be an instance over (X_1, \dots, X_k) that is n -wise consistent. Then (r_1, \dots, r_k) is also globally consistent.*

PROOF. Without loss of generality, we can assume $n \leq k$. We prove the theorem by induction on $|J|$. For $k = n$, the theorem is obviously true. Therefore, suppose that $k > n$ and that the theorem holds for all J' with $|J'| < k$. Since J is

n -cyclic, there exists a minimal hinge of \mathcal{H}_J , say $\{X_1, \dots, X_l\}$ with $l \leq n$. By the induction hypothesis,

$$\tilde{r} = r_1 \bowtie \dots \bowtie r_l$$

satisfies $\pi_{X_i}(\tilde{r}) = r_i$ for all $i = 1, \dots, l$. We still have to show that $(\tilde{r}, r_{l+1}, \dots, r_k)$ is n -wise consistent over $(\cup_{i=1}^l X_i, X_{l+1}, \dots, X_k)$. Since $(\cup_{i=1}^l X_i) \bowtie X_{l+1} \bowtie \dots \bowtie X_k$ is clearly also n -cyclic, the induction hypothesis and Lemma 6.2 will then yield the desired result. Therefore, choose for each connected component of \mathcal{H}_J with respect to $\{X_1, \dots, X_l\}$ a fixed corresponding separating edge and let, for $1 \leq i \leq l$, \mathcal{F}_i be the set of all the edges of connected components for which X_i is chosen as corresponding separating edge. Now take n instances out of $\tilde{r}, r_{l+1}, \dots, r_k$. We may assume that \tilde{r} is among those n instances, since in the other case the proof of their being consistent is trivial. For the sake of simplicity of notation, suppose that those n instances are $\tilde{r}, r_{l+1}, \dots, r_{l+n-1}$. Again, without loss of generality, (by rearranging indices) we can assume that $\mathcal{F}_1 \neq \emptyset$ and that $\mathcal{F}_1 = \{X_{l+1}, \dots, X_t\}$ with $l+1 \leq t \leq l+n-1$. Define

$$s = r_{l+1} \bowtie \dots \bowtie r_t.$$

Clearly, because of the assumptions, $\pi_{X_j}(s) = r_j$ for all $j = 1, \dots, t$. Note that by Lemma 6.2 (r_1, s) is a pair-wise consistent instance over

$$\left(X_1, \bigcup_{j=l+1}^t X_j \right), \quad (*)$$

since $(r_1, r_{l+1}, \dots, r_t)$ is globally consistent over $(X_1, X_{l+1}, \dots, X_t)$ by the n -wise consistency of (r_1, \dots, r_k) over (X_1, \dots, X_k) . Now

$$\begin{aligned} \pi_{(\cup_{i=1}^l X_i) \cap (\cup_{j=l+1}^t X_j)}(\tilde{r}) &= \pi_{X_1 \cap (\cup_{j=l+1}^t X_j)}(\tilde{r}) && \text{(by definition of } \mathcal{F}_1) \\ &= \pi_{X_1 \cap (\cup_{j=l+1}^t X_j)}(\pi_{X_1}(\tilde{r})) \\ &= \pi_{X_1 \cap (\cup_{j=l+1}^t X_j)}(r_1) \\ &= \pi_{X_1 \cap (\cup_{j=l+1}^t X_j)}(s) && \text{(by } (*) \text{ and Lemma 6.3)} \\ &= \pi_{(\cup_{i=1}^l X_i) \cap (\cup_{j=l+1}^t X_j)}(s) && \text{(by definition of } \mathcal{F}_1). \end{aligned}$$

Hence by Lemma 6.3,

$$\begin{cases} \pi_{\cup_{i=1}^l X_i}(\tilde{r} \bowtie s) = \tilde{r} \\ \pi_{\cup_{j=l+1}^t X_j}(\tilde{r} \bowtie s) = s \end{cases}$$

and hence by Lemma 6.2,

$$\pi_{X_i}(\tilde{r} \bowtie s) = r_i \quad \text{for } i = 1, \dots, t.$$

Note that $\tilde{r} \bowtie s = r_1 \bowtie \dots \bowtie r_t$. By continuing the above argument we eventually get

$$\begin{cases} \pi_{\cup_{i=1}^l X_i}(\tilde{r} \bowtie r_{l+1} \bowtie \dots \bowtie r_{l+n-1}) = \tilde{r} \\ \pi_{X_i}(r_1 \bowtie \dots \bowtie r_{l+n-1}) = r_i \quad \text{for } i = 1, \dots, l+n-1 \end{cases}$$

This completes the proof. \square

Theorems 6.1 and 6.2 can be summarized as follows.

