ELSEVIER

Artificial Intelligence 108 (1999) 1–30

Artificial Intelligence

Functional dependencies in Horn theories

Toshihide Ibaraki [a,1], Alexander Kogan [b,c,*], Kazuhisa Makino [d,2]

[a] *Department of Applied Mathematics and Physics, Graduate School of Informatics, Kyoto University, Kyoto, Japan 606*

[b] *Department of Accounting and Information Systems, Faculty of Management, Rutgers University, Newark, NJ 07102, USA*

[c] *RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, USA*

[d] *Department of Systems and Human Science, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, Japan*

Received 12 March 1997; received in revised form 29 July 1998

Abstract

This paper studies functional dependencies in Horn theories, both when the theory is represented by its clausal form and when it is defined as the Horn envelope of a set of models. We provide polynomial algorithms for the recognition of whether a given functional dependency holds in a given Horn theory, as well as polynomial algorithms for the generation of some representative sets of functional dependencies. We show that some problems of inferring functional dependencies (e.g., constructing an irredundant FD-cover) are computationally difficult. We also study the structure of functional dependencies that hold in a Horn theory, showing that every such functional dependency is in fact a single positive term Boolean function, and prove that for any Horn theory the set of its minimal functional dependencies is quasi-acyclic. Finally, we consider the problem of condensing a Horn theory, prove that any Horn theory has a unique condensation, and develop an efficient polynomial algorithm for condensing Horn theories.

Keywords: Knowledge representation; Horn theory; Functional dependency; Condensation; Computational complexity; Conjunctive normal form; Acyclic directed graph

1. Introduction

Relational databases have been invented, studied and deployed as essential tools of information storage and retrieval (see [10,33,34,42,43]). Functional dependencies have

* Corresponding author. Email: kogan@rutcor.rutgers.edu.

1 Email: ibaraki@kuamp.kyoto-u.ac.jp.

2 Email: makino@sys.es.osaka-u.ac.jp.

0004-3702/99/$ – see front matter
PII: S0004-3702(98)00114-3

been recognized to be one of the most important concepts in the relational database theory (see [1,14]). Functional dependencies state that the values of certain attributes in a relation are determined by the values of some other attributes. They are commonly used in the logical database design to express integrity constraints, and thus to express domain knowledge. The problems of inferring functional dependencies from relations have been studied in [30,35]. Thorough theoretical studies of functional dependencies in relational databases (see [12,14,15,28,39]) have established a close connection with Horn clauses.

Horn clauses were introduced in formal logic (see [21,36]), and gained prominence in logic programming (see [13]) and artificial intelligence (see [9,11,24]). In artificial intelligence, the implementation of a knowledge base as a Horn theory is often preferred, since linear time complexity of solving Horn satisfiability problems (see [13,37]) provides the benefits of computationally tractable reasoning, while Horn clauses have the expressive power sufficient for many applications.

A Horn theory is characterized by the condition that the intersection of any two models is again a model. A theory can be viewed as the set of its models, and reasoning with models has been developed in recent AI studies (see [24,27,29]). In model-based representation, a theory is represented by a subset of its models, which are commonly called characteristic models [24,27,29]. From the database theory point of view, the set of models is in fact a relation. This relation may have functional dependencies, which reveal important structural properties of the theory by describing the intrinsic determinants of values of certain attributes. Individual functional dependencies can provide valuable insights into hidden laws of the problem domain, and can be used by domain experts for evaluating and verifying the theory. The inference of functional dependencies in a Horn theory can thus provide a means of its qualitative analysis, and can also be considered to be a form of knowledge discovery.

The knowledge of functional dependencies in a theory may allow to simplify the theory by eliminating those variables whose values are determined by the values of other variables. This “condensation” procedure will result in a theory which does not have any functional dependencies, can have much fewer variables than the original theory, and can be structurally simpler than the original theory. The computational expense of condensing a theory can be offset by the speedup of queries to the knowledge base, and therefore condensation can provide significant computational benefits. Moreover, the condensed theory can be viewed as the “core” of the original theory, and thus condensation can reveal important structural information about the problem domain.

Knowledge condensation represents a special type of knowledge preprocessing, which attempts to spend some computational resources at the preliminary stage to transform a knowledge base in such a way that the transformed one can be used to reason and answer queries with less computational effort. The computational expense of knowledge preprocessing is quickly amortized over the large number of queries during routine operations. A well developed type of knowledge preprocessing is known under the name of knowledge compilation (see [25,40]), which constructs Horn upper and lower bounds of a general Boolean theory and attempts to use them for answering queries. For some queries, such attempts can be successful, providing fast answers which would be impossible to obtain using the original Boolean theory. If the Horn bounds do not provide an answer, then the original theory has to be used to answer queries. It is interesting to note that knowledge compilation attempts to reduce the size of the Horn upper bound by introducing additional variables in the theory (see [25]), while knowledge condensation aims to simplify the problem by eliminating redundant variables from the theory.

Another well developed type of knowledge preprocessing is knowledge compression (see [6,17–20]), which shortens the length of a Horn CNF without changing the Horn theory it represents. While knowledge compilation aims to reduce an intractable problem to a tractable one, knowledge compression, similarly to knowledge condensation, is developed for Horn theories, and therefore simplifies a problem which is already tractable. Such simplifications are nevertheless very important, since Horn theories used in practical applications can have very long representations. This is typical in many applications where propositional Horn theories can be generated automatically, e.g., when first-order Horn theories are instantiated over finite but large domains. In these situations, the possible significant size reductions provided by knowledge compression and condensation become essential. Both knowledge compression and condensation can be used together with knowledge compilation to simplify the Horn bounds it produces.

This paper is devoted to the studies of functional dependencies in Horn theories. It focuses on characterizing the combinatorial structure of such functional dependencies, and on developing efficient polynomial algorithms for recognizing, inferring and using them. We consider the problems arising when a theory is represented by its Horn clausal form, as well as when it is defined as the Horn envelope (see [26]) of a set of models (i.e., it is represented by characteristic models).

The results of this paper reveal new properties of Horn theories, and can be used to make knowledge representation and reasoning computationally more efficient. We provide polynomial algorithms to recognize whether a given functional dependency holds in a given Horn theory, as well as polynomial algorithms to generate some representative sets of functional dependencies. We show that some problems of inferring functional dependencies (e.g., constructing an irredundant FD-cover) are computationally difficult. We also study the structure of functional dependencies that hold in a Horn theory, show that every such functional dependency is in fact a single positive term Boolean function, and prove that for any Horn theory the set of its minimal functional dependencies is quasi-acyclic.

Finally, we apply the obtained structural and algorithmic results about functional dependencies in Horn theories to the problem of condensing a Horn theory. We prove that, in contrast with the case of general Boolean theories, any Horn theory has a unique condensation. We show that a Horn theory can be totally condensed using a very limited number of functional dependencies, and develop an efficient polynomial algorithm for condensing Horn theories. The condensation of a Horn theory represented as the Horn envelope of a set of models always reduces the size of the representation, and therefore is computationally advantageous. The condensation of a Horn CNF may result (in the worst case) in a moderate polynomial increase in the length of the CNF. On the other hand, examples show that the potential reduction in the length of the CNF resulting from condensation can be exponential. It makes sense at the preprocessing stage to attempt condensing a Horn CNF, since in the worst case (where the size increases) only a polynomial amount of computational effort is wasted, while in the case of success (where the size decreases) the computational benefits will be utilized continuously over the long run. Additionally, knowledge compression can be applied to the condensed theory, which may be easier to compress than the original one since it contains fewer variables and can have simpler structure.

