Characterisations of multivalued dependency implication over undetermined universes

Sebastian Link

Department of Computer Science, The University of Auckland, New Zealand

1. Introduction

The relational model of data, introduced by Codd [22], has greatly contributed to the success of database management systems. In this model of data the database is viewed as a collection of relations, where each relation is a set of tuples over some domain of possible values. From the viewpoint of finite model theory, a relational database is a finite structure over a relational signature [67]. This structure provides a mere syntactic definition of the data, and does not allow one to represent semantics in the form of the relationship between certain data values.

One approach to overcome this deficiency is to specify the semantics of the relations explicitly. These semantic specifications are known as integrity constraints since they prescribe which database instances are meaningful for the application at hand and which are meaningless. Of particular importance are the integrity constraints known as data dependencies, or dependencies for short. The intuitive meaning of “dependency” is that the occurrence of certain data values in a relation enforces some properties or existence of other data values. In this sense, the latter entries are “dependent” on the former ones. There are at least 100 hundred different classes of dependencies which can be utilised for improving the semantics of the target database [37,89,92].

Many papers in dependency theory, i.e. the study of the language for specifying the semantics of a database, deal with different aspects of the implication problem. The problem is to decide for an arbitrary set $\Sigma$ of dependencies and an
arbitrary dependency \( \varphi \) whether \( \Sigma \) logically implies \( \varphi \). The reason for the prominence of this problem is manifold. An algorithm for testing the implication of dependencies enables us to test whether two given sets of dependencies are equivalent or whether a given set of dependencies is redundant. A solution to these problems is a big step towards automated database schema design [12,14] which some researchers see as the ultimate goal in dependency theory [9]. Moreover, such an algorithm can be used in relational normalisation theory and practice involving many normal form proposals \([8,9,14,15,18,23]\), requirements engineering and schema validation \([61,72]\), data mining \([73]\), in database security \([19,20]\), view maintenance \([57,62]\) and in query optimisation \([27,30]\). More recently, the implication problem has received a lot of attention in other data models as well \([4,6,39,46,50,51,53,59,64,66,86,88,90,96–99]\). New application areas involve data cleaning \([38]\), data transformations \([25]\), consistent query answering \([21]\) and data exchange \([35,36,41,75]\).

Multivalued dependencies (MVDs) \([26,34,102]\) are an important class of dependencies. Informally, a relation \( r \) over the universe \( R \) of attributes satisfies the MVD \( X \rightarrow Y \) whenever the value on \( Y \) determines the set of values on \( X \) in a relation independently of the set of values on \( R \setminus Y \). This suggests that the universe \( R \) is overloaded in the sense that it carries two independent facts \( XY \) and \( X(R \setminus Y) \). Indeed, the relation \( r \) exhibits the MVD \( X \rightarrow Y \) precisely when \( r \) is decomposable into its projections \( r[XY] \) and \( r[X(R \setminus Y)] \) without loss of information, i.e., when \( r \) is equal to the natural join \( r[XY] \Join r[X(R \setminus Y)] \) \([34]\), cf. Example 1. Multivalued dependencies generalise functional dependencies \([23]\) which are expressions of the form \( X \rightarrow Y \) and which are satisfied in a relation if every pair of tuples that agree on all the attributes in \( X \) also agree on all the attributes in \( Y \). Hence, the values on the attributes in \( X \) functionally determine the values on the attributes in \( Y \). The satisfaction of the functional dependency \( X \rightarrow Y \) is a sufficient, but not necessary, condition for \( r \) to be the lossless join of its projections \( r[XY] \) and \( r[X(R \setminus Y)] \). The satisfaction of the corresponding multivalued dependency \( X \rightarrow Y \), however, provides a sufficient and necessary condition. Recently, extensions of multivalued dependencies have been found very useful for various design problems in advanced data models such as the nested relational data model \([39,52]\), fuzzy databases \([86]\), temporal databases \([59]\), the Entity-Relationship model \([90]\), data models that support nested lists \([50,56,68]\), data models that can handle partial information \([54,69]\), and XML \([83,94,95]\).

**Example 1.** Let \( R \) denote the relation schema \( \{Employee, Child, Salary\} \). The relation schema is used to store information about the name of an employee, the name of an employee’s child and the salary of the employee. Intuitively, the information on an employee’s child is independent of the information on the employee’s salary. This can be expressed as the multivalued dependency \( Employee \rightarrow Child \). The context of this example will be used to illustrate our findings throughout the paper. A snapshot of a possible relation \( r \) over \( R \) could be:

<table>
<thead>
<tr>
<th>Employee</th>
<th>Child</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homer</td>
<td>Bart</td>
<td>4000</td>
</tr>
<tr>
<td>Homer</td>
<td>Lisa</td>
<td>4500</td>
</tr>
</tbody>
</table>

The projections \( r[\{Employee,Child\}] \) and \( r[\{Employee,Salary\}] \) of \( r \), respectively, are given by:

<table>
<thead>
<tr>
<th>Employee</th>
<th>Child</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homer</td>
<td>Bart</td>
<td>4000</td>
</tr>
<tr>
<td>Homer</td>
<td>Lisa</td>
<td>4500</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Employee</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homer</td>
<td>4000</td>
</tr>
<tr>
<td>Homer</td>
<td>4500</td>
</tr>
</tbody>
</table>

The natural join \( r[\{Employee,Child\}] \Join r[\{Employee,Salary\}] \) is given by:

<table>
<thead>
<tr>
<th>Employee</th>
<th>Child</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homer</td>
<td>Bart</td>
<td>4000</td>
</tr>
<tr>
<td>Homer</td>
<td>Lisa</td>
<td>4500</td>
</tr>
<tr>
<td>Homer</td>
<td>Bart</td>
<td>4500</td>
</tr>
<tr>
<td>Homer</td>
<td>Lisa</td>
<td>4000</td>
</tr>
</tbody>
</table>

Consequently, \( r \) violates the multivalued dependency \( Employee \rightarrow Child \).

The characteristic of decomposing a relation without loss of information is fundamental to relational database design, in particular for the Fourth Normal Form proposal 4NF \([34]\). A relation schema that satisfies the 4NF condition is guaranteed to be free of data redundancies defined with respect to both functional and multivalued dependencies, and is therefore also free of update anomalies \([93]\). Consequently, it is a desirable goal in database design to obtain a database schema in which all relation schemata satisfy the 4NF condition. It has been stated in a number of practitioner reports, e.g. \([76,85]\), that modelling multivalued dependencies is rather difficult and often confuses people in practice. If many participants are involved in the database design process, then modelling becomes even more challenging. Moreover, many practitioners and academics are under the impression that data violating the 4NF condition is rarely encountered in practice. However, it has been shown that this is a misconception \([101]\). Consequently, the need to understand multivalued dependencies and...
how to use 4NF is extremely important. Indeed, a lot of research has been devoted to studying the implication problem of multivalued dependencies [7,11,16,31,40,44,58,65,68,70,71,78,81,91].

The classical notion of a multivalued dependency (MVD) [34] is dependent on the underlying universe $R$. Syntactically, this dependence is reflected by the $R$-complementation rule which enables us to conclude that every relation over $R$ that satisfies the MVD $X \rightarrow Y$ will also satisfy the MVD $X \rightarrow R \rightarrow Y$. In Example 1 for instance, the MVD $Employee \rightarrow Salary$ can be inferred by a single application of the $R$-complementation rule to the MVD $Employee \rightarrow Employee, Child$. In all sets of sound and complete inference rules of MVDs, the $R$-complementation rule (or a slight variation of it) is special in the sense that it is the only inference rule in that axiomatisation which is dependent on $R$. This dependence on the underlying universe imposes an additional constraint on solving the implication problem: the underlying universe has to be fixed before any attempt can be made to derive any implied multivalued dependencies. For a set $\Sigma \cup \{\varphi\}$ of MVDs over a relation schema $R$, we will therefore speak of the $R$-implication of $\varphi$ by $\Sigma$. This restriction distinguishes MVDs from other dependencies, e.g. functional dependencies whose satisfaction does not depend on the underlying universe. For instance, the well-known synthesis approach towards achieving the Third Normal Form condition is only possible because this restriction does not hold for functional dependencies [14,18]. In fact, one of the open problems in relational database design is a generalisation of the synthesis approach to multivalued dependencies. This problem, however, appears to be difficult to address when the underlying set of attributes is assumed to be fixed.

The dependence of MVD implication on the underlying universe has motivated an investigation of multivalued dependencies in the context of an undetermined universe, i.e., where the assumption of a fixed underlying relation schema is dropped. Biskup [17] introduced an alternative notion of semantic implication in which the underlying universe is left undetermined. In the same paper, Biskup established a sound and complete set $\mathcal{E}_1$ of inference rules for the implication of MVDs in undetermined universes. If $\mathcal{E}_1^\top$ results from adding the $R$-complementation rule to $\mathcal{E}_1$, then $\mathcal{E}_1^\top$ becomes an axiomatisation for the $R$-implication of MVDs, for all fixed universes $R$. In fact, for every inference of an MVD by $\mathcal{E}_1^\top$ there is an inference of the same MVD by $\mathcal{E}_1$ in which the $R$-complementation rule is applied at most once, and if it is applied, then in the last step of the inference. This indicates that the $R$-implication rule simply reflects a part of the decomposition process, and does not necessarily infer semantically meaningful consequences.

Example 2. Consider the universe $R$ of Example 1. The MVD $Employee \rightarrow Salary$ is $R$-implied by the MVD $Employee \rightarrow Employee, Child$. However, in the universe $R' = \{Employee, Child, Salary, Year\}$ the MVD $Employee \rightarrow Salary$ is not $R'$-implied by the MVD $Employee \rightarrow Employee, Child$ as the following example shows:

<table>
<thead>
<tr>
<th>Employee</th>
<th>Child</th>
<th>Salary</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homer</td>
<td>Lisa</td>
<td>4000</td>
<td>2008</td>
</tr>
<tr>
<td>Homer</td>
<td>Lisa</td>
<td>4500</td>
<td>2009</td>
</tr>
</tbody>
</table>

Consequently, the MVD $Employee \rightarrow Salary$ is a meaningful consequence in the universe $R$, but not a consequence in any other universes.

Unfortunately, research has not been continued in this direction but focused almost exclusively on the original notion of $R$-implication. Only recently, an $O(\log |Attr(\Sigma) \cup X| \times |\Sigma|)$-algorithm for deciding the implication of an MVD $X \rightarrow Y$ by a finite set $\Sigma$ of MVDs in undetermined universes has been established [68]. Here, $|\Sigma|$ denotes the space required for writing down the MVDs in $\Sigma$ and $Attr(\Sigma)$ denotes the set of the attributes that occur in $\Sigma$. Since research on data dependencies experiences a recent revival [2,4,35,36,38,41,42,49,51,68,69,83,90,94,95] it seems desirable to further extend the knowledge on the relational theory. An advancement of such knowledge may simplify the quest of finding suitable and comprehensible extensions of MVDs to other data models. Furthermore, the results of this paper show that fixing a universe of attributes is not an assumption that is necessary for utilising MVDs, and e.g., for data modelling, database design or query optimisation based on MVDs.

Our research bridges at least three different areas: (i) dependency theory and database design, (ii) logic and (iii) probability theory. Indeed, Bayesian networks provide a semantic modelling tool which greatly facilitates the acquisition of probabilistic knowledge [79]. While multivalued dependencies allow us to decompose a database relation into two of its projections without the loss of information, conditional independencies allow us to decompose a joint probability distribution into two of its marginalizations without the loss of information [100]. Consequently, the probability of an event can be obtained, in principle, by appropriate marginalizations of the joint probability distribution. It has been shown that the associated $R$-implication problems of multivalued dependencies and conditional independencies coincide [100]. Our results in this paper show that fixing a probability space of discrete variables is not an assumption that is necessary for utilising conditional independencies. This provides a further motivation for the study of multivalued dependencies, in particular in view of the recent interest in probabilistic databases [24,87].

Contributions. In this paper we will characterise the notion of MVD implication in undetermined universes [17] from a logical, proof-theoretical and algebraic perspective. These findings extend several important results from fixed universes to
the undetermined context. That is, for all these results the assumption of having a fixed underlying universe can be dropped.