THEOREM 6.3. *Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$. Let $J: X_1 \bowtie \dots \bowtie X_k$ be a reduced and connected jd over R . Let $n \geq 2$. Then J is n -cyclic if and only if each n -wise consistent instance (r_1, \dots, r_k) over (X_1, \dots, X_k) is also globally consistent.*

Theorem 6.3 generalizes the well-known result that acyclicity is equivalent with "pair-wise consistency implies global consistency." From Theorem 6.3 we can derive an important corollary, as follows:

COROLLARY 6.1. *Let $n > 1$ be a fixed integer and let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$. Let $J: X_1 \bowtie \dots \bowtie X_k$ be an n -cyclic reduced and connected jd over R . Let m be the number of attributes in Ω . Then checking an instance (r_1, \dots, r_k) over (X_1, \dots, X_k) for global consistency can be done in a time that is polynomial in m, k , and l , where $l = \max\{|r_i|; i = 1, \dots, k\}$.*

PROOF. Because of J being n -cyclic, it suffices to check (r_1, \dots, r_k) for n -wise consistency. Let r_{t_1}, \dots, r_{t_n} be n relations out of r_1, \dots, r_k . Making the join $r_{t_1} \bowtie \dots \bowtie r_{t_n}$ can be done in $O(l^n m)$ time. Checking $r_{t_i} \subseteq \pi_{X_{t_i}}(r_{t_1} \bowtie \dots \bowtie r_{t_n})$ can be done in $O(l^{n+1} m)$ time. Hence, to check this for all $i = 1, \dots, n$, we need $O(l^{n+1} m)$ time, since n is fixed. Since we can choose n relations out of k in $\binom{k}{n} \leq k^n$ ways, we obtain a total time complexity of $O(mk^n l^{n+1})$. \square

Corollary 6.1 says that checking for global consistency remains polynomial if we restrict the complexity of the jds we allow, but not necessarily to acyclicity.

APPENDIX

Here we include the details of the proofs of the technical Lemmas 6.5 and 6.6. For the convenience of the reader, we also recall the statements of the lemmas.

LEMMA 6.5. *Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega, k > 2$. Let $X = \bigcup_{i=1}^k X_i$. Suppose $J: X_1 \bowtie \dots \bowtie X_k$ is a reduced and connected undecomposable jd without attributes occurring in only one edge. Define the instance (r_1, \dots, r_k) over (X_1, \dots, X_k) as follows:*

$$\begin{aligned} r_1 &= \{0_{X_1}\} \cup \{0_{X_1 \cap X_j}(t, j)_{X_1 \setminus X_j}; 2 \leq j \neq t \leq k\}; \\ r_i &= \{0_{X_1 \cap X_i \cap X_j} t_{(X_i \cap X_j) \setminus X_1}(t, j)_{X_i \setminus X_j}; 2 \leq j \neq t \leq k\}, \quad i \neq 1. \end{aligned}$$

Then (r_1, \dots, r_k) is $k-1$ -wise consistent, but not globally consistent.

PROOF. Recall that the constructions made in this lemma are illustrated in Example 6.3. We already showed that (r_1, \dots, r_k) is $k-1$ -wise consistent. Hence it suffices here to show that they are not k -wise consistent.

Suppose that (r_1, \dots, r_k) is globally consistent. Then there exists $\tau \in r_1 \bowtie \dots \bowtie r_k$ with $\tau[X_1] = 0_{X_1}$ and $\tau[X_i] \in r_i$ for $i \neq 1$. Therefore we can denote $\tau[X_i]$ for $i \neq 1$ as

$$\tau[X_i] = 0_{X_1 \cap X_i \cap X_j} t_{(X_i \cap X_j) \setminus X_1}(t_i, j_i)_{X_i \setminus X_j} \quad \text{with} \quad 2 \leq j_i \neq t_i \leq k.$$

We distinguish two possible cases:

Case 1. There exists $i \neq 1$ with $X_i \neq X_{j_i}$. Suppose $2 \leq r \neq j_i \leq k$. Since J is undecomposable, we have, by Theorem 4.1, that $\mathcal{E}_J \setminus \{X_{j_i}\}$ is connected with respect to $\{X_{j_i}\}$. Hence there exists a sequence X_{u_1}, \dots, X_{u_w} with

$$\begin{aligned} X_{u_1} &= X_i; \\ X_{u_w} &= X_r; \\ (X_{u_w} \setminus X_{j_i}) \cap (X_{u_{w+1}} \setminus X_{j_i}) &\neq \emptyset \quad \text{for } 1 \leq w < k. \end{aligned}$$

Consider

$$\tau[X_{u_2}] = 0_{X_1 \cap X_{u_2} \cap X_{j_i}} t_{u_2(X_{u_2} \cap X_{j_i}) \setminus X_1}(t_{u_2}, j_{u_2})_{X_{u_2} \setminus X_{j_i}}.$$

Since $X_{u_2} \cap (X_{u_1} \setminus X_{j_i}) \neq \emptyset$, it follows that necessarily $(t_{u_2}, j_{u_2}) = (t_i, j_i)$ and

$$\tau[X_{u_2}] = 0_{X_1 \cap X_{u_2} \cap X_{j_i}} t_{i(X_{u_2} \setminus X_{j_i}) \setminus X_1}(t_i, j_i)_{X_{u_2} \setminus X_{j_i}},$$

since this is the only tuple in r_{u_2} containing the value (t_i, j_i) . By continuing this argument we eventually obtain

$$\tau[X_r] = 0_{X_1 \cap X_r \cap X_{j_i}} t_{i(X_r \cap X_{j_i}) \setminus X_1}(t_i, j_i)_{X_r \setminus X_{j_i}},$$

and this holds for $2 \leq r \neq j_i \leq k$. Since $\tau[X_1] = 0_{X_1}$, it follows that

$$\begin{aligned} \forall a \in X_1: \tau(a) &= 0; \\ \forall a \in \cup \{(X_r \cap X_{j_i}) \setminus X_1 \mid 2 \leq r \neq j_i \leq k\}: \tau(a) &= t_i; \\ \forall a \in \cup \{X_r \setminus X_{j_i} \mid 2 \leq r \neq j_i \leq k\}: \tau(a) &= (t_i, j_i). \end{aligned}$$

By assumption (no attributes occurring in only one edge), we have that $\cup \{X_r; 1 \leq r \neq j_i \leq k\} = X$. Hence $X \setminus X_1 = \cup \{X_r; 2 \leq r \neq j_i \leq k\} \setminus X_1$. This gives us that

$$((\cup \{X_r; 2 \leq r \neq j_i \leq k\}) \cap X_{j_i}) \setminus X_1 = X_{j_i} \setminus X_1.$$

Since $X_1 \cup (X_{j_i} \setminus X_1) \cup (X \setminus (X_1 \cup X_{j_i})) = X$, it follows that

$$\tau = 0_{X_1} t_{iX_{j_i} \setminus X_1}(t_i, j_i)_{X \setminus (X_1 \cup X_{j_i})},$$

and hence $X \setminus (X_1 \cup X_{j_i}) = (\cup \{X_r; 2 \leq r \neq j_i \leq k\}) \setminus X_{j_i}$. Thus

$$\begin{aligned} X_1 &= X_1 \cap (\cup \{X_r; 2 \leq r \leq k\}) \\ &\subseteq (X_1 \cap X_{j_i}) \cup (X_1 \cap (\cup \{X_r; 2 \leq r \neq j_i \leq k\})) \\ &\subseteq X_{j_i} \cup (X_1 \cap ((\cup \{X_r; 2 \leq r \neq j_i \leq k\}) \setminus X_{j_i})) \\ &= X_{j_i} \cup (X_1 \cap (X \setminus (X_1 \cup X_{j_i}))) \\ &= X_{j_i}. \end{aligned}$$

This is clearly in contradiction with our assumption of J being reduced. Hence we are in the second case.