2. Notation and basic concepts

Propositional variables taking the values in $\{0, 1\}$ (meaning *false* and *true*, respectively, and assuming $0 < 1$) will be denoted by lower case Latin letters (usually from the end of the alphabet), with $\overline{x}$ denoting the negation of x. Propositional variables and their negations will be called *literals*, with the variables themselves called *positive literals* and their negations called *negative literals*. Upper case Latin letters (usually from the end of the alphabet) will be used to denote sets of propositional variables, with the letter V reserved to denote the set of all variables (in most cases assumed to be $\{x_1, x_2, \ldots, x_n\}$). Boolean vectors (points, or models) in $\{0, 1\}^n$ will be denoted by lower case Greek letters, with $\alpha[X]$ denoting the restriction of a point $\alpha \in \{0, 1\}^n$ to the set of variables in $X \subseteq V$. We will denote as $\alpha \leqslant \beta$ the condition that $\alpha_i \leqslant \beta_i$ for all $i = 1, 2, \ldots, n$, and as $\alpha < \beta$ the condition that $\alpha \leqslant \beta$ and $\alpha \neq \beta$. We will say that α and β are *comparable* if either $\alpha \leqslant \beta$ or $\beta \leqslant \alpha$ holds.

2.1. Theories

A set of Boolean vectors (also called *models*) in $\{0, 1\}^n$ is called a *theory* (or a *Boolean function* $\{0, 1\}^n \to \{0, 1\}$, identified with its set of *true* points, i.e., the points assigned the value 1), and it will usually be represented by an upper case Greek letter like Σ. We will denote by $\Sigma[X]$ a theory Σ restricted to the variables in X. The number of models of a theory Σ will be denoted by $|\Sigma|$.

We shall call a disjunction of literals a *clause*, and in many cases will not distinguish between a clause and the set of literals it contains. A clause C is said to *subsume* a clause C' if C' contains all the literals in C. It is well known that any theory can be represented as a conjunction of clauses called *conjunctive normal form* (CNF). In some cases, we will not make a distinction between a CNF and the theory it represents. The *length* of a CNF $\mathcal{F}$ (i.e., the number of literals in it) will be denoted by $|\mathcal{F}|$. A CNF is called *irredundant* if the removal of any clause from it results in a CNF that does not represent the same theory.

A clause C is called an *implicate* of a theory Σ if its set of models contains Σ, and this will be denoted as $\Sigma \models C$. Clearly, each clause of a CNF is an implicate of the theory represented by the CNF. A clause C is called a *prime* implicate of a theory Σ if $\Sigma \models C$ and there is no distinct clause C' such that $\Sigma \models C' \models C$ (in other words, Σ does not have a distinct implicate C' that subsumes C). A CNF consisting only of prime implicates of the theory it represents is called *prime*.

A clause containing a single literal is called a *unit* clause, while a clause containing two literals will be called *quadratic*. It can be seen easily that, for any non-empty theory Σ, if a unit clause is an implicate of Σ, then (i) it is a prime implicate of Σ, (ii) no other prime implicate of Σ involves the variable of this clause, and (iii) all the models of Σ have the same value in the variable of this clause. In other words, a unit implicate means that

Σ is degenerate in the variable of the unit clause, and without loss of generality we shall assume from now on that all theories considered in this paper do not have unit implicates. Clearly, if a theory has no unit implicate, then every quadratic implicate of such a theory is prime.

If an arbitrary theory is given by its set of models, all its unit implicates correspond to the constant-zero or constant-one columns. If a Boolean theory is given by a CNF, it is NP-hard to check whether it has any unit implicates. This, however, will not present any problems in the context of this paper, since the discussion here will be focused on the Horn theories, and the discovery of unit implicates, if any, of a Horn CNF $\mathcal{F}$ can be accomplished in $\mathrm{O}(|V||\mathcal{F}|)$ time. [3]

2.2. Functional dependencies

For two subsets of variables $X, Y \subseteq V$, an expression $X \to Y$, called a *functional dependency*, means that the values of the variables in X determine the values of the variables in Y. A functional dependency $X \to Y$ is said to *hold* in a theory Σ if, for any $\alpha, \beta \in \Sigma$ such that $\alpha[X] = \beta[X]$, it holds that $\alpha[Y] = \beta[Y]$. Obviously, functional dependencies are monotone with respect to set inclusion. More precisely, if a functional dependency $X \to Y$ holds in a theory Σ, then a functional dependency $X' \to Y'$ also holds in Σ for any $X' \supseteq X$ and any $Y' \subseteq Y$.

Since a functional dependency $X \to Y$ holds in a theory Σ if and only if the functional dependency $X \to y$ holds in Σ for every $y \in Y$, without loss of generality, we will restrict our attention to functional dependencies of type $X \to y$. A functional dependency $X \to y$ in Σ states that the variable y is a Boolean function of the variables in X (i.e., $y = f(X)$).

Theorem 2.1. *Given a theory Σ, one can check in $\mathrm{O}(|V||\Sigma|)$ time whether a functional dependency $X \to y$ holds in Σ.*

Proof. To check whether $X \to y$ holds in Σ, we construct a binary decision tree using all the variables in X one by one for branching at the decision nodes. The root of the tree contains all the points in Σ. Each node of the tree contains the set of points in Σ which have exactly the same values in the variables along the path from the root to the node. The branching stops when a node has no points, or when all the variables in X have already been used for branching. The functional dependency $X \to y$ does not hold in Σ if and only if there exists a leaf in the resulting tree which contains points in Σ that have the opposite values in y. Since the number of leaves of the tree does not exceed $2|\Sigma|$, this can be checked in $\mathrm{O}(|\Sigma|)$ time. Since the depth of the tree does not exceed $|X|$, the construction of the tree can be done in $\mathrm{O}(|V||\Sigma|)$ time, and therefore the total time needed to check whether $X \to y$ holds in Σ is $\mathrm{O}(|V||\Sigma|)$. □

On the other hand, if a theory is represented by a CNF, the problem of checking whether a functional dependency holds in the theory becomes difficult.

[3] In the following we use the notation $\phi = \mathrm{O}(\psi)$ to denote that there exists a constant c such that $\phi \leqslant c\psi$.

Theorem 2.2. *Given a CNF $\mathcal{F}$ and a functional dependency $X \to y$, it is* CoNP-*complete to check whether this functional dependency holds in the theory represented by $\mathcal{F}$.*

Proof. The problem is obviously in CoNP, since if the functional dependency $X \to y$ does not hold in the theory represented by $\mathcal{F}$, it can be demonstrated by two points α and β that satisfy $\mathcal{F}$ and $\alpha[X] = \beta[X]$ and $\alpha[y] \neq \beta[y]$.

We shall now show that this problem is hard by a reduction from the satisfiability problem, which is known to be NP-complete. Given an arbitrary CNF $\mathcal{F}'$, in order to check whether it is satisfiable (i.e., whether $\mathcal{F}'$ has a model), we introduce two new variables x_0 and y_0, and create a new CNF $\mathcal{F} = \mathcal{F}' \wedge (\overline{x}_0 \vee y_0)$. We claim that the given CNF $\mathcal{F}'$ is satisfiable if and only if the dependency $x_0 \to y_0$ does not hold in the theory represented by $\mathcal{F}$. Indeed, each model of $\mathcal{F}'$ corresponds to three models of $\mathcal{F}$ with the variables (x_0, y_0) taking the values $(0,0)$, $(0,1)$, and $(1,1)$, respectively. The first two combinations show that the dependency $x_0 \to y_0$ does not hold. On the other hand, if $\mathcal{F}'$ has no models, $x_0 \to y_0$ trivially holds. □

A functional dependency $X \to y$ in Σ is *minimal* if there is no $X' \subset X$ such that the functional dependency $X' \to y$ holds in Σ. Because of monotonicity, it is essential to know only the set of minimal functional dependencies in Σ, which will be denoted by $\mathcal{M}(\Sigma)$.