More specifically, our characterisations can be summarised as follows:

- For all fixed universes $R$, the $R$-implication of MVDs $X \rightarrow Y$ over $R = XYZ$ is equivalent to the logical implication of Boolean formulae of the form $\bigwedge\{A': A \in X\} \Rightarrow (\bigwedge\{A': A \in Y\} \lor \bigwedge\{A': A \in Z\})$ where $A'$ denotes the propositional variable that corresponds to the attribute $A \in R$ [82]. Our first main contribution establishes the propositional fragment that is equivalent to MVD implication over undetermined universes. Roughly, the MVD $X \rightarrow Y$ corresponds to a formula $\bigwedge\{A': A \in X\} \Rightarrow \bigwedge\{A': A \in Y\}$. A truth assignment $\theta$ is a model of the latter formula if (i) $\theta$ assigns truth values to variables that correspond to attributes in $XYZ$ and $Z$ denotes any finite set of variables) and (ii) $\theta$ is a model of the formula $\bigwedge\{A': A \in X\} \Rightarrow (\bigwedge\{A': A \in Y\} \lor \bigwedge\{A': A \in Z\})$ under the usual interpretation of Boolean propositional logic [32]. Let $\Sigma' \cup \{\varphi'\}$ denote the set of formulae that correspond to the finite set $\Sigma \cup \{\varphi\}$ of MVDs. We say that $\Sigma'$ logically implies $\varphi'$ if every truth assignment to at least all the variables that occur in $\Sigma' \cup \{\varphi'\}$ is a model of $\varphi'$ whenever it is a model of all formulae in $\Sigma'$. We show that there is a counter-example relation $r$ to the implication of $\varphi$ by $\Sigma$ in undetermined universes if and only if there is a truth assignment $\theta_r$ that is a counter-example to the logical implication of $\varphi'$ by $\Sigma'$.

- In fact, we show that there is a counter-example relation $r$ to the implication of $\varphi$ by $\Sigma$ in undetermined universes if and only if there is a 2-tuple counter-example subrelation of $r$ to the implication of $\varphi$ by $\Sigma$ in undetermined universes. The counter-example truth assignment $\theta_r$ assigns true to precisely those variables $A'$ that correspond to attributes $A$ on which the two tuples of the 2-tuple subrelation of $r$ agree. These results extend the correspondences that have been established in the context of a fixed universe [82]. The difference here is that the existence of a counter-example relation is no longer limited to truth assignments to a pre-determined set of variables.

We establish finite axiomatisations of the new propositional fragment, and an upper time bound of $O(\log(|A'| A \in Attr(r) \cup X) \times ||\Sigma'||)$ for deciding the implication problem with instance $(\Sigma', \bigwedge\{A': A \in X\} \Rightarrow \bigwedge\{A': A \in Y\})$. In fact, our correspondence allows us to apply recent findings on the MVD implication over undetermined universes [68] to this propositional fragment.

- Let $(\Sigma, \varphi)$ denote an instance of the MVD implication problem in undetermined universes. We characterise this problem in terms of the FD implication problem $(\Sigma_{FD}, \varphi_{FD})$ and in terms of the MVD implication problem $(\Sigma, \varphi)$ over the fixed universe $R^{fix}$, which consists of all the attributes that occur in $\Sigma \cup \{\varphi\}$. Here, for an MVD $\varphi = X \rightarrow Y$ let $\varphi_{FD}$ denote the functional dependency $X \rightarrow Y$, and for a finite set $\Sigma$ of MVDs let $\Sigma_{FD}$ denote the set $\{\varphi_{FD} | \varphi \in \Sigma\}$. This characterisation is significant as the whole theory of functional and multivalued dependencies that has been developed in the context of a fixed universe becomes accessible to that of undetermined universes. It is in this sense that we will capitalise on this result when we establish our remaining characterisations.

- A functional dependency $X \rightarrow Y$ corresponds to the formula $\bigwedge\{A': A \in X\} \Rightarrow \bigwedge\{B' | B \in Y\}$, which is equivalent to a set of Boolean propositional Horn clauses [33]. By our previous characterisation, the MVD implication problem $(\Sigma, \varphi)$ is equivalent to the two implication problems (i) $(\Sigma_{FD}, \varphi'_{FD})$ and (ii) $(\Sigma', \varphi')$ with truth assignments to variables in $\{A' | A \in R^{fix}\}$. Here, $\varphi'_{FD}$ denotes the formula that corresponds to the FD $\varphi_{FD}$ and $\Sigma'_{FD}$ denotes the union over $\sigma_{FD}$ for all $\sigma_{FD} \in \Sigma_{FD}$.

- The chase offers a convenient proof-theoretical tool to decide the $R$-implication problem for a broad class of data dependencies [28,70,71,91], e.g. when $\Sigma \cup \{\varphi\}$ consists of a set of functional and join dependencies [3] (multivalued dependencies are subsumed by join dependencies). This is particularly interesting since the class of join dependencies does not enjoy a finite ground axiomatisation [80], even though Gentzen-style axiomatisations do exist [13,84]. If $\Sigma$ consists of a functional dependency and a join dependency, and $\varphi$ denotes a join dependency, then it is NP-complete to decide whether $\Sigma$ $R$-implies $\varphi$ [71]. However, if $\Sigma$ consists of a set of functional and join dependencies, and $\varphi$ denotes either a functional or a multivalued dependency, then the chase runs in time $O(|R| \times ||\Sigma||)$ [71]. The chase has also considerable applications in the context of data exchange [36], query optimisation [1,29] and view maintenance [57,62].

We will combine the chase for deciding the $R^{fix}$-implication of functional dependencies with the chase for deciding the $R^{fix}$-implication of multivalued dependencies in order to obtain an algorithm $\text{Chase}(\Sigma, \varphi)$ for deciding the implication problem $(\Sigma, \varphi)$ of multivalued dependencies in undetermined universes. We derive an upper time bound of $O(|R^{fix}| \times ||\Sigma||)$. An immediate question for future work is how this chase can be extended to decide the implication of functional and join dependencies in undetermined universes.

- Our last characterisation of the MVD implication problem in undetermined universes is in terms of closed attribute sets. An attribute set is closed with respect to an FD $X \rightarrow Y$ if the attribute set contains all the attributes in $Y$, whenever it contains all the attributes in $X$. Moreover, an attribute set is closed with respect to an MVD $X \rightarrow Y$ over $R = XYZ$ if the attribute set contains all the attributes in $Y$ or the attribute set contains all the attributes in $Z$, whenever it contains all the attributes in $X$. We show that the finite MVD set $\Sigma$ implies the MVD $\varphi$ in undetermined universes precisely when the following two conditions are satisfied: (i) every attribute subset of $R^{fix}$ that is closed with respect to all members of $\Sigma_{FD}$ is also closed with respect to $\varphi_{FD}$, and (ii) every attribute subset of $R^{fix}$ that is closed with respect to all members of $\Sigma$ is also closed with respect to $\varphi$. This characterisation extends a result from the context of fixed universes [45].
Let $\Sigma \cup \{\varphi\}$ denote a finite set of multivalued dependencies. Then the following are equivalent:

1. $\Sigma$ implies $\varphi$ in undetermined universes,
2. $\Sigma$ implies $\varphi$ in the world of 2-tuple relations over undetermined universes,
3. $\Sigma'$ logically implies $\varphi'$,
4. $\Sigma_{FD}$ $R^2$-implies $\varphi_{FD}$ and $\Sigma$ $R^2$-implies $\varphi$,
5. $\Sigma_{FD}$ logically implies $\varphi'_{FD}$ and $\Sigma'$ logically implies $\varphi'$ with truth assignments over $\{A' \mid A \in R^2\}$,
6. Choose($\Sigma, \varphi$) returns ‘Yes’,
7. Every attribute subset of $R^2$ that is closed with respect to all members of $\Sigma_{FD}$ is also closed with respect to $\varphi_{FD}$, and every attribute subset of $R^2$ that is closed with respect to all members of $\Sigma'$ is also closed with respect to $\varphi'$.

The main contributions of this paper are summarised in Table 1.

### Table 1
Characterisations of MVD implication over undetermined universes.

<table>
<thead>
<tr>
<th>Characterisation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$\Sigma$ implies $\varphi$ in undetermined universes</td>
</tr>
<tr>
<td>2.</td>
<td>$\Sigma$ implies $\varphi$ in the world of 2-tuple relations over undetermined universes</td>
</tr>
<tr>
<td>3.</td>
<td>$\Sigma'$ logically implies $\varphi'$</td>
</tr>
<tr>
<td>4.</td>
<td>$\Sigma_{FD}$ $R^2$-implies $\varphi_{FD}$ and $\Sigma$ $R^2$-implies $\varphi$</td>
</tr>
<tr>
<td>5.</td>
<td>$\Sigma_{FD}$ logically implies $\varphi'_{FD}$ and $\Sigma'$ logically implies $\varphi'$ with truth assignments over ${A' \mid A \in R^2}$</td>
</tr>
<tr>
<td>6.</td>
<td>Choose($\Sigma, \varphi$) returns ‘Yes’</td>
</tr>
<tr>
<td>7.</td>
<td>Every attribute subset of $R^2$ that is closed with respect to all members of $\Sigma_{FD}$ is also closed with respect to $\varphi_{FD}$, and every attribute subset of $R^2$ that is closed with respect to all members of $\Sigma'$ is also closed with respect to $\varphi'$</td>
</tr>
</tbody>
</table>

### Organisation
The paper is structured as follows. Section 2 defines basic notions from the relational model of data, the concept of a multivalued dependency, the semantic notion of $R$-implication, and the syntactic notion of inference. Finally, the correspondence between $R$-implication of MVDs and a Boolean propositional fragment is summarised. In Section 3 we present the basic concepts for multivalued dependencies in undetermined universes, and list some previous results on the associated implication problem. The remainder of the paper follows closely the order of the results in Table 1. The propositional fragment that corresponds to MVD implication is developed in Section 4, and the equivalence between 1, 2 and 3 is established. We also establish axiomatisations and an upper time bound for deciding implication in this fragment. In Section 5 we show the equivalence between 1, 4 and 5. The procedure of the chase for deciding implication of functional and multivalued dependencies in fixed universes is summarised in Section 6. We capitalise on the previous characterisation between 1 and 4, and establish a chase for deciding the implication of MVDs over undetermined universes. This shows the equivalence between 1 and 6. The remaining equivalence between 1 and 7 is established in Section 7. The paper concludes and lists some directions of future work in Section 8.

### 2. Multivalued dependencies in fixed universes

In this section we will fix notions and notations fundamental to the original definition of multivalued dependencies in the relational model of data. In particular, we will summarise the correspondence between the $R$-implication of multivalued dependencies and the logical implication of a fragment in Boolean propositional logic [82].

Let $\mathcal{A} = \{A_1, A_2, \ldots\}$ be a countably infinite set of symbols. In the context of the relational model of data, the elements of $\mathcal{A}$ are called attributes. One may think of an attribute as a column header in a table. A relation schema (or universe) is a finite subset of $\mathcal{A}$, usually denoted by $R$. Each attribute $A$ of a relation schema $R$ is associated with a countably infinite domain $\text{dom}(A)$. The domain $\text{dom}(A)$ represents the set of possible values that might occur in the column of a table with header $A$. Following a common notation from relational database theory, if $X$ and $Y$ denote finite sets of attributes, then we may write $XY$ instead of the set union $X \cup Y$. If $X = \{A_1, \ldots, A_m\}$, then we may write $A_1 \cdots A_m$ for $X$. In particular, we may write simply $A$ to represent the singleton $\{A\}$.

A tuple over the relation schema $R$ ($R$-tuple or simply tuple, if $R$ is understood) is a function $t : R \rightarrow \bigcup_{A \in R} \text{dom}(A)$ such that for all $A \in R$ we have $t(A) \in \text{dom}(A)$.

A relation over $R$ is a finite set of tuples over $R$. If a relation $r$ is given without reference to its relation schema $R$ over which it is defined, then we denote $r$ also by $\text{Attr}(r)$, i.e., the set of attributes over which $r$ is defined. One may think of a relation over a relation schema as a table in which each element of the relation represents a row of the table. The attributes of the relation schema form the properties by which every single row of every possible table with these attributes as column headers is specified.

Let $r[X] = \{t \in r \mid t \in r\}$ denote the projection of the relation $r$ over $R$ onto $X \subseteq R$. For $X, Y \subseteq R$, $r_1 \subseteq \text{dom}(X)$ and $r_2 \subseteq \text{dom}(Y)$ let $r_1 \bowtie r_2 = \{t \in \text{dom}(XY) \mid \exists r_1 \in r_1, t_2 \in r_2 \text{ with } t[X] = t_1[X] \text{ and } t[Y] = t_2[Y]\}$ denote the natural join of $r_1$ and $r_2$. Note that the $0$-ary relation $\{()\}$ is the projection $r[\emptyset]$ of a non-empty relation $r$ onto $\emptyset$ as well as the left and right identity of the natural join operator.