Case 2. For all $2 \leq i \leq k$, $X_i = X_{j_i}$. This means that for $i \neq 1$ we have

$$\tau[X_i] = 0_{X_1 \cap X_i} t_{iX_i \setminus X_1}.$$

Since, by Theorem 4.1, $\{X_2, \dots, X_k\}$ is connected with respect to $\{X_1\}$, we can deduce as in Case 1 that for $2 \leq i, j \leq k$ we have that $t_i = t_j$. Let us call this value t . Since $\cup \{X_i; i \neq 1\} = X$, we have that

$$\tau = 0_{X_1} t_{X \setminus X_1}.$$

This implies

$$\tau[X_i] = 0_{X_1 \cap X_i} t_{X_i \setminus X_1}.$$

This tuple, however, is *not* in r_t . Hence our assumption is false and (r_1, \dots, r_k) is not globally consistent. This completes the proof. \square

LEMMA 6.6. *Let $R(\Omega)$ be a relation scheme and let $X_1, \dots, X_k \subseteq \Omega$. Suppose that $J: X_1 \bowtie \dots \bowtie X_k$ is a reduced and connected jd over R and let $\mathcal{E}' \{X_1, \dots, X_l\}$, $l < k$ be a hinge of \mathcal{R}_J . Let $n \leq l$ and suppose (r_1, \dots, r_l) is an n -wise consistent instance over (X_1, \dots, X_l) . Then (r_1, \dots, r_l) can be extended to an n -wise consistent instance (r_1, \dots, r_k) over (X_1, \dots, X_k) .*

PROOF. We recall that we chose for each connected component of \mathcal{R}_J with respect to \mathcal{E}' a fixed corresponding separating edge of \mathcal{E}' . For $l+1 \leq j \leq k$, we denoted with X'_j the separating edge corresponding to the connected component to which X_j belongs and with r'_j the instance in $\{r_1, \dots, r_l\}$ defined over X'_j . Finally, for $l+1 \leq j \leq k$, the instance r_j over X_j was defined as

$$r_j = \{t \mid \exists t' \in r'_j: t[X_j \cap X'_j] = t'[X_j \cap X'_j] \ \& \ \forall a \in X_j \setminus X'_j: t(a) = 0\}.$$

It remains to prove that (r_1, \dots, r_k) is indeed an n -wise consistent instance over (X_1, \dots, X_k) .

First, let us put for $1 \leq j \leq l$:

$$r'_j = r_j.$$

Suppose r_{t_1}, \dots, r_{t_n} are n instances out of r_1, \dots, r_k . We show that $\pi_{X_{t_j}}(r_{t_1} \bowtie \dots \bowtie r_{t_n}) = r_{t_j}$ for $1 \leq j \leq n$. It suffices to show the inclusion from the right to the left, because the other one is obvious. We distinguish two cases.

Case 1. $1 \leq t_j \leq l$. Let $\tau' \in r_{t_j}$. Because of the n -wise consistency of r_1, \dots, r_l , there exists $\tilde{r} \in r'_{t_1} \bowtie \dots \bowtie r'_{t_n}$ with $\tilde{r}[X_{t_j}] = \tau'$. Now define τ over $\cup_{j=1}^n X_{t_j}$ by

$$\begin{cases} \tau \left[\left(\bigcup_{j=1}^n X_{t_j} \right) \cap \left(\bigcup_{j=1}^n X_{u_j} \right) \right] = \tilde{r} \left[\left(\bigcup_{j=1}^n X_{t_j} \right) \cap \left(\bigcup_{j=1}^n X_{u_j} \right) \right] \\ \forall a \in \left(\bigcup_{j=1}^n X_{t_j} \right) \setminus \left(\bigcup_{j=1}^n X_{u_j} \right): \tau(a) = 0. \end{cases}$$

Let $1 \leq j' \leq n$. If $1 \leq t_{j'} \leq l$, clearly, $\tau[X_{t_{j'}}] = \tilde{r}[X_{t_{j'}}] \in r_{t_{j'}}$. Suppose now that $l+1 \leq t_{j'} \leq k$. Then $\tau[X_{t_{j'}} \cap X'_{t_{j'}}] = \tilde{r}[X_{t_{j'}} \cap X'_{t_{j'}}]$, and for all $a \in X_{t_{j'}} \setminus X'_{t_{j'}}$, $\tau(a) = 0$. Hence, by definition, $\tau[X_{t_{j'}}] \in r_{t_{j'}}$. Thus, for all $1 \leq j' \leq n$, we have that $\tau[X_{t_{j'}}] \in r_{t_{j'}}$, and hence $\tau \in r_{t_1} \bowtie \dots \bowtie r_{t_n}$. In particular, $\tau[X_{t_j}] = \tau'$. This completes the proof in Case 1.