We shall call functional dependencies with a single variable in the left-hand side *simple*. Since we consider theories without unit implicates, any simple functional dependency that holds in a theory is minimal. Moreover, if a functional dependency $x \to y$ holds in Σ, then the functional dependency $y \to x$ must also hold in Σ, because a Boolean function of a single variable, which is not a constant, can be either an identity ($y = x$), or its negation ($y = \overline{x}$). This implies the following statements, in which we use simplified notations such as $xyZ \to w$ to mean $\{x, y\} \cup Z \to w$.

Lemma 2.3. *If a simple functional dependency $x \to y$ holds in a theory Σ, then*
- *the functional dependency $y \to x$ must also hold in Σ,*
- *no minimal functional dependency in Σ has the form $xyZ \to w$,*
- *$xZ \to w$ is a minimal functional dependency in Σ if and only if $yZ \to w$ is a minimal functional dependency in Σ, and*
- *$Z \to x$ is a minimal functional dependency in Σ if and only if $Z \to y$ is a minimal functional dependency in Σ.*

The statements of the lemma (except for the second one) require the assumption that Σ does not have unit implicates. As was remarked above, if Σ does have a unit implicate, say x or $\overline{x}$, then the variable x is degenerate in the sense that all the models of Σ have the same value in x. In this case, for every variable $y \in V$ the functional dependency $y \to x$ obviously holds in Σ, and no other minimal functional dependency in Σ involves x. This implies that without loss of generality the study of functional dependencies can be restricted to theories without unit implicates.

The proof of Theorem 2.2 shows that if a theory Σ is represented by a CNF, then it is difficult to check whether even a simple functional dependency holds in Σ.

The fact that certain functional dependencies hold in a theory may imply that other functional dependencies must also hold in the same theory. More precisely, a functional dependency is said to be *implied* by a set of functional dependencies if it can be derived from these dependencies by the repetitive application of the following Armstrong's rules of inference (see [1,33]):

(1) *inclusion rule*: if $X \subseteq Y$, then $Y \to X$;
(2) *augmentation rule*: if $X \to Y$, then $XZ \to Y$ for any set Z; and
(3) *transitivity rule*: if $X \to Y$ and $Y \to Z$, then $X \to Z$.

The set of all the functional dependencies implied by the functional dependencies in a set $\mathcal{D}$ will be called the *closure* of $\mathcal{D}$, and will be denoted by $\widehat{\mathcal{D}}$. For a theory Σ, a set of minimal functional dependencies $\mathcal{D}$ will be called an *FD-cover* of Σ if its closure $\widehat{\mathcal{D}}$ is the set of all the functional dependencies that hold in Σ. An FD-cover of Σ is called *irredundant* if no proper subset of it is an FD-cover of Σ. Clearly, for any Σ, the set $\mathcal{M}(\Sigma)$ is an FD-cover of Σ. However, it is possible that a subset of $\mathcal{M}(\Sigma)$ also provides an FD-cover of Σ.

For a set of functional dependencies $\mathcal{D}$, a theory Σ is called an *Armstrong relation* for $\mathcal{D}$ if the set of all the functional dependencies that hold in Σ coincides with the closure $\widehat{\mathcal{D}}$. The concept of Armstrong relations is very important in the theory of relational databases, and has been well studied (see [1,2,28,34]). It is known that, for any set of functional dependencies $\mathcal{D}$, there exists an Armstrong relation. However, such relation may not be Boolean. If we restrict the set of relations to theories in $\{0,1\}^n$, there are sets of functional dependencies $\mathcal{D}$ for which there is no Armstrong relation. For example, let us consider $\mathcal{D} = \{x \to y, yz \to w\}$. Since $\mathcal{D}$ does not contain $y \to x$, all the models of any Armstrong relation of $\mathcal{D}$ should have the same value in y. Therefore, $yz \to w$ implies that $z \to w$ must also hold, and since it is not in $\mathcal{D}$, this set of functional dependencies has no Armstrong relation among general Boolean theories.

2.3. Condensation

If a functional dependency $X \to y$ holds in a theory Σ, then the value of y is redundant in every model of Σ in the sense that y can be determined from X, i.e., by a Boolean function $y = f(X)$. It therefore may be beneficial to "reduce" Σ by eliminating y and considering instead the theory $\Sigma[V \setminus y]$. If the description of the function f is preserved, then this reduction will not result in any loss of information. The reduced theory $\Sigma[V \setminus y]$ is simpler to work with since it has fewer variables, and its structure is not complicated in any way by this reduction. Moreover, this reduced theory will have fewer functional dependencies than the original one, since, as can be seen easily, its set of functional dependencies consists of those and only those dependencies that hold in Σ and do not involve y.

If the theory $\Sigma[V \setminus y]$ still has some functional dependencies, then the reduction procedure can be repeated. We shall call *condensation* the iterative application of the reduction procedure until the resulting theory has no functional dependencies. The resulting theory Σ^c, which has no functional dependencies, will be called a *condensation* of Σ. Generally, a theory that does not have any functional dependencies will be called *condensed*.

The condensation procedure does not specify which functional dependency to use for reducing a theory at each step, and if the theory has several functional dependencies, one will be chosen arbitrarily. Therefore, the result of the condensation procedure is generally non-deterministic, and a theory may be condensed into many different ones.

The reduction procedure described above is similar to the *normalization process* which is routinely used in the logical design of relational databases. The peculiarity of the normalization process consists in the fact that the description of function f is preserved in the form of a relation. In a knowledge-based system this function can be stored in other ways (i.e., as a clausal form, a formula, a decision tree, etc.). It may happen, however, that the structure of this function f is complicated. Then the task of preserving and using this functional description may be far from trivial. This complication, if it happens, may offset the benefits of reduction and may even impose some computational penalties. This problem manifests itself in the practice of relational databases, where *denormalization* is commonly used to speed up the database performance.

2.4. Horn theories

A clause is called *Horn* if it contains at most one positive literal. Clauses containing exactly one positive literal are called *definite*, while clauses containing no positive literals are called *negative*. A CNF is called *Horn* if it contains only Horn clauses. A CNF containing only negative clauses will be called *negative*, while a CNF containing only definite clauses will be called *definite*. A theory is called *Horn* if there exists a Horn CNF representing it. It is known (see [17,18]) that every prime implicate of a Horn theory is Horn, and therefore any prime CNF of a Horn theory is Horn. The most important property of Horn CNFs is the linear time complexity of the *satisfiability problem* (see [13]), i.e., the problem of checking whether the theory represented by the CNF contains at least one model. Based on this, for a given Horn CNF and any clause, it can be checked in linear time whether this clause is an implicate of the CNF, and if yes, a prime implicate subsuming this clause can be found easily.