#### 2.1. Semantic implication of multivalued dependencies

A multivalued dependency (MVD) [26,34,102] over the relation schema $R$ is an expression $X \rightarrow Y$ where $X, Y \subseteq R$. A relation $r$ over $R$ is said to satisfy the MVD $X \rightarrow Y$ if and only if for all $t_1, t_2 \in r$ with $t_1[X] = t_2[X]$ there is some $t \in r$ with $t[XY] = t_1[XY]$ and $t[X(R \rightarrow Y)] = t_2[X(R \rightarrow Y)]$. If $\Sigma$ denotes a set of multivalued dependencies over $R$, then we say that a relation satisfies $\Sigma$, if the relation satisfies every member of $\Sigma$. If a relation does not satisfy a multivalued dependency, then we also say that the relation violates the multivalued dependency.
Informally, the relation \( r \) satisfies \( X \rightarrow Y \) when a value on \( X \) determines the set of values on \( Y \) independently of the set of values on \( R - Y \). This actually suggests that the relation schema \( R \) is overloaded in the sense that it carries two independent facts \( XY \) and \( X(R - Y) \). More precisely, Fagin has been shown [34] that MVDs “provide a necessary and sufficient condition for a relation to be decomposable into two of its projections without loss of information (in the sense that the original relation is guaranteed to be the natural join of the two projections)”. This means that \( r \) satisfies \( X \rightarrow Y \) if and only if \( r = r[XY] \Rightarrow r[X(R - Y)] \). This characteristic of MVDs is fundamental to relational database design and the 4NF condition [34]. A lot of research has therefore been devoted to studying the behaviour of these dependencies.

For the design of a relation schema dependencies are normally specified as semantic constraints on the relations which are intended to be instances of the schema. That is, only those relations are permitted which satisfy all of the dependencies that have been specified. Consequently, the specification of such dependencies restricts database instances to those which are considered meaningful to the application at hand.

**Example 3.** Consider again the relation schema \( \{ \text{Employee}, \text{Child}, \text{Salary} \} \). The multivalued dependency \( \text{Employee} \rightarrow \text{Child} \) expresses the fact that each employee name determines the set of names of the employee’s children independently of the employee’s salary.

The two-tuple relation \( r \) of Example 1 does not satisfy the multivalued dependency \( \text{Employee} \rightarrow \text{Child} \). Consequently, if this MVD is specified over \( \{ \text{Employee}, \text{Child}, \text{Salary} \} \), then \( r \) is excluded from the set of valid instances of the schema.

A dependency \( \varphi \) is said to be specified implicitly by a set \( \Sigma \) of dependencies, whenever every relation that satisfies all the dependencies in \( \Sigma \) also satisfies \( \varphi \). In order to emphasise the dependence of this notion of implication on the underlying relation schema \( R \), we refer to \( R \)-implication.

**Definition 1.** Let \( R \) be a relation schema, and let \( \Sigma \cup \{ \varphi \} \) be a set of multivalued dependencies over \( R \). Then \( \Sigma \) \( R \)-implies \( \varphi \), denoted by \( \Sigma \models_R \varphi \), if and only if every relation \( r \) over \( R \) that satisfies \( \Sigma \) also satisfies \( \varphi \).

**Example 4.** Consider the MVD \( \text{Employee} \rightarrow \text{Child} \) over relation schema \( R = \{ \text{Employee}, \text{Child}, \text{Salary} \} \). This MVD \( R \)-implies the MVD \( \text{Employee} \rightarrow \text{Salary} \).

### 2.2. Syntactic inference of multivalued dependencies

In order to determine all logical consequences of a set of MVDs one can use the inference rules in Table 2 [77]. These inference rules have the form

| \( \frac{X \rightarrow Y \subseteq X}{X \rightarrow Y} \) (reflexivity, \( \mathcal{R} \)) & \( \frac{X \subseteq X \rightarrow Y}{X \rightarrow Y} \) (augmentation, \( \mathcal{A} \)) & \( \frac{X \rightarrow Y \rightarrow Z}{X \rightarrow Y} \) (pseudo-transitivity, \( T \)) |
| \( \frac{X \rightarrow Y \rightarrow Z}{X \rightarrow Y} \) (additive transitivity, \( T^+ \)) & \( \frac{X \rightarrow Y \rightarrow Z}{X \rightarrow Y} \) (subset, \( \mathcal{S} \)) & \( \frac{X \rightarrow Y \rightarrow Z}{X \rightarrow Y} \) (\( R \)-complementation, \( \mathcal{C}_R \)) |
| \( \frac{X \rightarrow Y \rightarrow Z}{X \rightarrow Y} \) (intersection, \( \mathcal{I} \)) |

and inference rules without a premise are called *axioms*. Intuitively, an application of such a rule mechanically infers the expression in the conclusion of the rule, given that the expressions in the premise of the rule have already been inferred previously and given that the expressions in the premise and conclusion of the rule also meet the condition of the rule.

Let \( \Sigma \cup \{ \varphi \} \) be a set of MVDs over the relation schema \( R \). Furthermore, we use \( \Theta \) to denote a set of inference rules. Within this paper, we only consider the inference rules from Table 2. The notion of syntactical inference \( (\models_\Theta) \) with respect to a set \( \Theta \) of inference rules can be defined analogously to the notion in the relational data model [1, pp. 164–168]. That is, a finite sequence \( \gamma' = \{ \gamma_1, \ldots, \gamma_l \} \) of MVDs is called an *inference from* \( \Sigma \) *by* \( \Theta \) if every \( \gamma_i \) is either an element of \( \Sigma \) or is obtained by applying one of the rules of \( \Theta \) to appropriate elements of \( \{ \gamma_1, \ldots, \gamma_{i-1} \} \). We say that the inference \( \gamma \) infers \( \gamma_l \), i.e., the last element of the sequence \( \gamma' \), and write \( \Sigma \models_\Theta \gamma_l \). For a set \( \Sigma \) of MVDs over a relation schema \( R \), let \( \Sigma_\Theta^+ = \{ \varphi \mid \Sigma \models_\Theta \varphi \} \) be its *syntactic closure* under inferences by \( \Theta \). An inference rule is called *\( R \)-sound* if the set of dependencies in the premise of the rule \( R \)-implies the dependency in the conclusion under the condition of the rule. It is well known that all the rules above are \( R \)-sound for all \( R \) [77]. The set \( \Theta \) is called *\( R \)-sound* for the \( R \)-implication of MVDs if and only if for every set \( \Sigma \) of MVDs over the relation schema \( R \) we have \( \Sigma_\Theta^+ \subseteq \Sigma_{\Theta}^+ \).
is said to be an axiomatisation for the $R$-implication of MVDs if for all relation schemata $R$, $\Xi$ is both $R$-sound and $R$-complete for the $R$-implication of MVDs. An axiomatisation $\Xi$ is said to be finite if the set $\Xi$ is finite. For instance, the set $\Xi = \{ [R, A, T, C, h, D, t] \}$ of inference rules forms an axiomatisation for the $R$-implication of multivalued dependencies [11,16,77]. Finally, the implication problem for multivalued dependencies is the problem of deciding whether for an arbitrary relation schema $R$ and an arbitrary set $\Sigma \cup \{ \phi \}$ of multivalued dependencies over $R$, it is true that $\Sigma \models_\Xi \phi$ holds.

**Example 5.** Let $R = \{ \text{Employee}, \text{Child, Salary} \}$, and let $\Sigma$ consist of the single MVD $\text{Employee} \rightarrow \text{Child}$. The MVD $\phi = \text{Employee} \rightarrow \text{Employee, Salary}$ can be inferred from $\Sigma$ by a single application of the $R$-complementation rule to the MVD $\text{Employee} \rightarrow \text{Child}$. 

### 2.3 A fragment of propositional logic

Let $\mathcal{V}$ denote a countably infinite set of propositional variables, and let $\mathbb{N}$ denote the set of non-negative integers. For a finite subset $\mathcal{V}(R) \subseteq \mathcal{V}$ the set $\mathcal{F}[\mathcal{V}(R)]$ of formulae over $\mathcal{V}(R)$ is the set

$$\{(A_1' \land \ldots \land A_n') \Rightarrow (B_1' \land \ldots \land B_m') \mid A_1', \ldots, A_n', B_1', \ldots, B_m' \in \mathcal{V}(R); \ l, m \in \mathbb{N} \}.$$ 

In what follows we assume that the conjunction $\land$ binds stronger than the material implication $\Rightarrow$. Therefore, we denote formulae in $\mathcal{F}[\mathcal{V}(R)]$ by $A_1' \land \ldots \land A_n' \Rightarrow B_1' \land \ldots \land B_m'$. Let $true$ and $false$ denote the Boolean truth values. We call a function $\theta : \mathcal{V}(R) \rightarrow \{true, false\}$ a truth assignment over $\mathcal{V}(R)$.

We will now extend $\theta$ to a function $\Theta : \mathcal{F}[\mathcal{V}(R)] \rightarrow \{true, false\}$. For a formula $\phi = A_1' \land \ldots \land A_n' \Rightarrow B_1' \land \ldots \land B_m' \in \mathcal{F}[\mathcal{V}(R)]$ let $n \in \mathbb{N}$ and $(C_1', \ldots, C_n') \subseteq \mathcal{V}(R)$ be such that $\{ A_1', \ldots, A_n', B_1', \ldots, B_m', C_1', \ldots, C_n' \} = \mathcal{V}(R)$. We define $\Theta(\phi)$ to be true if and only if for some $i \in \{1, \ldots, n\}$ we have $\theta(A_i') = false$, or for all $j = 1, \ldots, m$ we have $\theta(B_j') = true$ or for all $k = 1, \ldots, n$ we have $\theta(C_k') = true$. We say that $\phi'$ is true (or $\phi'$ holds) under $\theta$, or that $\theta$ is a model of $\phi'$, if $\Theta(\phi') = true$. In the case where $l = 0$, we also write $true \Rightarrow B_1' \land \ldots \land B_m'$, and similarly for $m = 0$. Notice that $\phi' \in \mathcal{F}[\mathcal{V}(R)]$ is a tautology, i.e. holds under all truth assignments over $\mathcal{V}(R)$, if $m = 0$ or $n = 0$. We call $\theta$ a model of a finite set $\Sigma'$ of formulae over $\mathcal{V}(R)$, if $\theta$ is a model of every element of $\Sigma'$. If $\theta$ is not a model of $\phi'$, then we also say that $\theta$ violates $\phi'$. Occasionally, we also write $\theta \not\models \phi'$ as short for the conjunction $\mathcal{V}_1 \land \cdots \land \mathcal{V}_n$ where $\mathcal{X} = \{ \mathcal{V}_1', \ldots, \mathcal{V}_n' \} \subseteq \mathcal{V}$. Note that $\theta$ is a model of $\phi'$ precisely if $\theta$ is a model of the Boolean propositional formula [32]

$$(A_1' \land \cdots \land A_n') \Rightarrow ((B_1' \land \cdots \land B_m') \lor (C_1' \land \cdots \land C_n')),$$

i.e., we have adapted the usual interpretation of the Boolean connectives $\land, \lor, \Rightarrow$.

**Example 6.** Let $\mathcal{V}(R) = \{ \mathcal{V}, \mathcal{C}, \mathcal{S} \}$. The formula $\mathcal{V} \Rightarrow \mathcal{C}$ is violated by the truth assignment $\theta$ that assigns $true$ to $\mathcal{V}$ and $false$ to $\mathcal{C}$ and $\mathcal{S}$. However, the truth assignment $\theta'$ that assigns $true$ to $\mathcal{V}$ and $\mathcal{S}$ and $false$ to $\mathcal{C}$ is a model of $\mathcal{V} \Rightarrow \mathcal{C}$.

For a set $\Sigma' \cup \{ \phi' \} \subseteq \mathcal{F}[\mathcal{V}(R)]$ we say that $\Sigma' \mathcal{V}(R)$-implies $\phi'$, denoted by $\Sigma' \models_{\mathcal{V}(R)} \phi'$, whenever every truth assignment over $\mathcal{V}(R)$ that is a model of $\Sigma'$ is also a model of $\phi'$. That is, there is no counter-example truth assignment that is a model of $\Sigma'$ and violates $\phi'$.