Case 2. $l+1 \leq t_j \leq k$. Let $\tau' \in r_{t_j}$. Then there exists $\tau'' \in r'_{t_j}$ such that $\tau'[X_{t_j} \cap X'_{t_j}] = \tau''[X_{t_j} \cap X'_{t_j}]$, and for all $a \in X_{t_j} \setminus X'_{t_j}$, $\tau'(a) = 0$. As in Case 1,

there exists $\hat{\tau} \in r'_1 \bowtie \dots \bowtie r'_n$ with $\hat{\tau}[X'_i] = \tau''$. The construction of τ and the rest of the proof for Case 2 goes as in Case 1. \square

ACKNOWLEDGMENTS

I wish to thank R. Fagin and M. Vardi for some interesting discussions about this subject. I also wish to thank the referees whose comments on a previous version of this paper were very helpful in improving it.

REFERENCES

1. AHO, A. V., BEERI, C., AND ULLMAN, J. D. The theory of joins in relational databases. *ACM Trans. Database Syst.* 4, 3 (1979), 297-314.
2. BEERI, C., FAGIN, R., MAIER, D., MENDELZON, A. O., ULLMAN, J. D., AND YANNAKAKIS, M. Properties of acyclic database schemes. In *Proceedings 13th Annual ACM Symposium on the Theory of Computing* (1981), ACM, New York, 355-362.
3. BEERI, C., FAGIN, R., MAIER, D., AND YANNAKAKIS, M. On the desirability of acyclic database schemes. *J. ACM* 30, 3 (July 1983).
4. BERGÉ, C. *Graphes et Hypergraphes*. Dunod, Paris, 1970.
5. CODD, E. F. A relational model of data for large shared data banks. *Commun. ACM* 13, 6 (June 1970), 377-387.
6. CODD, E. F. Further normalizations of the relational data base model. In *Data Base Systems*, R. Rustin, Ed., Prentice Hall, Englewood Cliffs, N.J., 1972, 33-64.
7. FAGIN, R. Multivalued dependencies and a new normal form for relational databases. *ACM Trans. Database Syst.* 2, 3 (1977), 262-278.
8. FAGIN, R. Acyclic database schemes (of various degrees): A painless introduction. In *Proceedings of CAAP 1983* (L'Aquila, 1983).
9. FAGIN, R., MENDELZON, A. O., AND ULLMAN, J. D. A simplified universal relation assumption and its properties. *ACM Trans. Database Syst.* 7, 3 (Sept. 1982), 343-360.
10. GYSSENS, M., AND PAREDAENS, J. A decomposition methodology for cyclic databases. In *Advances in Database Theory, vol 2*, H. Gallaire, J. Minker, and J. M. Nicolas, Eds., Plenum Press, New York, 1973.
11. GYSSENS, M., AND PAREDAENS, J. On the decomposition of join dependencies. In *Proceedings of the 3rd Symposium on Principles of Database Systems* (Waterloo, Ont., April 1984).
12. GYSSENS, M. Embedded join dependencies as a tool for decomposing full join dependencies. In *Proceedings of the 4th Symposium on Principles of Database Systems* (Portland, Ore., Apr. 1985), 205-214.
13. GYSSENS, M. Decompositions of join dependencies in the relational database model. Ph.D. thesis, Univ. of Antwerp, June 1985.
14. MAIER, D., MENDELZON, A. O., AND SAGIV, Y. Testing implication of data dependencies. *ACM Trans. Database Syst.* 4, 4 (1980), 455-469.
15. PAREDAENS, J., AND VAN GUCHT, D. An application of the theory of graphs and hypergraphs to the decompositions of relational database schemes. In *Proceedings of CAAP, 1983* (L'Aquila, Mar. 9-11, 1983).
16. RISSANEN, J. Theory of joins for relational databases—A tutorial survey. In *Proceedings of the 7th Symposium on the Mathematical Foundations of Computer Science; Lecture Notes in Computer Science 64*, Springer Verlag, 1978, 537-551.
17. THALHEIM, B. A complete axiomatization for join dependencies in relations. Tech. Rep., Technische Univ. Dresden, Sektion Math., 07-08-84.
18. ULLMAN, J. *Principles of Database Systems*. 2nd ed., Pitman, Marshfield, Mass., 1982.
19. ZANIOLO, C. Analysis and design of relational schemata for database systems. Tech. Rep. UCLA-ENG-7669, Dept. of Computer Science, Univ. of California, Los Angeles, 1976.

Received October 1984; revised July 1985; accepted September 1985.

ACM Transactions on Database Systems, Vol. 11, No. 1, March 1986.