For two points $\alpha, \beta \in \{0,1\}^n$, the point γ defined by $\gamma_i = \alpha_i \wedge \beta_i$, $i = 1, 2, \ldots, n$, will be called the intersection of α and β and denoted by $\alpha \cap \beta$. It is well known (see [11,36]) that a theory is Horn if and only if it is closed under intersection, i.e., $\alpha, \beta \in \Sigma$ imply $\alpha \cap \beta \in \Sigma$. This property leads to an alternative way of representing Horn theories, i.e., a Horn theory can be represented by a subset of its models which has the property that all the other models can be obtained as intersections of some models in the subset. The smallest such subset is called the *characteristic set* [24,27,29]. For an arbitrary theory Σ, its intersection closure is called the *Horn envelope* of Σ and is denoted by $H(\Sigma)$ (see [26]). Clearly, $H(\Sigma)$ is the minimum Horn superset of Σ; i.e., for any Horn theory $\Sigma' \supseteq \Sigma$, it holds that $H(\Sigma) \subseteq \Sigma'$. In this paper, we shall consider a Horn theory $\Sigma \subseteq \{0,1\}^n$ that is represented either by a Horn CNF or by a subset Σ' of Σ satisfying $\Sigma = H(\Sigma')$.

It is natural to establish a correspondence between functional dependencies and Horn clauses by introducing for a functional dependency $X \to y$ the definite Horn clause $y \vee \bigvee_{x \in X} \overline{x}$. This correspondence has been well studied (see [12,14,15,39]), and has been shown to establish the following equivalence between a set of functional dependencies $\mathcal{D}$ and its corresponding definite Horn CNF $\mathcal{F}$: a functional dependency $X \to y$ is implied by $\mathcal{D}$ if and only if the definite Horn clause $y \vee \bigvee_{x \in X} \overline{x}$ is an implicate of $\mathcal{F}$. Therefore, a set of functional dependencies can be naturally interpreted as a Horn CNF. In what follows we will occasionally make no distinction between a set of functional dependencies and the corresponding definite Horn CNF. Since a Horn CNF represents a Horn theory (or a Horn Boolean function, see [17,18]), we shall call the Horn theory represented by the set of functional dependencies holding in a theory Σ the *associated* Horn theory of Σ, and view $\mathcal{M}(\Sigma)$ as the set of all prime implicates of this associated Horn theory. This paper is devoted to studying various properties of $\mathcal{M}(\Sigma)$ when Σ itself is a Horn theory.

2.5. Examples

To illustrate the concepts introduced in this section, let us consider a theory Σ, which is the set of row vectors in the following matrix Γ:

$$\Gamma = \begin{pmatrix} x & y & z & w \\ \hline 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}.$$

In the following, we will sometimes write $\Sigma = \Gamma$ if Σ is the set of row vectors in the matrix Γ. One can check that Σ can be represented by the following CNF:

$$(x \vee w)(y \vee z)(\overline{w} \vee y)(\overline{w} \vee z)(\overline{x} \vee \overline{y} \vee \overline{z}).$$

Furthermore, using the decision tree construction described in the proof of Theorem 2.1, it can be verified that the following set of functional dependencies is an FD-cover of Σ:

$$\mathcal{D} = \{x \to w, w \to x, xy \to z, yz \to x, xz \to y\}.$$

The set of all minimal functional dependencies that hold in Σ is then given by:

$$\mathcal{M}(\Sigma) = \mathcal{D} \bigcup \{wy \to z, yz \to w, wz \to y\}.$$

The condensation of Σ using the sequence of functional dependencies $\{x \to w, xy \to z\}$ results in the condensed theory Σ_1^c, while the condensation of Σ using the sequence of functional dependencies $\{w \to x, wz \to y\}$ results in the condensed theory Σ_2^c:

$$\Sigma_1^c = \begin{pmatrix} x & y \\ \hline 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \Sigma_2^c = \begin{pmatrix} z & w \\ \hline 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

In the condensation of Σ_1^c the eliminated variables have the following expressions: $w = \overline{x}$, $z = \overline{x} \vee \overline{y}$, while in the condensation of Σ_2^c the eliminated variables have the following expressions: $x = \overline{w}$, $y = \overline{z} \vee w$.

It can be seen easily that the above theory Σ is not Horn; its Horn envelope $H(\Sigma)$ is shown in Fig. 1.

The Horn theory $H(\Sigma)$ can be represented by the following Horn CNF:

$$(\overline{x} \vee \overline{y} \vee \overline{z})(\overline{y} \vee \overline{z} \vee w)(\overline{w} \vee y)(\overline{w} \vee z).$$

$$H(\Sigma) = \begin{pmatrix} x & y & z & w \\ \hline 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad H(\Sigma)^c = \begin{pmatrix} x & y & z \\ \hline 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

Fig. 1. Horn envelope $H(\Sigma)$ and its condensation.

The set of all minimal functional dependencies that hold in $H(\Sigma)$ coincides with the FD-cover of $H(\Sigma)$ and consists of a single functional dependency:

$$\mathcal{M}(H(\Sigma)) = \mathcal{D} = \{yz \to w\}.$$

The theory $H(\Sigma)$ has a unique condensation which is shown in Fig. 1. The eliminated variable has the following expression: $w = yz$. The condensed theory $H(\Sigma)^c$ can be represented by a CNF consisting of a single negative clause:

$$\overline{x} \vee \overline{y} \vee \overline{z}.$$

3. Recognizing functional dependencies in Horn theories

The most basic problem about functional dependencies in Horn theories is the *recognition problem*, i.e., given a Horn theory Σ and a functional dependency $X \to y$, check whether this functional dependency holds in Σ. It was remarked in Section 2.2 (see Theorem 2.2) that in the case of general Boolean theories the computational complexity of the recognition problem depends on how the theory is represented.

We will consider first the case in which a Horn theory is represented by a Horn CNF.

Theorem 3.1. *Given a Horn CNF $\mathcal{F}$ and a functional dependency $X \to y$, it can be checked in* $O(|X||\mathcal{F}|)$ *time whether this functional dependency holds in the theory represented by $\mathcal{F}$.*

Proof. Let Σ be the theory represented by $\mathcal{F}$. The functional dependency $X \to y$ does not hold in Σ if and only if there exist $\alpha, \beta \in \Sigma$ such that $\alpha[X] = \beta[X]$ and $\alpha[y] \neq \beta[y]$. Let us introduce a new variable z' for every $z \in V \setminus (X \cup y)$, and let us denote by $\mathcal{F}'$ the CNF obtained from $\mathcal{F}$ by substituting $\overline{y}$ for y and z' for z, for every $z \in V \setminus (X \cup y)$. The CNF $\mathcal{F}'$ can be constructed in $O(|X||\mathcal{F}|)$ time by using an $O(|X|)$ time procedure for checking whether $z \in X$. It can be seen easily that the functional dependency $X \to y$ does not hold in Σ if and only if the CNF $\mathcal{F} \wedge \mathcal{F}'$ is satisfiable, i.e., there exists a solution to the following equation:

$$\mathcal{F} \wedge \mathcal{F}' = 1. \tag{1}$$

The satisfiability problem (1) may not be Horn because of the substitution of $\overline{y}$ for y in $\mathcal{F}'$. It can, however, be solved in $O(|\mathcal{F}|)$ time, since it is equivalent to the Horn satisfiability problem obtained by substituting $y = 1$ in (1). Indeed, let $\mathcal{F}_1$ and $\mathcal{F}_0$ be the CNFs obtained from (1) by substituting $y = 1$ and $y = 0$, respectively. Then (1) has a solution if and only if at least one of these two Horn satisfiability problems has a solution. One can easily see that $\mathcal{F}_1$ can be obtained from $\mathcal{F}_0$ by substituting $z := z'$ and $z' := z$ for all $z \in V \setminus (X \cup y)$. This means that $\mathcal{F}_1$ has a solution if and only if so does $\mathcal{F}_0$, which is also equivalent to the condition that (1) has a solution. Thus, the linear time algorithm for the Horn satisfiability problem (see [13]) can be employed to construct an $O(|X||\mathcal{F}|)$ time algorithm for checking whether a functional dependency holds in the theory represented by a Horn CNF. □