**Example 7.** Let $\mathcal{V}(R) = \{ \mathcal{V}, \mathcal{C}, \mathcal{S} \}$. Let $\Sigma'$ consist of the single formula $\mathcal{V} \Rightarrow \mathcal{C}$, and let $\phi'$ denote the formula $\mathcal{V} \Rightarrow \mathcal{V} \land \mathcal{S}$. It follows that $\Sigma' \mathcal{V}(R)$-implies $\phi'$.

### 2.4 The correspondence

Let $R$ be some relation schema, and let $\Sigma \cup \{ \phi \}$ be a set of MVDs over $R$. We say that $\phi$ is $R$-implied by $\Sigma$ in the world of 2-tuple relations, denoted by $\Sigma \models_\Xi^\phi \phi$, if for all 2-tuple relations $r$ over $R$, the MVD $\phi$ is satisfied by $r$ whenever $\Sigma$ (i.e. all MVDs in $\Sigma$) is satisfied by $r$ [33]. That is, there is no counter-example 2-tuple relation that satisfies all the MVDs in $\Sigma$ but violates $\phi$. Note that if $\phi$ is $R$-implied by $\Sigma$, then $\phi$ is $R$-implied by $\Sigma$ in the world of 2-tuple relations, but the converse is not obvious.

Let $\phi : R \rightarrow \mathcal{V}(R)$ denote a bijection between a relation schema $R$ and a (finite) set $\mathcal{V}(R)$ of propositional variables. For an attribute $A \in R$ we usually simply write $A'$ instead of $\phi(A)$. We will now extend $\phi$ to a function $\Phi$ that maps an MVD $\phi$ over $R$ to a formula over $\mathcal{V}(R)$. Let $\phi$ denote the multivalued dependency $A_1, \ldots, A_n \rightarrow B_1, \ldots, B_m$ over $R$. The function $\Phi$ applied to $\phi$ is the formula $A_1' \land \cdots \land A_n' \Rightarrow B_1' \land \cdots \land B_m'$, denoted by $\phi'$. Note that $\phi' \in \mathcal{F}[\mathcal{V}(R)]$. We call $\phi'$ the formula that corresponds to $\phi$. Instead of writing $\Phi(\phi)$ we usually write $\phi'$, and instead of writing $\{ \sigma' \mid \sigma \in \Sigma \}$ we usually simply write $\Sigma'$. We call $\Sigma'$ the set of formulae over $\mathcal{V}(R)$ that corresponds to $\Sigma \cup \{ \phi \}$. Equivalently are:

**Theorem 1.** (See [82]) Let $R$ be some relation schema, and let $\Sigma \cup \{ \phi \}$ be a set of multivalued dependencies over $R$. Let $\Sigma' \cup \{ \phi' \}$ be the set of formulae over $\mathcal{V}(R)$ that corresponds to $\Sigma \cup \{ \phi \}$. Equivalent are:
1. \( \Sigma \models \varphi \),
2. \( \Sigma \not\models \varphi \),
3. \( \Sigma' \models \forall(R) \varphi' \).

The equivalence of 1 and 2 follows from the observation that if there is a counter-example relation \( r \) that satisfies \( \Sigma \) but violates \( \varphi \), then there is always a 2-tuple subrelation \( s \) of \( r \) that satisfies \( \Sigma \) but violates \( \varphi \) [82, Lemma 9]. For the equivalence of 2 and 3, a correspondence is established between 2-tuple counter-example relations \( s \) that satisfy \( \Sigma \) and violate \( \varphi \) and truth assignments \( \theta_i \) that are models of \( \Sigma' \) but not models of \( \varphi' \). Indeed \( \theta_i(A') = true \) precisely if the two tuples \( t_1, t_2 \in s \) agree on \( A \), i.e. if \( t_1[A] = t_2[A] \) holds.

**Example 8.** Consider Examples 5 and 7. The set \( \Sigma \cup \{ \varphi \} \) of Example 5 corresponds to \( \Sigma' \cup \{ \varphi' \} \) of Example 7. Consider now the set \( \Sigma' \cup \{ true \Rightarrow V_E \} \) that corresponds to the set \( \Sigma \cup \{ \theta \rightarrow \text{Employee} \} \). The relation \( s \)

<table>
<thead>
<tr>
<th>Employee</th>
<th>Child</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homer</td>
<td>Lisa</td>
<td>4500</td>
</tr>
<tr>
<td>Marge</td>
<td>Lisa</td>
<td>5000</td>
</tr>
</tbody>
</table>

forms a counter-example for the \( R \)-implication of \( \theta \rightarrow \text{Employee} \) by \( \Sigma \). Similarly, the truth assignment \( \theta_i \) that assigns \( true \) to \( V_C \) and \( false \) to \( V_E \) and \( V_S \) forms a counter-example for the \( \forall(R) \)-implication of \( true \Rightarrow V_E \) by \( \Sigma' \). Notice that \( \theta_i \) assigns \( true \) to precisely those variables that correspond to attributes on which the two tuples of \( s \) match.

### 3. Multivalued dependencies in undetermined universes

In this section we will briefly review the framework for defining multivalued dependencies over undetermined universes, due to Biskup [17]. The motivation of this framework has already been discussed in the introduction of this paper. We will summarise the main notions and some of the results on the associated implication problem [17,68]. In the following section we will characterise the notion of MVD implication in undetermined universes by capitalising on the results presented in this section.

A multivalued dependency over \( \mathcal{A} \) is a syntactic expression \( X \rightarrow Y \) with finite subsets \( X, Y \subseteq \mathcal{A} \). For an MVD \( X \rightarrow Y \) over \( \mathcal{A} \), denoted by \( \varphi \), we write \( \text{Attr}(\varphi) \) for the set \( XY \). For a finite set \( \Sigma \) of MVDs we write \( \text{Attr}(\Sigma) \) for the union of \( \text{Attr}(\varphi) \) over all \( \varphi \in \Sigma \). A relation over \( \mathcal{A} \) is defined as a finite set \( r \) of tuples with the same domain, i.e., the same finite subset \( \text{Attr}(r) \subseteq \mathcal{A} \).

That is, each tuple of \( r \) is a function \( t: \text{Attr}(r) \rightarrow \bigcup_{A \subseteq \text{Attr}(r)} \text{dom}(A) \) such that for all \( A \in \text{Attr}(r) \) we have \( t(A) \in \text{dom}(A) \). The MVD \( X \rightarrow Y \) over \( \mathcal{A} \) is satisfied by the relation \( r \) over \( \mathcal{A} \) if and only if \( XY \subseteq \text{Attr}(r) \) and \( r = r[XY] \Rightarrow r[X \cup (\text{Attr}(r) - Y)] \).

Biskup introduced the following notion of semantic implication [17].

**Definition 2.** Let \( \Sigma \cup \{ \varphi \} \) be a finite set of multivalued dependencies over \( \mathcal{A} \). The set \( \Sigma \) implies \( \varphi \), denoted by \( \Sigma \models_\mathcal{A} \varphi \), if and only if for every relation \( r \) over \( \mathcal{A} \) with \( \text{Attr}(\Sigma \cup \{ \varphi \}) \subseteq \text{Attr}(r) \) the following holds: if \( r \) satisfies \( \Sigma \), then \( r \) satisfies \( \varphi \).

In this definition, the underlying relation schema is left undetermined. The only requirement is that the MVDs must apply to the relations. If \( \text{Attr}(\Sigma \cup \varphi) \subseteq R \), then it follows immediately that \( \Sigma \) \( R \)-implies \( \varphi \) whenever \( \Sigma \) implies \( X \rightarrow Y \). The converse, however, is false [17] as the following example demonstrates.

**Example 9.** For \( R = \{ \text{Employee, Child, Salary} \} \) and \( \Sigma = \{ \text{Employee} \rightarrow \text{Child} \} \) we have that \( \Sigma \) \( R \)-implies \( \text{Employee} \rightarrow \text{Salary} \). However, \( \Sigma \) does not imply \( \text{Employee} \rightarrow \text{Salary} \). Consider for instance the following relation \( r \) over \( \mathcal{A} \) where \( \text{Attr}(r) = \{ \text{Employee, Child, Salary, Year} \} \).

<table>
<thead>
<tr>
<th>Employee</th>
<th>Child</th>
<th>Salary</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homer</td>
<td>Lisa</td>
<td>4000</td>
<td>2008</td>
</tr>
<tr>
<td>Homer</td>
<td>Lisa</td>
<td>4500</td>
<td>2009</td>
</tr>
</tbody>
</table>

The two relations \( r[\text{Employee, Child}] \) and \( r[\text{Employee, Salary, Year}] \)

<table>
<thead>
<tr>
<th>Employee</th>
<th>Child</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homer</td>
<td>Lisa</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Employee</th>
<th>Salary</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homer</td>
<td>4000</td>
<td>2008</td>
</tr>
<tr>
<td>Homer</td>
<td>4500</td>
<td>2009</td>
</tr>
</tbody>
</table>
indicate that \( r \) violates \( \text{Employee} \implies \text{Salary} \). Consequently, \( \Sigma \) does not imply \( \text{Employee} \implies \text{Salary} \).

The definitions of soundness, completeness, axiomatisation and the implication problem are simply adapted to the context of undetermined universes by dropping the reference to the underlying relation schema \( R \) from the corresponding definitions in the context of fixed universes. While the singletons \( R, A, T, T^*, S, U, D, I \) are all sound, the \( R \)-complementation rule \( C_R \) is \( R \)-sound, but not sound [17].

Biskup [17] proves that the set \( \mathcal{S}_1 = \{ R, A, T, T^*, S \} \) forms a finite axiomatisation for the implication of MVDs. The major proof argument shows that for every inference of an MVD \( X \implies Y \) from \( \Sigma \) by \( \mathcal{S}_1 \) there is an inference of \( X \implies Y \) from \( \Sigma \) by \( \mathcal{S}_1 \) in which the \( R \)-complementation rule \( C_R \) is applied at most once, and if it is applied, then it is applied in the last step of the inference only. This shows that

\[
X \implies Y \in \Sigma^+_{\mathcal{S}_1} \quad \text{if and only if} \quad X \implies Y \in \Sigma^+_{\mathcal{S}_1}, \quad \text{or} \quad X \implies (R - Y) \in \Sigma^+_{\mathcal{S}_1},
\]

where \( \text{Attr}(\Sigma \cup \{ X \implies Y \}) \subseteq R \). Note that \( \mathcal{S}_1 \) is almost \( R \)-complete for the \( R \)-implication of MVDs.

Moreover, all axiomatisations that are subsets of the rule set in Table 2 (without the \( R \)-complementation rule \( C_R \)) have been identified [68]. This result complements Mendelzon’s findings [74] in fixed universes.

**Theorem 2.** (See [68].) Let \( \mathcal{S} \) denote a subset of the inference rules from Table 2 without the \( R \)-complementation rule \( C_R \). Then \( \mathcal{S} \) forms an axiomatisation for the implication of multivalued dependencies in undetermined universes precisely when \( \mathcal{S} \) is a superset of at least one of the following sets: \( \mathcal{S}_1, \mathcal{S}_2 = \{ R, S, T, U \} \) or \( \mathcal{S}_3 = \{ R, S, T^*, D \} \).

Let \( \Sigma \) be a finite set of MVDs, and \( X \) some finite set of attributes over \( A \). Let \( \text{Dep}_U(X) = \{ Y \mid X \implies Y \in \Sigma^+_{\mathcal{S}_1} \} \) be the set of all attribute sets \( Y \) over \( A \) such that \( X \implies Y \) can be inferred from \( \Sigma \) by \( \mathcal{S}_1 \). The set \( X^*_U = \bigcup \text{Dep}_U(X) \) is called the scope of \( X \) with respect to \( \Sigma \) [68]. Since the union, intersection and difference rules are sound, it follows that \( \text{Dep}_U(X), \subseteq, \cup, \cap, -, \emptyset, X^*_U \) is a finite Boolean algebra, with top-element \( X^*_U \). Recall that an element \( a \in P \) of a poset \( (P, \subseteq, 0) \) with least element 0 is called an atom of \( (P, \subseteq, 0) \) if and only if \( a \neq 0 \) and every element \( b \in P \) with \( b \subseteq a \) satisfies \( b = 0 \) or \( b = a \). \( (P, \subseteq, 0) \) is called atomic if and only if for every element \( b \in P \) with \( b \neq 0 \) there is an atom \( a \in P \) with \( a \subseteq b \). In particular, every finite Boolean algebra is atomic. The dependency basis \( \text{DepB}_U(X) \) of \( X \) with respect to \( \Sigma \) is the set of all atoms of \( \text{Dep}_U(X), \subseteq, \emptyset \) [68].