Corollary 3.2. *Given a Horn CNF $\mathcal{F}$ and a functional dependency $X \to y$ in the theory represented by $\mathcal{F}$, it can be checked in $O(|X|^2|\mathcal{F}|)$ time whether this functional dependency is minimal.*

Proof. The procedure consists in removing variables from X one by one and checking whether the resulting functional dependency still holds. If the functional dependency is not minimal, a minimal one will be produced as a by-product of this procedure. □

Let us consider next the case in which we are given a set of models Σ. We would like to check whether a given functional dependency holds in the Horn envelope $H(\Sigma)$. The following lemma provides a structural characterization important for this situation. Let Σ_0^x denote the set of all the models of Σ that have the value 0 in x, i.e.,

$$\Sigma_0^x = \{\alpha \mid \alpha \in \Sigma,\ \alpha[x] = 0\}. \tag{2}$$

Similarly,

$$\Sigma_1^x = \{\alpha \mid \alpha \in \Sigma,\ \alpha[x] = 1\}. \tag{3}$$

Since we limit our attention to theories without unit implicates, both Σ_0^x and Σ_1^x are non-empty for all variables $x \in V$.

Lemma 3.3. *A functional dependency $X \to y$ holds in the Horn envelope $H(\Sigma)$ of a theory Σ if and only if there exists a subset $X' \subseteq X$ such that the following two conditions hold*:

(1) *all the points α in Σ_1^y satisfy $\alpha[X'] = (11\ldots1)$, and*
(2) *for every point $\alpha \in \Sigma_0^y$, there exists $x \in X'$ such that $\alpha[x] = 0$.*

Proof. Let us denote by α_1 the point of $H(\Sigma)$ obtained by the intersection of all the points in Σ_1^y. We will show below that the functional dependency $X \to y$ does not hold in $H(\Sigma)$ if and only if there exists a point $\beta_0 \in \Sigma_0^y$ such that $\alpha_1[X] \leqslant \beta_0[X]$.

The functional dependency $X \to y$ does not hold in $H(\Sigma)$ if and only if there exist $\alpha, \beta \in H(\Sigma)$ such that $\alpha[X] = \beta[X]$ and $\alpha[y] \neq \beta[y]$. We can assume that $\alpha[y] = 1$ and $\beta[y] = 0$. By the closure property of $H(\Sigma)$, α must have been obtained by the intersection of some points in Σ_1^y, and therefore $\alpha_1 \leqslant \alpha$. The point β must have been obtained by the intersection of some points in Σ, with at least one point β_0 from Σ_0^y used in this intersection. Obviously, $\beta \leqslant \beta_0$. Then

$$\alpha_1[X] \leqslant \alpha[X] = \beta[X] \leqslant \beta_0[X].$$

Conversely, if $\alpha_1[X] \leqslant \beta_0[X]$, then the point $\beta = \alpha_1 \cap \beta_0$ satisfies $\beta[X] = \alpha_1[X]$ and $\beta[y] = 0$, while, obviously, $\alpha_1[y] = 1$.

Now there does not exist $\beta_0 \in \Sigma_0^y$ such that $\alpha_1[X] \leqslant \beta_0[X]$ if and only if for every $\beta \in \Sigma_0^y$ there exists a coordinate $x \in X$ such that $\beta[x] = 0$ and $\alpha_1[x] = 1$ (since $\Sigma_0^y \neq \emptyset$ holds by the assumption that no unit clause exists in $H(\Sigma)$).

By construction, if $\alpha_1[x] = 1$ then for every point $\alpha \in \Sigma_1^y$ we have $\alpha[x] = 1$. Finally, let $X' = \{x \in X \mid \alpha_1[x] = 1\}$. Then it is easy to see that $\alpha_1[X] \leqslant \beta_0[X]$ holds if and only if conditions (1) and (2) of the lemma hold. □

Corollary 3.4. *A functional dependency $X \to y$ is minimal in the Horn envelope $H(\Sigma)$ of a theory Σ if and only if the following three conditions hold*:
(1) *all the points α in Σ_1^y satisfy $\alpha[X] = (11\ldots1)$,*
(2) *for every point $\alpha \in \Sigma_0^y$, there exists $x \in X$ such that $\alpha[x] = 0$, and*
(3) *for every $x \in X$, there exists a point $\alpha \in \Sigma_0^y$ such that $\alpha[x] = 0$ and $\alpha[X \setminus x] = (11\ldots1)$.*

Proof. The first two conditions are the conditions of Lemma 3.3, and the third condition states that the removal of any variable from X results in a functional dependency that violates the second condition. □

Remark that in the case of a simple functional dependency we have $X \setminus x = \emptyset$, and therefore condition (3) of the corollary trivially holds for simple functional dependencies.

Since $\Sigma \models H(\Sigma)$, any functional dependency that holds in $H(\Sigma)$ also holds in Σ. On the other hand, there may exist functional dependencies that hold in Σ and do not hold in $H(\Sigma)$. Interestingly, Corollary 3.4 implies that a minimal functional dependency in $H(\Sigma)$ is also a minimal functional dependency in Σ.

As was noted in Section 2.2, one can check quickly whether a functional dependency $X \to y$ holds in a theory Σ, if all the models of Σ are given. However, since the Horn envelope $H(\Sigma)$ may contain a number of models which is exponential in $|\Sigma|$, this result cannot be applied directly to recognizing whether $X \to y$ holds in $H(\Sigma)$. As a corollary of the structural characterization in Lemma 3.3, we get the following result showing how to check fast whether $X \to y$ holds in $H(\Sigma)$.

Theorem 3.5. *Given a theory Σ and a functional dependency $X \to y$, it can be checked in* $\mathrm{O}(|V||\Sigma|)$ *time whether this functional dependency holds in the Horn envelope $H(\Sigma)$.*

Proof. Lemma 3.3 implies that the following linear time algorithm checks whether the dependency $X \to y$ holds in $H(\Sigma)$.
(1) Split Σ into Σ_0^y and Σ_1^y.
(2) Determine the maximum subset $X' \subseteq X$ such that $\alpha[X'] = (11\ldots1)$ for all $\alpha \in \Sigma_1^y$.
(3) For every $\beta \in \Sigma_0^y$, check whether there exists an $x \in X'$ such that $\beta[x] = 0$.

The dependency $X \to y$ does not hold in $H(\Sigma)$ if and only if there exists a $\beta \in \Sigma_0^y$ such that $\beta[X'] = (11\dots1)$. $\square$

Corollary 3.6. *Given a theory Σ and a functional dependency $X \to y$, it can be checked in* $\mathrm{O}(|V||\Sigma|)$ *time whether this functional dependency is minimal in the Horn envelope* $H(\Sigma)$.

Proof. Corollary 3.4 shows that checking the minimality consists in simply maintaining for each $x \in X$ an indicator bit whose value is initialized at 0 and set to 1 whenever step (3) of the algorithm described in the proof of Theorem 3.5 encounters some $\beta \in \Sigma_0^y$ such that $\beta[x] = 0$ and $\beta[X \setminus x] = (11\dots1)$. $\square$

Interestingly, for a model representation, checking the minimality of a functional dependency does not result in any discernible increase in the computing time as compared with checking whether the dependency holds. By contrast, Theorem 3.1 and Corollary 3.2 suggest that the computing time for a CNF representation does increase, although marginally.

4. Structure of functional dependencies in Horn theories

We will analyze in this section structural properties of the set of minimal functional dependencies that hold in an arbitrary Horn theory. We start this analysis by establishing a connection between minimal functional dependencies in a Horn theory and certain prime implicates of that theory.

Theorem 4.1. *A functional dependency $X \to y$ holds and is minimal in a Horn theory Σ if and only if all clauses $\overline{y} \vee x$, $x \in X$, and the clause $y \vee \bigvee_{x \in X} \overline{x}$ are prime implicates of Σ.*

Proof. Corollary 3.4 implies that a functional dependency $X \to y$ is minimal in a Horn theory Σ if and only if the following three conditions hold:

(1) for every model $\alpha \in \Sigma$ with $\alpha[y] = 1$, we have $\alpha[X] = (11\dots1)$;
(2) for every model $\alpha \in \Sigma$ with $\alpha[y] = 0$, there exists an $x \in X$ such that $\alpha[x] = 0$, and
(3) for every $x \in X$, there exists a model $\alpha \in \Sigma$ such that $\alpha[y] = 0$, $\alpha[x] = 0$ and $\alpha[X \setminus x] = (11\dots1)$.

Let us first discuss the "only if" part of the theorem. Condition (1) states that, for every $x \in X$, the clause $\overline{y} \vee x$ is an implicate of Σ. Since we consider only theories without unit implicates, every quadratic implicate is prime.

Condition (2) states that the clause $y \vee \bigvee_{x \in X} \overline{x}$ is an implicate of Σ, and condition (3) states that the removal of any literal $\overline{x}$ results in a clause which is not an implicate of Σ. Furthermore, the clause $\bigvee_{x \in X} \overline{x}$ is not an implicate of Σ, since otherwise $\overline{y}$ would be an implicate of Σ (implied by $(\bigvee_{x \in X} \overline{x}) \wedge \bigwedge_{x \in X} (\overline{y} \vee x)$), contradicting the assumption that Σ has no unit implicate. Therefore, $y \vee \bigvee_{x \in X} \overline{x}$ is prime.

Let us now discuss the "if" part. The fact that the clause $\overline{y} \vee x$ is an implicate of Σ for every $x \in X$ implies condition (1). The fact that the clause $y \vee \bigvee_{x \in X} \overline{x}$ is an implicate of

Σ implies condition (2). The fact that for every $x \in X$ the clause $y \vee \bigvee_{x' \in X \setminus x} \overline{x'}$ is not an implicate of Σ implies condition (3). $\square$

Note that this proof can be modified (using Lemma 3.3 instead of Corollary 3.4) to show the following result.

Theorem 4.2. *A functional dependency $X \to y$ holds in a Horn theory Σ if and only if all clauses $\overline{y} \vee x$, $x \in X$, and the clause $y \vee \bigvee_{x \in X} \overline{x}$ are implicates of Σ.*

Note that Theorems 4.1 and 4.2 provide a slightly different (from Theorem 3.1 and Corollary 3.2) way of checking whether a functional dependency $X \to y$ holds (and is minimal) in the theory represented by a Horn CNF $\mathcal{F}$. It consists in simply checking (e.g., as described in [17,18]) whether $y \vee \bigvee_{x \in X} \overline{x}$ and $\overline{y} \vee x$, for all $x \in X$, are (prime) implicates of $\mathcal{F}$.

Corollary 4.3. *For a Horn theory Σ, the set of models of $\mathcal{M}(\Sigma)$ is a superset of Σ.*

Corollary 4.4. *If a functional dependency $X \to y$ is minimal in a Horn theory Σ, then, for every $\alpha \in \Sigma$, we have $\alpha[y] = \bigwedge_{x \in X} \alpha[x]$.*

This corollary states that minimal functional dependencies in Horn theories always take the functional form of single positive term Boolean functions (i.e., $y = \bigwedge_{x \in X} x$ for some X), while minimal functional dependencies in general Boolean theories can be arbitrary Boolean functions without redundant[4] variables. To show the latter statement, for any Boolean function without redundant variables $f(X)$, with $|X| = n$, we construct a theory Σ with $n+1$ variables $\{X, y\}$ and 2^n models obtained by adding to every $\alpha \in \{0,1\}^n$ the $(n+1)$st coordinate $y = f(\alpha)$. One can easily see that $X \to y$ is a minimal functional dependency in Σ.

Corollary 4.5. *A simple functional dependency $x \to y$ holds in a Horn theory Σ if and only if the variables x and y are logically equivalent in Σ, i.e., $\alpha[x] = \alpha[y]$ for all $\alpha \in \Sigma$.*

Note that in the case of general Boolean theories without unit implicates, if $x \to y$ holds, then either x and y are logically equivalent, or x and y are logically complementary, i.e., $\alpha[x] = \overline{\alpha[y]}$ for all models α.

To analyze the structure of functional dependencies in Horn theories, we shall associate to a set of of functional dependencies $\mathcal{D}$ a directed graph $G(\mathcal{D})$ whose set of vertices is the set of variables V, and an oriented arc $x \to y$ is in $G(\mathcal{D})$ if and only if the set $\mathcal{D}$ contains a functional dependency $X \to y$ such that $x \in X$. A similar construction was used in [38] in the study of unique Horn satisfiability, and in [19] for the compression of Horn knowledge bases. The following statement establishes a fundamental structural property of the graph $G(\mathcal{M}(\Sigma))$ of the set of all minimal functional dependencies in a Horn theory Σ.

[4] A variable is called *redundant* in a Boolean function if changing the value of only this variable never changes the value of the function. It is well known that almost all Boolean functions do not have redundant variables.

Theorem 4.6. *For a Horn theory Σ, the graph $G(\mathcal{M}(\Sigma))$ has an oriented cycle involving variables x and y if and only if the simple functional dependencies $x \to y$ and $y \to x$ hold in Σ.*

Proof. Theorem 4.1 implies that if the arc $x' \to y'$ is in $G(\mathcal{M}(\Sigma))$, then $\overline{y}' \vee x'$ is a prime implicate of Σ. Clearly, if $\overline{y}' \vee x'$ and $\overline{z}' \vee y'$ are implicates of Σ, then $\overline{z}' \vee x'$ is also an implicate of Σ. Therefore, the existence of an oriented path from x to y in $G(\mathcal{M}(\Sigma))$ implies that $\overline{y} \vee x$ is a prime implicate of Σ. Similarly, the existence of an oriented path from y to x implies that $\overline{x} \vee y$ is a prime implicate of Σ. Then, by Theorem 4.1, both $x \to y$ and $y \to x$ are minimal functional dependencies in Σ.

Conversely, if simple functional dependencies $x \to y$ and $y \to x$ hold in Σ, then they are minimal, and therefore $G(\mathcal{M}(\Sigma))$ contains both arcs $x \to y$ and $y \to x$. $\square$

Theorem 4.6 and Lemma 2.3 imply the following corollary.

Corollary 4.7. *For a Horn theory Σ, every strongly connected component of the graph $G(\mathcal{M}(\Sigma))$ is a complete directed graph, and any minimal non-simple functional dependency in Σ involves at most one variable from every strongly connected component of $G(\mathcal{M}(\Sigma))$.*

Theorem 4.6 and Corollary 4.5 imply that all the cycles in $G(\mathcal{M}(\Sigma))$ are due to the presence of logically equivalent variables in Σ. Intuitively, a group of logically equivalent variables can be replaced by a single variable without losing any essential information about a theory. This intuition was formalized in the procedure of 2-*condensation* introduced in [19] for the purpose of optimal compression of quasi-acyclic Horn knowledge bases. We call Horn theories without logically equivalent variables 2-*condensed*.