It was shown [68] that the implication problem of multivalued dependencies in undetermined universes can be reduced to the implication problem of multivalued dependencies in a certain fixed universe [7,40]. This made it also possible to establish an upper time bound of \( O(\log |\text{Attr}(\Sigma) \cup X| \times \| \Sigma \|) \) for deciding the implication problem \( \Sigma \models_X X \implies Y \) in undetermined universes [68].

**Corollary 3.** (See [68].) The implication problem \( \Sigma \models_X X \implies Y \) can be decided in time \( O((1 + \min\{s, \log p\}) \cdot n) \) where \( s \) denotes the number of dependencies in \( \Sigma \), \( p \) the number of sets in \( \text{DepB}_U(X) \) that have non-empty intersection with \( Y \) and \( n \) denotes the total number of occurrences of attributes in \( \Sigma \).

### 4. The correspondence in undetermined universes

In this section, we extend the logical characterisation of MVD implication over fixed universes to that over undetermined universes. Therefore, we will first define the syntax and semantics of a fragment of Boolean propositional logic. We will then establish our first characterisation by proving that the implication of MVDs in undetermined universes corresponds exactly to the logical implication of formulae in this fragment. Subsequently, we will apply this correspondence to establish all finite axiomatisations of the propositional fragment with respect to a given set of sound inference rules. Finally, we will apply our correspondence to establish an upper time bound for deciding the implication problem of formulae in this fragment.

#### 4.1. The propositional fragment

Recall that \( \mathcal{V} \) denotes our countably infinite set of propositional variables. The set \( \mathcal{F}_\mathcal{V} \) of formulae over \( \mathcal{V} \) is the set

\[
\{ A'_1 \land \cdots \land A'_l \Rightarrow B'_1 \land \cdots \land B'_m \mid A'_1, \ldots, A'_l, B'_1, \ldots, B'_m \in \mathcal{V}; \; l, m \in \mathbb{N} \}.
\]

For \( \mathcal{V}' = A'_1 \land \cdots \land A'_l \Rightarrow B'_1 \land \cdots \land B'_m \in \mathcal{F}_\mathcal{V} \) we use \( \mathcal{V}(\mathcal{V}') \) to denote the set \( \{ A'_1, \ldots, A'_l, B'_1, \ldots, B'_m \} \). For a finite subset \( \Sigma' \subseteq \mathcal{F}_\mathcal{V} \) we use \( \mathcal{V}(\Sigma') \) to denote the union of \( \mathcal{V}(\sigma) \) over all \( \sigma \in \Sigma' \).
A truth assignment over $V$ is defined as a truth assignment over a finite subset $V(\theta) \subseteq V$, i.e., as a function $\theta : V(\theta) \to \{true, false\}$. We will now extend $\theta$ to a function $\theta : F_V \to \{true, false\}$. For a formula $\varphi' = A_1' \land \cdots \land A_l' \Rightarrow B_1' \land \cdots \land B_m' \in F_V$ we define $\theta(\varphi')$ to be true if and only if $V(\varphi') \subseteq V(\theta)$ and for some $i \in \{1, \ldots, l\}$ we have $\theta(A_i') = false$, or for all $j = 1, \ldots, m$ we have $\theta(B_j') = true$ or for all $k = 1, \ldots, n$ we have $\theta(C_k') = true$, where $V(\theta) = \{A_1', \ldots, A_l', B_1', \ldots, B_m', C_1', \ldots, C_n'\}$.

**Definition 3.** Let $\Sigma' \cup \{\varphi'\}$ denote a finite set of formulae over $V$. We say that $\Sigma'$ implies $\varphi'$, denoted by $\Sigma' \models \varphi'$, if and only if for every truth assignment $\theta$ over $V$ with $V(\Sigma' \cup \{\varphi'\}) \subseteq V(\theta)$ the following holds: if $\theta$ is a model of $\Sigma'$, then $\theta$ is a model of $\varphi'$.

In this definition, the underlying set of propositional variables is left undefined. The only requirement is that the variables of the formulae apply to the truth assignments. If $V(\Sigma' \cup \{\varphi'\}) \subseteq V(\theta)$, then it follows immediately that $\Sigma' \dagger(\theta)$-implies $\varphi'$ whenever $\Sigma'$ implies $\varphi'$. The converse, however, is false as the following example demonstrates.

**Example 10.** For $V(\Sigma) = \{V_E, V_C, V_S\}$ and $V'(\Sigma) = \{V_E \Rightarrow V_C\}$ we have that $\Sigma' \dagger(\theta)$-implies $V_E \Rightarrow V_S$. However, $\Sigma'$ does not imply $V_E \Rightarrow V_S$. Consider for instance the following truth assignment $\theta$ over $V$ with $V(\theta) = \{V_E, V_C, V_S, V_Y\}$: $\theta$ assigns $V_E$ and $V_C$ the truth value true, and assigns $V_S$ and $V_Y$ the truth value false. It is easy to observe that $\theta$ is a model of $V_E \Rightarrow V_C \land (V_S \land V_Y)$ but not a model of $V_E \Rightarrow V_S \lor (V_C \land V_Y)$. Consequently, $\Sigma'$ does not imply $\varphi'$.

### 4.2. The correspondence

Let $\Sigma \cup \{\varphi\}$ be a finite set of MVVDs over $\Delta$. We say that $\varphi$ is implied by $\Sigma$ in the world of 2-tuple relations, denoted by $\Sigma \vdash_{\Delta} \varphi$, if for all 2-tuple relations $r$ over $\Delta$ with $Attr(\Sigma \cup \{\varphi\}) \subseteq Attr(r)$ the satisfaction of $\Sigma$ by $r$ implies the satisfaction of $\varphi$ by $r$. That is, there is no counter-example 2-tuple relation over $\Delta$ such that (i) $Attr(\Sigma \cup \{\varphi\}) \subseteq Attr(r)$, (ii) $r$ satisfies $\Sigma$, and (iii) $r$ violates $\varphi$. Note that if $\varphi$ is implied by $\Sigma$, then $\varphi$ is implied by $\Sigma$ in the world of 2-tuple relations.

Let $\phi : \Delta \to V$ denote a bijection between the attribute set $\Delta$ and the set $V$ of propositional variables. For an attribute $A \in \Delta$ we usually simply write $A'$ instead of $\phi(A)$. We will now extend $\phi$ to a function $\Phi$ that maps an MVD over $\Delta$ to a formula over $V$. Let $\varphi$ denote the multivalued dependency $A_1 \rightarrow\leftarrow \ldots \rightarrow\leftarrow A_l \rightarrow B_1 \rightarrow\ldots \rightarrow\rightarrow B_m$ over $\Delta$. The function $\Phi$ applied to $\varphi$ is the formula $A_1' \land \cdots \land A_l' \Rightarrow B_1' \land \cdots \land B_m'$ denoted by $\varphi'$. Note that $\varphi' \in F_V$. We call $\varphi'$ the formula that corresponds to $\varphi$. For a finite set $\Sigma \cup \{\varphi\}$ of MVVDs over $\Delta$ we write $\varphi'$ instead of writing $\Phi(\varphi)$, and instead of writing $\{\varphi' | \sigma \in \Sigma\}$ we usually simply write $\Sigma'$. We call $\Sigma'$ the set of formulae over $V$ that correspond to the set of MVVDs $\Sigma$ over $\Delta$.

**Theorem 4.** Let $\Sigma \cup \{\varphi\}$ be a finite set of multivalued dependencies over $\Delta$. Let $\Sigma' \cup \{\varphi'\}$ be the set of formulae over $V$ that correspond to $\Sigma \cup \{\varphi\}$. Equivalent are:

1. $\Sigma \vdash_{\Delta} \varphi$.
2. $\Sigma' \vdash_{\Sigma} \varphi'$.
3. $\Sigma' \vdash_{\Sigma} \varphi'$.

**Proof.** According to Definition 2 we have that $\Sigma \vdash_{\Delta} \varphi$ if and only if for every relation $r$ over $\Delta$ with $Attr(\Sigma \cup \{\varphi\}) \subseteq Attr(r)$ the satisfaction of $\Sigma$ by $r$ implies the satisfaction of $\varphi$ by $r$. That is, $\Sigma \vdash_{\Delta} \varphi$ if and only if for every relation $r$ over $\Delta$ with $Attr(\Sigma \cup \{\varphi\}) \subseteq Attr(r)$ we have that $\Sigma$ $Attr(r)$-implies $\varphi$. According to Theorem 1 that means that $\Sigma \vdash_{\Delta} \varphi$ if and only if for every 2-tuple relation $r$ over $\Delta$ with $Attr(\Sigma \cup \{\varphi\}) \subseteq Attr(r)$ we have that $\Sigma$ $Attr(r)$-implies $\varphi$. This, however, means by Theorem 1 that $\Sigma \vdash_{\Delta} \varphi$ if and only if $\Sigma \vdash_{\Sigma} \varphi$. This shows the equivalence between 1 and 2.

According to Definition 2 we have that $\Sigma \vdash_{\Delta} \varphi$ if and only if for every relation $r$ over $\Delta$ with $Attr(\Sigma \cup \{\varphi\}) \subseteq Attr(r)$ the satisfaction of $\Sigma$ by $r$ implies the satisfaction of $\varphi$ by $r$. That is, $\Sigma \vdash_{\Delta} \varphi$ if and only if for every relation $r$ over $\Delta$ with $Attr(\Sigma \cup \{\varphi\}) \subseteq Attr(r)$ we have that $\Sigma$ $Attr(r)$-implies $\varphi$. According to Theorem 1 that means that $\Sigma \vdash_{\Delta} \varphi$ if and only if for every truth assignment $\theta$ over $V$ with $V(\Sigma' \cup \{\varphi'\}) \subseteq V(\theta)$ we have that $\Sigma'$ $V(\theta)$-implies $\varphi'$. This, however, means by Theorem 1 that $\Sigma \vdash_{\Delta} \varphi$ if and only if $\Sigma' \vdash_{\Sigma} \varphi'$. This shows the equivalence between 1 and 3.

**Corollary 5.** Let $\Sigma \cup \{\varphi\}$ be a finite set of multivalued dependencies over $\Delta$. If $\Sigma$ does not imply $\varphi$, then there is a 2-tuple relation $r$ over $\Delta$ such that $Attr(\Sigma \cup \{\varphi\}) \subseteq Attr(r)$, $r$ satisfies $\Sigma$ and $r$ violates $\varphi$.

**Example 11.** Consider Examples 9 and 10 again. The set $\Sigma' \cup \{\varphi'\}$ in Example 10 corresponds to the set $\Sigma \cup \{\varphi\}$ of MVVDs in Example 9. Note the correspondence between the counter-example relation $r$ for the implication of $\varphi$ by $\Sigma$ in Example 9 and the counter-example truth assignment $\theta$ for the logical implication of $\varphi'$ by $\Sigma'$ in Example 10. Indeed, $\theta$ assigns true to precisely those variables $A'$ that correspond to variables $A$ on which the two tuples in $r$ agree.

### 4.3. Axiomatisations

In this section, we will apply Theorem 4 to establish axiomatisations for the logical implication of $F_V$. 
Consider the rules for MVD implication from Table 2. Using our mapping $\Phi$ of MVDs over $\mathfrak{A}$ to formulae over $\mathbf{F}_V$ we obtain inference rules for the implication of $\mathbf{F}_V$. For instance, the pseudo-transitivity rule
\[
X \rightarrow Y, Y \rightarrow Z \\
\frac{}{X \rightarrow Z}
\]
for MVD implication becomes the pseudo-transitivity rule
\[
\bigwedge X \Rightarrow \bigwedge Y, \bigwedge Y \Rightarrow \bigwedge Z \\
\frac{}{\bigwedge X \Rightarrow \bigwedge Z}
\]
for $\mathbf{F}_V$-implication. For the sake of simplicity, we use $\bigwedge X$ as short for $\bigwedge \{A' \mid A \in X\}$.

For an arbitrary set $\mathcal{S}$ of inference rules from Table 2 without the $R$-complementation rule, let $\mathcal{S}'$ denote the corresponding set of inference rules for the implication of $\mathbf{F}_V$. Let $\mathcal{S}' \cup \{\varphi\}$ denote the finite set of formulae over $V$ that corresponds to the finite set $\Sigma \cup \{\varphi\}$ of MVDs over $\mathfrak{A}$. It is easy to see that $\Sigma \vdash \varphi$ if and only if $\mathcal{S}' \vdash \varphi'$. The following result follows then immediately from Theorem 4.