Given a Horn theory Σ, the procedure of 2-condensation constructs the 2-condensed Horn theory Σ^{2c} by replacing each group of logically equivalent variables with a single representative. For a variable $x \in V$, let us denote by $r(x)$ the representative of x in Σ^{2c}. Note that if x and y are logically equivalent in Σ, then $r(x) = r(y)$. Similarly, let $r(X)$ denote the set of representatives of a set of variables $X \subseteq V$. The following statement was proven in [19].

Proposition 4.8 (Hammer and Kogan [19]). *A definite Horn clause $y \vee \bigvee_{x \in X} \overline{x}$ is a prime implicate of a Horn theory Σ if and only if either $r(y) \vee \bigvee_{x \in X} \overline{r(x)}$ is a prime implicate of the 2-condensed theory Σ^{2c}, or $X = \{x\}$ and x is logically equivalent to y in Σ.*

Proposition 4.8 and Theorem 4.1 imply the following statement.

Corollary 4.9. *A functional dependency $X \to y$ is minimal in a Horn theory Σ if and only if either $r(X) \to r(y)$ is a minimal functional dependency in the 2-condensed Horn theory Σ^{2c}, or $X = \{x\}$ and x is logically equivalent to y in Σ.*

This corollary shows that for most purposes it is sufficient to study functional dependencies in 2-condensed Horn theories.

If Σ is represented by a Horn CNF $\mathcal{F}$, then a prime Horn CNF representing Σ^{2c} can be constructed in $O(|V||\mathcal{F}| + |\mathcal{F}|^2)$ time (see [19]). The procedure consists in inferring all the quadratic prime implicates of Σ, identifying groups of logically equivalent variables, reducing $\mathcal{F}$ to an equivalent prime CNF $\mathcal{F}'$, and replacing in $\mathcal{F}'$ equivalent variables from the same group with a single representative.

If Σ is represented by a set of models Σ' such that $H(\Sigma') = \Sigma$, then any two variables are logically equivalent in Σ if and only if the corresponding columns in Σ' are identical. Therefore, removing all but one columns from every group of identical columns in Σ' results in Σ'' such that $H(\Sigma'') = \Sigma^{2c}$. This 2-condensation can be easily done in $O(|\Sigma'||V|)$ by constructing a binary decision tree on the columns of Σ', using each row of Σ' one by one at the decision nodes, and then removing from Σ' all but one column from the group of columns of every leaf in the resulting tree.

The procedure of 2-condensation can be viewed as a restriction of the procedure of condensation introduced in Section 2.3, since 2-condensation is achieved if the procedure of condensation is applied using only simple functional dependencies. The condensation of Horn theories is discussed in detail in Section 6.

Corollary 4.5 implies the following statement.

Corollary 4.10. *No simple functional dependency holds in a* 2*-condensed Horn theory.*

Theorem 4.6 and Corollary 4.10 imply the following important structural property of functional dependencies in 2-condensed Horn theories.

Theorem 4.11. *For a* 2*-condensed Horn theory* Σ^{2c}*, the graph* $G(\mathcal{M}(\Sigma^{2c}))$ *contains no oriented cycles.*

The acyclicity of $G(\mathcal{M}(\Sigma^{2c}))$ is a very important structural property of the Horn theory represented by $\mathcal{M}(\Sigma^{2c})$. Such Horn theories are called *acyclic*. They were studied in [19], where it was proven that any acyclic Horn theory has a unique irredundant and prime CNF. In view of the equivalence of sets of functional dependencies and definite Horn CNFs, this result together with Theorem 4.11 immediately imply the next theorem.

Theorem 4.12. *The set of functional dependencies holding in a* 2*-condensed Horn theory has a unique irredundant FD-cover.*

While the set of functional dependencies holding in a 2-condensed Horn theory corresponds to an acyclic Horn theory, the set of functional dependencies holding in a general Horn theory corresponds to a *quasi-acyclic* Horn theory [5] (following the terminology of [19]). In this general case, the irredundant FD-cover will not be unique any more, but the results of [19] together with the presentation above show that all the irredundant FD-covers have essentially the same structure. An irredundant FD-cover of a Horn theory Σ consists of the unique irredundant FD-cover of the 2-condensed theory Σ^{2c} (with an arbitrary substitution of original variables for their representatives in Σ^c) and an

[5] A Horn theory is called quasi-acyclic if its 2-condensation is acyclic.

irredundant FD-cover of the set of simple functional dependencies in Σ. A minimum size irredundant FD-cover will be obtained when an irredundant FD-cover of the set of simple functional dependencies in Σ is chosen to consist of dependencies forming a single simple cycle in each group of logically equivalent variables.

5. Inferring functional dependencies in Horn theories

It follows from Corollary 4.9 that for the purpose of inferring functional dependencies holding in a Horn theory, we can assume that the theory is 2-condensed. Although the irredundant FD-cover of any 2-condensed Horn theory is unique, this FD-cover can be very large as compared with the length of the CNF representation of the theory.

Theorem 5.1. *For every $n \geqslant 2$, there exists a 2-condensed Horn theory of $2n+1$ variables, which has the CNF representation of size $\mathrm{O}(n)$ and the irredundant FD-cover of size*[6] $\Omega(2^n)$.

Proof. Consider the following Horn CNF:

$$\bigwedge_{i=1}^{n}(x_i \vee \overline{x}_0) \wedge \bigwedge_{i=1}^{n}(y_i \vee \overline{x}_i) \wedge \left(x_0 \vee \bigvee_{i=1}^{n} \overline{y}_i\right).$$

This CNF has $2n+1$ clauses and $5n+1$ literals. It can be checked that this is the unique irredundant prime CNF of the Horn theory it represents. It can also be checked that all its quadratic prime implicates except $(y_i \vee \overline{x}_0)$, $i = 1, 2, \ldots, n$, are contained in this CNF, and the Horn theory is 2-condensed. One can verify that in addition to the quadratic prime implicates, the theory has 2^n other prime implicates, each of which has the form $x_0 \vee \bigvee_{i=1}^{n} \overline{z}_i$, with $z_i \in \{x_i, y_i\}$. Then, by Theorem 4.1, the set of minimal functional dependencies in this Horn theory consists of all the dependencies of the form $\bigcup_{i=1}^{n} z_i \to x_0$, where $z_i \in \{x_i, y_i\}$. Every dependency of this form is needed in the irredundant FD-cover, since it is not implied by other dependencies of this form. Therefore, the irredundant FD-cover consists of 2^n minimal functional dependencies. $\square$

A similar result can also be shown when a 2-condensed Horn theory is represented by a set of models.