**Corollary 6.** Let $\mathcal{S}$ denote an arbitrary set of inference rules from Table 2 without the $R$-complementation rule. Then $\mathcal{S}'$ is a finite axiomatisation for the implication of $\mathbf{F}_V$ if and only if $\mathcal{S}'$ is a superset of at least one of the sets $\mathcal{S}'_1, \mathcal{S}'_2, \mathcal{S}'_3$.

### 4.4. Time-complexity of implication problem

We can define the notion of a dependency basis for a finite subset $X'$ of propositional variables in $V$ with respect to a finite set $\Sigma'$ of formulae in $\mathbf{F}_V$ in the same way we have defined the dependency basis $\text{Dep}B(X')$ for a finite attribute subset $X'$ of $\mathfrak{A}$ with respect to a finite set $\Sigma$ of MVDs over $\mathfrak{A}$. The following result follows then immediately from Corollary 3.

**Corollary 7.** The implication problem $\Sigma' \vdash \bigwedge X' \Rightarrow \bigwedge Y'$ can be decided in time $O((1 + \min(s, \log p)) \cdot n)$ where $s$ denotes the number of formulae in $\Sigma'$, $p$ the number of sets in $\text{Dep}B(X')$ that have non-empty intersection with $Y'$ and $n$ denotes the total number of occurrences of propositional variables in $\Sigma'$.

### 5. Characterisation by the implication in a fixed universes

In this section we will establish a characterisation of MVD implication in undetermined universes by the $R$-implication of MVDs in a certain fixed universe $R$ and the implication of functional dependencies. In subsequent sections we will apply this result to obtain proof-theoretical and algebraic characterisations as well. At the end of this section, we also establish an alternative logical characterisation of MVD implication in undetermined universes.

It has been shown [68] that the dependency basis $\text{Dep}B(X)$ of a finite attribute set $X$ with respect to a finite set $\Sigma$ of MVDs can be obtained by computing the scope $X^5_{\Sigma}$ of $X$ with respect to $\Sigma$ and computing the dependency basis $\text{Dep}B_R(X)$ of $X$ with respect to $\Sigma$ in any fixed universe $R$ such that $R^{\min} = \text{Attr}(\Sigma) \cup X \subseteq R$ holds.

**Corollary 8.** (See [68, Theorem 6.4].) Let $\Sigma \cup \{X \rightarrow Y\}$ be a finite set of multivalued dependencies over $\mathfrak{A}$. Let $R$ denote some relation schema such that $R^{\min} \subseteq R$ holds. Then $\Sigma \vdash \bigwedge X \rightarrow Y$ if and only if $Y \subseteq X^5_{\Sigma}$ and $X \rightarrow Y$ is $R$-implied by $\Sigma$.

Furthermore, it has been shown [68, Theorem 6.3] that Algorithm 1 computes the scope $X^5_{\Sigma}$ of a finite attribute set $X$ with respect to a finite set $\Sigma$ of MVDs over $\mathfrak{A}$.

**Algorithm 1** (Scope$(\Sigma, X)$).

**Input:** $(\Sigma, X)$ where $\Sigma$ is a finite set of MVDs, and $X$ is a finite set of attributes over $\mathfrak{A}$

**Output:** the scope $X^5_{\Sigma}$ of $X$ with respect to $\Sigma$

**Method:**

\[
\text{VAR} \ X^5_{\text{new}}, X^5_{\text{old}}, X^5_{\text{alg}} : \text{finite set of attributes}; \ MVDList : \text{List of MVDs};
\]

\[
\begin{align*}
(1) & \ X^5_{\text{new}} := X; \\
(2) & \ MVDList := \text{List of MVDs in } \Sigma; \\
(3) & \text{REPEAT} \\
(4) & \ X^5_{\text{old}} := X^5_{\text{new}}; \\
(5) & \text{Remove all attributes in } X^5_{\text{new}} \text{ from the left-hand side of all MVDs in } MVDList; \\
(6) & \text{FOR all MVDs } \emptyset \rightarrow Y \text{ in } MVDList \text{ LET } X^5_{\text{new}} := X^5_{\text{new}} \cup Y; \\
(7) & \text{UNTIL } X^5_{\text{new}} = X^5_{\text{old}}; \\
(8) & \text{RETURN}(X^5_{\text{new}});
\end{align*}
\]
At this stage we consider some of the existing concepts for the so-called functional dependencies [23]. The reason is that we can apply these concepts to characterise multivalued dependency implication in the undetermined context.

Functional dependencies between finite sets of attributes have played a central role in the study of relational databases [5,14,18,23,63], and seem to be central for the study of database design in other data models as well [4,55,66,88,98,99]. The notion of a functional dependency is well-understood and the semantic interaction between these dependencies has been syntactically captured by Armstrong’s well-known axioms [5]. A functional dependency (FD) [23] over the relation schema $R$ is an expression $X \rightarrow Y$ where $X, Y \subseteq R$. A relation $r$ over $R$ satisfies the FD $X \rightarrow Y$ if and only if every pair of tuples in $r$ that agrees on all of the attributes in $X$ also agrees on all of the attributes in $Y$. That is, $r$ satisfies $X \rightarrow Y$ if and only if for all $t_1, t_2 \in r$ with $t_1[X] = t_2[X]$ we have $t_1[Y] = t_2[Y]$. The closure $X^n_\Sigma$ of an attribute set $X \subseteq R$ under a set $\Sigma$ of FDs over $R$ is the set of all attributes $A \in R$ such that $X \rightarrow A$ is $R$-implied by $\Sigma$ [8]. Note that $X \rightarrow Y$ is $R$-implied by $\Sigma$ if and only if $Y \subseteq X^n_\Sigma$ [8].

**Remark 1.** The satisfaction of a functional dependency $X \rightarrow Y$ by a relation $r$ only depends on the values that occur in the projection $r[XY]$ of $r$ onto the attribute set $XY$. The only requirement on the relation $r$ is that $XY \subseteq \text{Attr}(r)$.

It follows that if we define functional dependencies and their implication over undetermined universes, then for an arbitrary finite set $\Sigma \cup \{\varphi\}$ of FDs over some relation schema $R$ with $\text{Attr}(\Sigma \cup \{\varphi\}) \subseteq R$ we have $\Sigma \models \varphi$ if and only if $\Sigma \models \varphi$.

Let $\Sigma_{\text{FD}} = \{X \rightarrow Y | X \rightarrow Y \in \Sigma\}$ denote the finite set of FDs that corresponds to the finite set of MVDs over $\mathfrak{A}$. After replacing each occurrence of the sequence of letters MVD by the sequence FD and after replacing the symbol $\rightarrow$ by the symbol $\models$, the method of Algorithm 1, on input $(\Sigma_{\text{FD}}, X)$, has as output the closure $X^n_{\Sigma_{\text{FD}}}$ of $X$ with respect to $\Sigma_{\text{FD}}$. In this case, Algorithm 1 is the exact algorithm for computing the closure $X^n_\Sigma$ of $X$ under $\Sigma_{\text{FD}}$, cf. [8,60].

**Corollary 9.** Let $\Sigma$ denote a finite set of multivalued dependencies, and let $X$ denote a finite set of attributes over $\mathfrak{A}$. Then $X^n_\Sigma = X^n_{\Sigma_{\text{FD}}}$. 

Before we derive the next characterisation we require the propositional fragment of $\mathfrak{A}$ that corresponds to the implication of functional dependencies [33]. The set $H_{\mathfrak{V}(\mathfrak{R})}$ of implicational Horn statements over $\mathfrak{V}(\mathfrak{R})$ is exactly the set $F_{\mathfrak{V}(\mathfrak{R})}$.

**Theorem 10.** Let $\Sigma \cup \{\varphi\}$ be a finite set of multivalued dependencies over $\mathfrak{A}$, and let $R^{\text{fix}} = \text{Attr}(\Sigma \cup \{\varphi\})$. Equivalent are:

1. $\Sigma \models_{\mathfrak{A}} \varphi$.
2. $\Sigma_{\text{FD}} \models_{\mathfrak{R}^{\text{fix}}} \varphi_{\text{FD}}$ and $\Sigma \models_{\mathfrak{R}^{\text{fix}}} \varphi$.
3. $\Sigma_{\text{FD}} \vdash_{\mathfrak{V}(\mathfrak{R}^{\text{fix}})} \varphi_{\text{FD}}'$ and $\Sigma' \vdash_{\mathfrak{V}(\mathfrak{R}^{\text{fix}})} \varphi'$. 

**Proof.** The equivalence between 2 and 3 follows directly from Theorem 1 and Fagin’s equivalence between the implication of functional dependencies and propositional Horn clauses [33].

It remains to show the equivalence between 1 and 2. Let $\varphi$ denote the multivalued dependency implication $X \rightarrow Y$. Since $R^{\text{min}} \subseteq R^{\text{fix}}$ we conclude by Corollary 8 that $\Sigma \models_{\mathfrak{A}} \varphi$ if and only if $Y \subseteq X^n_{\Sigma_{\text{FD}}}$ and $\Sigma \models_{\mathfrak{R}^{\text{fix}}} \varphi$. By Corollary 9 we conclude that $\Sigma \models_{\mathfrak{A}} \varphi$ if and only if $Y \subseteq X^n_{\Sigma_{\text{FD}}}$ and $\Sigma \models_{\mathfrak{R}^{\text{fix}}} \varphi$. A well-known result by Beeri and Bernstein [8] says that $Y \subseteq X^n_{\Sigma_{\text{FD}}}$ if and only if $\Sigma_{\text{FD}} \models_{\mathfrak{R}^{\text{fix}}} \varphi_{\text{FD}}$. Consequently, it follows that $\Sigma \models_{\mathfrak{A}} \varphi$ if and only if $\Sigma_{\text{FD}} \models_{\mathfrak{R}^{\text{fix}}} \varphi_{\text{FD}}$ and $\Sigma \models_{\mathfrak{R}^{\text{fix}}} \varphi$. 

**Example 12.** Let $\Sigma$ denote the set of the FD $(\text{Employee} \rightarrow \text{Salary}, \text{Year})$ and let $\varphi$ denote the MVD $(\text{Employee} \rightarrow \text{Salary})$. Suppose we want to use Theorem 10 to decide whether $\Sigma$ implies $\varphi$.

$\Sigma_{\text{FD}}$ denotes the set $(\text{Employee} \rightarrow \text{Salary}, \text{Year})$ and $\varphi_{\text{FD}}$ denotes the FD $(\text{Employee} \rightarrow \text{Salary})$. Consequently, $\varphi_{\text{FD}}$ is $R^{\text{fix}}$-implied by $\Sigma_{\text{FD}}$ where $R^{\text{fix}}$ denotes the set $(\text{Employee}, \text{Salary})$. However, $\varphi$ is not $R^{\text{fix}}$-implied by $\Sigma$. Consequently, $\Sigma$ does not imply $\varphi$ by the equivalence between 1 and 2 in Theorem 10.

The same result can be derived by using the equivalence between 2 and 3 in Theorem 10. We have $V(R^{\text{fix}}) = \{V_E, V_S, V_Y\}$, $\Sigma_{\text{FD}}$ denotes the set $(V_E \Rightarrow V_S \wedge V_Y)$ and $\varphi_{\text{FD}}$ denotes $V_E \Rightarrow V_S$. Obviously, $\Sigma_{\text{FD}} \models_{\mathfrak{V}(R^{\text{fix}})} \varphi_{\text{FD}}$. Finally, $\Sigma'$ denotes the set $(V_E \Rightarrow V_S \wedge V_Y)$ and $\varphi'$ denotes $V_E \Rightarrow V_S$. However, the truth assignment $\theta$ such that $\theta(V_S) = \theta(V_Y) = \text{false}$ is a model of (the tautology) $V_E \Rightarrow V_S \wedge V_Y$ but not a model of $\varphi'$. That is, $\Sigma'$ does not $\mathfrak{V}(R^{\text{fix}})$-imply $\varphi$. According to Theorem 10 it follows that $\Sigma$ does not imply $\varphi$. 


6. Proof-theoretical characterisation

In this section we will show how the chase can be applied to deciding the implication problem of multivalued dependencies in undetermined universes. Therefore, we will first summarise how the chase works in the context of a fixed relation schema. We will not discuss any further applications of the chase to query optimisation or other fields.

6.1. The chase in fixed universes

A tableau [70] is a two-dimensional matrix in which columns correspond to attributes. The rows of a tableau consist of variables of the following types:

1. **distinguished variables**, usually denoted by subscripted a’s, and
2. **nondistinguished variables**, usually denoted by subscripted b’s.

A variable cannot appear in more than one column, and in each column there is exactly one distinguished variable. A multivalued dependency \( X \rightarrow Y \) over the relation schema \( R \), denoted by \( \phi \), has a corresponding tableau \( T_\phi \) as follows. For \( XY \), the tableau \( T_\phi \) has a row \( t_1 \) with distinguished variables in all the \( XY \) columns and distinct nondistinguished variables in the rest of the columns. For \( X(R - XY) \), the tableau \( T_\phi \) has a row \( t_2 \) with distinguished variables in all the \( X(R - XY) \) columns and distinct nondistinguished variables in the rest of the columns. We can also view a tableau as a relation over the domain of distinguished and nondistinguished variables. Note that rows \( t_1, t_2 \) of \( T_\phi \) are **joinable** on \( X \), and the resulting row \( t \) consists only of distinguished variables. That is, \( t[XY] = t_1[XY] \) and \( t[X(R - XY)] = t_2[X(R - XY)] \), and \( t[A] = a_A \) for all \( A \in R \).

**Example 13.** Let \( \Sigma = \{ \text{Employee} \rightarrow \text{Child} \} \) and let \( \phi = \text{Employee} \rightarrow \text{Salary} \). Suppose we want to use Theorem 10 to decide whether \( \Sigma \) implies \( \phi \). It follows that \( \phi \) is \( R^{\phi} \)-implied by \( \Sigma \) where \( R^{\phi} = \{ \text{Employee}, \text{Child}, \text{Salary} \} \). Moreover, \( \Sigma_{FD} = \{ \text{Employee} \rightarrow \text{Child} \} \) and \( \phi_{FD} = \text{Employee} \rightarrow \text{Salary} \). However, \( \phi_{FD} \) is not \( R^{\phi} \)-implied by \( \Sigma_{FD} \). Consequently, \( \Sigma \) does not imply \( \phi \) by the equivalence between 1 and 2 in Theorem 10. The same result can be derived by using the equivalence between 1 and 3 in Theorem 10. Indeed, \( \forall (R^{\phi}) = \{ V_E, V_C, V_S \} \). \( \Sigma' \) denotes the set \( \{ V_E \Rightarrow V_C \} \) and \( \phi' \) denotes \( V_E \Rightarrow V_S \). It follows that \( \phi' \) is \( V(R^{\phi}) \)-implied by \( \Sigma \). Finally, \( \Sigma_{FD} \) denotes the set \( \{ V_E \Rightarrow V_C \} \) and \( \phi_{FD}' \) denotes \( V_E \Rightarrow V_S \). However, the truth assignment \( \theta \) such that \( \theta(V_E) = \theta(V_C) = \text{true} \) and \( \theta(V_S) = \text{false} \) is a model of \( \Sigma_{FD} \) but not a model of \( \phi_{FD}' \). That is, \( \Sigma_{FD} \) does not \( V(R^{\phi}) \)-imply \( \phi_{FD}' \). According to Theorem 10, \( \Sigma \) does not imply \( \phi \).

The first five equivalences of Table 1 have been summarised more succinctly in Table 3.

**Table 3**

A summary of the first five equivalences.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ( \Sigma \vdash_{A} \phi )</td>
<td>2. ( \Sigma \vdash \phi )</td>
<td>3. ( \Sigma' \vdash \phi' )</td>
</tr>
<tr>
<td>4. ( \Sigma_{FD} \vdash_{A} \phi_{FD} ) and ( \Sigma \vdash_{A} \phi )</td>
<td>5. ( \Sigma_{FD} \vdash \phi_{FD} ) and ( \Sigma \vdash \phi )</td>
<td></td>
</tr>
</tbody>
</table>

\[ \Sigma \vdash_{A} \phi \]

\[ \Sigma \vdash \phi \]

\[ \Sigma' \vdash \phi' \]

\[ \Sigma_{FD} \vdash_{A} \phi_{FD} \] and \( \Sigma \vdash_{A} \phi \)

\[ \Sigma_{FD} \vdash \phi_{FD} \] and \( \Sigma \vdash \phi \)

Example 14. Consider the relation schema \( \{ \text{Employee, Child, Salary} \} \). The corresponding tableau for the MVD \( \text{Employee} \rightarrow \text{Child} \) looks as follows:

<table>
<thead>
<tr>
<th>Employee</th>
<th>Child</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_E )</td>
<td>( a_C )</td>
<td>( b_1 )</td>
</tr>
<tr>
<td>( a_E )</td>
<td>( b_2 )</td>
<td>( a_s )</td>
</tr>
</tbody>
</table>

where \( a_E, a_C, a_s \) denote the distinguished variables and \( b_1, b_2 \) denote the nondistinguished variables.

Let \( \Sigma \) be a set of FDs and MVDs over the relation schema \( R \). Each dependency in \( \Sigma \) has an associated rule that can be applied to any tableau \( T \) as follows.

1. **FD-rule.** An FD \( X \rightarrow Y \) in \( \Sigma \) has an associated rule for equating variables of \( T \) as follows. Suppose that rows \( t_1 \) and \( t_2 \) of \( T \) agree in all \( X \)-columns but disagree in an \( A \)-column, where \( A \) is an attribute in \( Y \). If one of \( t_1 \) and \( t_2 \) has a distinguished variable in its \( A \)-column, then rename the two rows so that \( t_1 \) is that row. The FD-rule for \( X \rightarrow Y \)
replaces all occurrences of the variable appearing in the A-column of \( t_2 \) with the variable appearing in the A-column of \( t_1 \).

2. **MVD-rule.** An MVD \( X \rightarrow Y \) in \( \Sigma \) has an associated rule for adding rows to \( T \) as follows. If rows \( t_1, t_2 \) of \( T \) are joinable on \( X \) with a result \( t \) and \( t \) is not already in \( T \), then \( t \) is added to \( T \).

Each one of the above rules transforms a tableau \( T \) to another tableau \( T' \). The rules can be applied repeatedly to a tableau \( T \) only a finite number of times, and the result is unique (up to renaming of nondistinguished variables) [70]. The chase of \( T \) by \( \Sigma \), denoted by \( \text{chase}_\Sigma(T) \), is the tableau obtained by applying the rules associated with \( \Sigma \) to \( T \) until no rule can be applied anymore. Let \( \varphi \) denote an MVD over the relation schema \( R \) with a corresponding tableau \( T^R_\varphi \). The MVD \( \varphi \) is \( R \)-implied by \( \Sigma \) if and only if \( \text{chase}_\Sigma(T^R_\varphi) \) contains a row consisting of distinguished variables only [70].

**Example 15.** Consider the relation schema \( R = \{ \text{Employee}, \text{Child}, \text{Salary} \} \), and let \( \Sigma \) consist of the single MVD \( \text{Employee} \rightarrow \text{Child} \). Suppose we want to decide whether the MVD \( \text{Employee} \rightarrow \text{Salary} \), denoted by \( \varphi \), is \( R \)-implied by \( \Sigma \). As in Example 14 the tableau \( T^R_\varphi \) is

\[
\begin{array}{ccc}
\text{Employee} & \text{Child} & \text{Salary} \\
\text{a}_E & \text{a}_C & b_1 \\
\text{a}_E & b_2 & a_5 \\
\end{array}
\]

The MVD-rule can be applied to \( T^R_\varphi \) to generate the following tableau:

\[
\begin{array}{ccc}
\text{Employee} & \text{Child} & \text{Salary} \\
\text{a}_E & \text{a}_C & b_1 \\
\text{a}_E & a_5 & a_5 \\
\text{a}_E & b_2 & b_1 \\
\text{a}_E & a_5 & a_5 \\
\end{array}
\]

which is \( \text{chase}_\Sigma(T^R_\varphi) \). It follows that \( \varphi \) is \( R \)-implied by \( \Sigma \) as \( \text{chase}_\Sigma(T^R_\varphi) \) contains the row \((a_E, a_C, a_5)\).

The FD \( X \rightarrow Y \) over the relation schema \( R \) has the following corresponding tableau \( T^R_X \) which has two rows: \( t_1 \) consists of distinguished variables only, and \( t_2 \) has distinguished variables in the \( X \)-columns and distinct nondistinguished variables elsewhere. The FD \( X \rightarrow Y \) is \( R \)-implied by \( \Sigma \) if and only if \( \text{chase}_\Sigma(T^R_X) \) has only distinguished variables in all of the \( Y \)-columns. The closure \( X^*_{\Sigma} \) is the set of all attributes \( A \in R \) such that the \( A \)-column of \( \text{chase}_\Sigma(T^R_X) \) has only distinguished variables [70, Corollary 1].

6.2. The chase in undetermined universes

We will now define a chase for multivalued dependencies in undetermined universes. Consider again Example 15. The set \( \Sigma = \{ \text{Employee} \rightarrow \text{Child} \} \) does not imply the MVD \( \varphi = \text{Employee} \rightarrow \text{Salary} \). Without Corollary 8 it does not seem obvious at all how the chase can be applied to make decisions about the implication of MVDs in undetermined universes. In this example, the reason that \( \varphi \) is not implied by \( \Sigma \) is that \( \text{Salary} \) is not in the scope \( \text{Employee}^*_{\Sigma} \) of \( \text{Employee} \) with respect to \( \Sigma \). This indicates that we require the computation of the scope of a finite attribute set with respect to a given finite set of MVDs in order to decide MVD implication in undetermined universes. For this purpose, it follows immediately from Corollary 9 that the chase of functional dependencies in fixed universes can be utilised to compute the scope.

**Corollary 11.** Let \( \Sigma \) be a finite set of multivalued dependencies, and let \( X \) be a finite set of attributes over \( A \). Then \( X^*_{\Sigma} \) is the set of all attributes \( A \in R^{\min}_{\Sigma} = \text{Attr}(\Sigma) \cup X \) such that the \( A \)-column of \( \text{chase}_{\Sigma}(T^R_X) \) has only distinguished variables.

Based on the chase in fixed universes [70, Theorem 4.5], and based on Corollary 11 we can devise a chase procedure that correctly decides the implication problem of MVDs in undetermined universes.

**Algorithm 2** (Chase\((\Sigma, X \rightarrow Y)\)).

**Input:** \((\Sigma, X \rightarrow Y)\) where \( \Sigma \cup \{ X \rightarrow Y \} \) is a finite set of MVDs over \( A \)

**Output:**

\[
\begin{cases}
\text{Yes,} & \text{if } \Sigma \models_{\Omega} X \rightarrow Y \\
\text{No,} & \text{otherwise}
\end{cases}
\]
Method:

VAR \( R_{\text{min}} \): finite set of attributes;

1. \( R_{\text{min}} := \text{Attr}(\Sigma) \cup X; \)
2. Compute \( X^\Sigma \) as the set of attributes \( A \) such that the \( A \)-column of \( \text{chase}_{\Sigma}(T_{\text{min}}^A) \) has only distinguished variables, and where \( \Sigma_{\text{FD}} = \{X \rightarrow Y \mid X \rightarrow Y \in \Sigma\}; \)
3. IF \( Y \not\subseteq X^\Sigma \), THEN RETURN (No); ELSE
   IF \( \text{chase}_{\Sigma}(T_{\text{min}}^A \rightarrow X) \) contains a row with distinguished variables only,
   THEN RETURN (Yes) ELSE RETURN (No);
ENDIF;
ENDIF;

Theorem 12. Let \( \Sigma \cup \{\varphi\} \) be a finite set of multivalued dependencies over \( \mathcal{A} \). Then \( \Sigma \models \varphi \) if and only if \( \text{Chase}(\Sigma, \varphi) = \text{Yes} \).

Proof. Let \( \varphi \) denote the MVD \( X \rightarrow Y \). It then follows from Corollary 8 that \( \Sigma \models \varphi \) if and only if \( Y \subseteq X^\Sigma \) and \( \Sigma \models R_{\text{min}} \varphi \).

Example 16. Let \( \Sigma = \{\text{Employee} \rightarrow \text{Child}\} \), and \( \varphi \) be the MVD \( \text{Employee} \rightarrow \text{Salary} \). The set \( R_{\text{min}} = \{\text{Employee}, \text{Child}\} \) and \( \text{Employee}^\Sigma = \{\text{Employee}, \text{Child}\} \) results from \( \text{chase}_{\Sigma}(T_{\text{Employee}}^A) \):

\[
\begin{array}{ll}
\text{Employee} & \text{Child} \\
\hline
a_E & a_C \\
\end{array}
\]

by chasing the tableau \( T_{\text{Employee}}^A \):

\[
\begin{array}{l|l}
\text{Employee} & \text{Child} \\
\hline
a_E & a_C \\
\end{array}
\]

with \( \Sigma_{\text{FD}} = \{\text{Employee} \rightarrow \text{Child}\} \). Consequently, \( \{\text{Salary}\} \not\subseteq \text{Employee}^\Sigma \) and therefore Algorithm 2 returns ‘No’.

The last example illustrates that Algorithm 2 may already return ‘No’ in case where \( Y \not\subseteq R_{\text{min}} \). Let us consider a final example to illustrate Algorithm 2.

Example 17. Let \( \Sigma = \{A \rightarrow BCD, E \rightarrow BCF\} \), and let \( \varphi \) be the MVD \( A \rightarrow BC \) and \( \psi \) be the MVD \( A \rightarrow DEF \). The set \( R_{\text{min}} = ABCDEF \) and \( A^\Sigma = ABCD \) results from \( \text{chase}_{\Sigma}(T_{\text{Employee}}^A) \):

\[
\begin{array}{ccccccc}
A & B & C & D & E & F \\
\hline
a_A & a_B & a_C & a_D & a_E & a_F \\
a_A & a_B & a_C & a_D & b_E & b_F \\
\end{array}
\]

by chasing the tableau \( T_{\text{Employee}}^A \):

\[
\begin{array}{ccccccc}
A & B & C & D & E & F \\
\hline
a_A & a_B & a_C & a_D & a_E & a_F \\
a_A & b_B & b_C & b_D & b_E & b_F \\
\end{array}
\]

with \( \Sigma_{\text{FD}} = \{A \rightarrow BCD, E \rightarrow BCF\} \). Since \( DEF \not\subseteq A^\Sigma \), it follows that Algorithm 2 returns ‘No’, i.e., \( \Sigma \) does not imply \( \psi \). Note that \( \Sigma \models R_{\text{min}} \text{-implies } \psi \). However, \( BC \subseteq A^\Sigma \) and therefore Algorithm 2 continues on input \((\Sigma, \varphi)\). The tableau \( T_{\psi}^A \) is:

\[
\begin{array}{ccccccc}
A & B & C & D & E & F \\
\hline
a_A & a_B & a_C & b_D & b_E & b_F \\
a_A & b_B & b_C & b_D & a_E & a_F \\
\end{array}
\]
After applying the MVD-rule to $A \rightarrow BCD$ we obtain the tableau:

\[
\begin{array}{cccccc}
A & B & C & D & E & F \\
\hline
a_A & a_B & a_C & b_D & b_E & b_F \\
a_A & b_B & b_C & a_D & a_E & a_F \\
a_A & a_B & a_C & b_D & a_E & a_F \\
a_A & b_B & b_C & b_D & b_E & b_F \\
a_A & b_B & b_C & b_D & a_E & a_F \\
a_A & a_B & a_C & a_D & a_E & a_F \\
\end{array}
\]

Subsequently, we apply the MVD-rule to $E \rightarrow BCF$ to obtain the tableau:

\[
\begin{array}{cccccc}
A & B & C & D & E & F \\
\hline
a_A & a_B & a_C & b_D & b_E & b_F \\
a_A & b_B & b_C & a_D & a_E & a_F \\
a_A & a_B & a_C & b_D & a_E & a_F \\
a_A & b_B & b_C & b_D & b_E & b_F \\
a_A & b_B & b_C & b_D & a_E & a_F \\
a_A & a_B & a_C & a_D & a_E & a_F \\
a_A & b_B & b_C & a_D & b_E & b_F \\
\end{array}
\]

Finally, we apply the MVD-rule to $A \rightarrow BCD$ to obtain $\text{chase}_\Sigma(T^{R_{\min}})$:

\[
\begin{array}{cccccc}
A & B & C & D & E & F \\
\hline
a_A & a_B & a_C & b_D & b_E & b_F \\
a_A & b_B & b_C & a_D & a_E & a_F \\
a_A & a_B & a_C & b_D & a_E & a_F \\
a_A & b_B & b_C & b_D & b_E & b_F \\
a_A & b_B & b_C & b_D & a_E & a_F \\
a_A & a_B & a_C & a_D & a_E & a_F \\
a_A & b_B & b_C & a_D & b_E & b_F \\
\end{array}
\]

Since this tableau contains the row $(a_A, a_B, a_C, a_D, a_E, a_F)$, Algorithm 2 returns ‘Yes’, i.e., $\Sigma \models_{\mathfrak{A}} \varphi$.

The time-complexity for deciding the implication of multivalued dependencies in undetermined universes using Algorithm 2 can easily be derived from [71, Theorem 13].

**Corollary 13.** Let $\Sigma \cup \{X \rightarrow Y\}$ denote a finite set of multivalued dependencies over $\mathfrak{A}$. Then Algorithm 2 can be implemented to decide $\Sigma \models_{\mathfrak{A}} X \rightarrow Y$ in time $O(|\text{Attr}(\Sigma) \cup X| \times \|\Sigma\|)$.

In fixed universes, the chase has been developed to decide the implication of arbitrary sets of functional and join dependencies [70]. For a set $\Sigma$ of a functional and a join dependency, and a join dependency $\varphi$ over a relation schema $R$, it is $NP$-complete to decide whether $\varphi$ is $R$-implied by $\Sigma$. However, if $\varphi$ denotes a functional or multivalued dependency and $\Sigma$ denotes an arbitrary set of functional and join dependencies over $R$, then the chase works in time $O(|R| \times \|\Sigma\|)$. It is a possible direction for future work to study join dependencies in undetermined universes, and to develop an appropriate extension of Algorithm 2.

### 7. Algebraic characterisation

In this section we establish a simple set-theoretic characterisation of MVD implication over undetermined universes.

#### 7.1. The characterisation in fixed universes

A set $S$ of attributes over $R$ is said to be closed with respect to the functional dependency $X \rightarrow Y$ over $R$ if $X \subseteq S$ implies that $Y \subseteq S$ [8]. We say that $S$ is closed with respect to the multivalued dependency $X \rightarrow Y$ over $R$ if $X \subseteq S$ implies that $Y \subseteq S$ or $R - Y \subseteq S$ [45]. Let $\Sigma$ denote a set of FDs and MVDs over $R$. We say that $S$ is closed with respect to $\Sigma$ if $S$ is closed with respect to every member $\sigma$ of $\Sigma$.

**Theorem 14.** (See [45, Theorem 3.1].) Let $\Sigma \cup \{\varphi\}$ be a set of functional and multivalued dependencies over the relation schema $R$. Then $\Sigma \models_{R} \varphi$ if and only if every set $S$ of attributes over $R$ that is closed with respect to $\Sigma$ is also closed with respect to $\varphi$. 

\[
\begin{array}{cccccc}
A & B & C & D & E & F \\
\hline
a_A & a_B & a_C & b_D & b_E & b_F \\
a_A & b_B & b_C & a_D & a_E & a_F \\
a_A & a_B & a_C & b_D & a_E & a_F \\
a_A & b_B & b_C & b_D & b_E & b_F \\
a_A & b_B & b_C & b_D & a_E & a_F \\
a_A & a_B & a_C & a_D & a_E & a_F \\
a_A & b_B & b_C & a_D & b_E & b_F \\
\end{array}
\]
Example 18. Let \( R \) denote the relation schema \( \{ \text{Employee}, \text{Child}, \text{Salary} \} \), let \( \Sigma \) consist of the single MVD \( \text{Employee} \rightarrow \text{Child} \) and let \( \phi \) denote the single MVD \( \text{Employee} \rightarrow \text{Salary} \). Then every set \( S \subseteq R \) is closed with respect to \( \Sigma \) if and only if \( S \) is closed with respect to \( \phi \). In particular, \( \Sigma \models_R \phi \).

7.2. The characterisation in undetermined universes

We can apply Theorem 10 and Theorem 14 to derive a simple set-theoretical characterisation of MVD implication in undetermined universes.

Theorem 15. Let \( \Sigma \cup \{ \phi \} \) be a finite set of multivalued dependencies over \( \mathcal{A} \), and let \( R^{\text{fix}} = \text{Attr}(\Sigma \cup \{ \phi \}) \). Then \( \Sigma \models_{\mathcal{A}} \phi \) if and only if both of the following conditions hold:

1. every set \( S \subseteq R^{\text{fix}} \) that is closed with respect to \( \Sigma_{\text{FD}} \) is also closed with respect to \( \psi_{\text{FD}} \).
2. every set \( S \subseteq R^{\text{fix}} \) that is closed with respect to \( \Sigma \) is also closed with respect to \( \phi \).

Proof. It follows from Theorem 10 that \( \Sigma \models_{\mathcal{A}} \phi \) if and only if \( \Sigma_{\text{FD}} \models_{R^{\text{fix}}} \psi_{\text{FD}} \) and \( \Sigma \models_{R^{\text{fix}}} \phi \). Theorem 15 follows then immediately from Theorem 14. \( \square \)

Example 19. Let \( \Sigma \) consist of the single MVD \( \text{Employee} \rightarrow \text{Child} \) and let \( \phi \) denote the single MVD \( \text{Employee} \rightarrow \text{Salary} \). Consequently, \( R^{\text{fix}} \) denotes the set \( \{ \text{Employee}, \text{Child}, \text{Salary} \} \). Moreover, \( \Sigma_{\text{FD}} \) consists of the functional dependency \( \text{Employee} \rightarrow \text{Child} \) and \( \psi_{\text{FD}} \) denotes the functional dependency \( \text{Employee} \rightarrow \text{Salary} \). Let \( S \) denote the set \( \{ \text{Employee}, \text{Child} \} \). Then \( S \) is closed with respect to \( \Sigma_{\text{FD}} \), but \( S \) is not closed with respect to \( \Sigma \). According to Theorem 15, \( \Sigma \) does not imply \( \phi \).

Example 20. Let \( \Sigma = \{ A \rightarrow BCD, E \rightarrow BCF \} \), and let \( \phi \) be the MVD \( A \rightarrow BC \) and \( \psi \) be the MVD \( A \rightarrow DEF \). Suppose we want to use Theorem 15 to decide whether \( \Sigma \models_{\mathcal{A}} \phi \) and whether \( \Sigma \models_{\mathcal{A}} \psi \). In both cases, \( R^{\text{fix}} \) denotes the set \( ABCDEF \). Furthermore, \( \Sigma_{\text{FD}} \) consists of the functional dependencies \( A \rightarrow BCD \) and \( E \rightarrow BCF \). \( \psi_{\text{FD}} \) denotes the FD \( A \rightarrow BC \) and \( \psi_{\text{FD}} \) denotes the FD \( E \rightarrow BCF \). The set \( S = ABCD \) is closed with respect to \( \Sigma_{\text{FD}} \), but it is not closed with respect to \( \psi_{\text{FD}} \). Consequently, \( \Sigma \) does not imply \( \psi \). It is easy to observe that every set \( S \subseteq R^{\text{fix}} \) that is closed with respect to \( \Sigma_{\text{FD}} \) is also closed with respect to \( \psi_{\text{FD}} \). Moreover, every set \( S \subseteq R^{\text{fix}} \) that is closed with respect to \( \Sigma \) is also closed with respect to \( \phi \). In particular, the set \( S = ADEF \) is closed with respect to both \( \Sigma \) and \( \phi \). It follows that \( \phi \) is implied by \( \Sigma \).

8. Conclusion and future work

The interaction of multivalued dependencies in relational databases has been well-studied in the context of a fixed underlying relation schema. Since the assumption of having such a fixed universe is commonly infeasible in practice, Biskup introduced an alternative notion of MVD implication in which the underlying universe is left undetermined [17]. We have characterised this alternative notion of MVD implication from different perspectives. In particular, we have shown that the assumption of a fixed universe is not necessary for establishing correspondences to fragments of propositional logic, to the Chase, and to closed attribute sets. The results of this paper can directly be applied to the theory of conditional independencies in Bayesian networks [100]. Finally, the following is a list of open problems that warrant future research:

- Find a synthesis approach towards database normalisation with respect to multivalued dependencies.
- Develop a normalisation and de-normalisation theory that takes into account the most common queries and updates.
- Include data dependencies into the framework of finite model theory [a start of this has been made [2]].
- Extend the knowledge on the relationship between data dependencies, query optimisation and physical database tuning.
- Investigate the implication of join dependencies in undetermined universes.
- Establish explicit correspondences between data dependencies in undetermined universes and notions of conditional independencies in Bayesian networks.
- Develop and investigate notions of multivalued and join dependencies in other data models, for instance in XML.
- Extend the knowledge on structural and computational properties of Armstrong databases for functional dependencies to multivalued dependencies [10,47,48,72].

References