Theorem 5.2. *For every $n \geqslant 2$, there exists a theory Σ having size $|\Sigma| = \mathrm{O}(n)$ and depending on $2n+1$ variables, such that its Horn envelope $H(\Sigma)$ is a 2-condensed Horn theory whose irredundant FD-cover is of size $\Omega(2^n)$.*

Proof. Let us consider the following theory Σ which has $2n+1$ variables $x_1, x_2, \ldots, x_{2n+1}$ and $3n+1$ models. The variable x_{2n+1} has the value 1 only in the single model $(1111\ldots11)$. Among the remaining $3n$ models, there is a group of n models such that

[6] In the following we use the notation $\phi = \Omega(\psi)$ to denote that there exists a constant c such that $\phi \geqslant c\psi$.

in the ith model of this group the only variables that have the value 0 are x_{2i-1} and x_{2i}, in addition to x_{2n+1}. The remaining group of $2n$ models is such that in the ith model of this group the only variable that has the value 1 is x_i. Informally, Σ can be given by

$$\Sigma = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & 1 & 1 & \cdots & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix}.$$

Since all the columns of Σ are distinct, its Horn envelope $H(\Sigma)$ is 2-condensed. By the first statement of Lemma 3.3, no functional dependency holding in $H(\Sigma)$ can have variables $x_1, \ldots, x_{2n}$ in the right-hand side, since for every $i \leqslant 2n$ there exists a model in Σ in which the only variable that has the value 1 is x_i.

By the second statement of Lemma 3.3, any functional dependency holding in $H(\Sigma)$ (and therefore having x_{2n+1} in the right-hand side) should include in the left-hand side either x_{2i-1} or x_{2i} or both, since for every $i \leqslant n$ there exists a model in Σ in which the only variables that have the value 0 are x_{2i-1}, x_{2i}, and x_{2n+1}. Then, by the third statement of Corollary 3.4, no minimal functional dependency holding in $H(\Sigma)$ can include both x_{2i-1} and x_{2i} in the left-hand side. This implies that all the minimal functional dependencies holding in $H(\Sigma)$ are of the form $\bigcup_{i=1}^{n} y_i \to x_{2n+1}$, where $y_i \in \{x_{2i-1}, x_{2i}\}$. Every dependency of this form is needed in the irredundant FD-cover, since it is not implied by other dependencies of this form. Therefore, the irredundant FD-cover consists of 2^n minimal functional dependencies. □

Theorems 5.1 and 5.2 prove that the set of all minimal functional dependencies $\mathcal{M}(\Sigma)$ of a Horn theory Σ may be exponential in the length of the input, and hence cannot be generated in polynomial time. It is therefore important to look for an alternative object of smaller size that would capture some crucial information about $\mathcal{M}(\Sigma)$. An interesting aggregate description of this set is provided by generating for every variable y the set of variables $F_\Sigma(y)$ that take part in the minimal functional dependencies $X \to y$ in Σ:

$$F_\Sigma(y) = \{x \in V \mid \exists Z\colon\ xZ \to y \in \mathcal{M}(\Sigma)\}. \tag{4}$$

Clearly, the sets $F_\Sigma(y)$ provide a way of describing the graph $G(\mathcal{M}(\Sigma))$ introduced in Section 4.

We shall study below the computational complexity of generating $F_\Sigma(y)$ when a Horn theory is represented by a set of models and by a CNF. We first consider the Horn envelope representation.

Theorem 5.3. *Given a theory Σ, the set $F_{H(\Sigma)}(y)$ can be constructed in* $\mathrm{O}(|V||\Sigma|^2)$ *time for every variable $y \in V$.*

Proof. Let us define $X_y \subseteq V$ as the set of variables x such that all $\alpha \in \Sigma_1^y$ satisfy $\alpha[x] = 1$. By Corollary 3.4(1), we have $F_{H(\Sigma)}(y) \subseteq X_y$. Let us analyze now the requirement imposed by Corollary 3.4(2) and (3) on the points in $\Sigma_0^y[X_y]$. Note first that if $\Sigma_0^y[X_y]$ contains a point with all 1's, then $F_{H(\Sigma)}(y)$ is empty.

Note that if $X' \to y$ is a functional dependency in $H(\Sigma)$, then there exists $X'' \subseteq X'$ such that $X'' \to y$ is a minimal functional dependency in $H(\Sigma)$. Moreover, for any $x \in X'$ which has an $\alpha \in \Sigma_0^y$ such that $\alpha[x] = 0$ and $\alpha[X' \setminus x] = (11\ldots1)$, the set X'' must contain this x. It can then be shown that a variable x belongs to $F_{H(\Sigma)}(y)$ if and only if there exists an $\alpha \in \Sigma_0^y$ such that $\alpha[x] = 0$ and $\alpha[X_y] \not< \beta[X_y]$ for any other point β in Σ_0^y. Indeed, if such α exists, then let

$$X' = X_y \setminus \{z \in V \mid \alpha[z] = 0, z \neq x\}.$$

By Corollary 3.3, $X' \to y$ holds in $H(\Sigma)$, and, as discussed above, every $X'' \subseteq X'$, such that $X'' \to y$ is a minimal functional dependency in $H(\Sigma)$, must contain x. Conversely, let us assume by contradiction that for every $\alpha \in \Sigma_0^y$ such that $\alpha[x] = 0$, $\alpha[X_y] < \gamma[X_y]$ holds for some point γ in Σ_0^y. Note that under this assumption, we can always find γ's such that $\gamma[x] = 1$. Indeed, if $\gamma[x] = 0$, then, by our assumption applied to γ, there must exist γ' such that

$$\gamma'[X_y] > \gamma[X_y] > \alpha[X_y].$$

Let us also assume that $X \subseteq X_y$ contains x, and $X \to y$ is a minimal functional dependency in $H(\Sigma)$. Let $\alpha \in \Sigma_0^y$ be a point satisfying the requirement of Corollary 3.4(3). By assumption, there must exist $\beta \in \Sigma_0^y$ such that $\beta[X_y] > \alpha[X_y]$ and $\beta[x] = 1$. Then this β must violate the requirement of Corollary 3.4(2), contradicting the assumption that $X \to y$ is a minimal functional dependency in $H(\Sigma)$.

Consequently, the set $F_{H(\Sigma)}(y)$ can be constructed by the following procedure:

(1) Split Σ into Σ_0^y and Σ_1^y.
(2) Determine the subset X_y of all $x \in V$ such that $\alpha[x] = 1$ for every $\alpha \in \Sigma_1^y$.
(3) Remove from $\Sigma_0^y[X_y]$ every point α such that $\alpha < \beta$ holds for some point β in this set; denote the resulting set by $\widehat{\Sigma_0^y}[X_y]$.
(4) Place in $F_{H(\Sigma)}(y)$ every variable $x \in X_y$ such that there exists $\alpha \in \widehat{\Sigma_0^y}[X_y]$ for which $\alpha[x] = 0$.

Steps (1), (2) and (4) of this algorithm have linear time complexity. Obviously, step (3) can be completed in $\mathrm{O}(|V||\Sigma|^2)$ time. □

Theorem 5.3 shows that the sets $F_\Sigma(y)$, and therefore the graph $G(\mathcal{M}(\Sigma))$, are easily constructible if a Horn theory Σ is given by a set of models. We shall show next that this problem becomes computationally difficult if a Horn theory is represented by a Horn CNF.

Problem: *Horn-CNF-Aggregate-Set*
Instance: A Horn CNF $\mathcal{F}$ representing a Horn theory Σ, and two variables x and y.
Question: Does x belong to $F_\Sigma(y)$?

Theorem 5.4. *The Horn-CNF-Aggregate-Set problem is* NP-*complete.*

Proof. One can easily see that the Horn-CNF-Aggregate-Set problem belongs to NP. Indeed, if a set X containing x is given, checking whether $X \to y$ is a minimal functional dependency in Σ can be done in polynomial time, by Corollary 3.2.

To show that the problem is NP-hard, we shall polynomially reduce the following NP-complete problem (see [32]) to our problem.

Problem: *Prime-Attribute-Name*
Instance: A definite Horn CNF $\mathcal{F}$ in variables $x_1, \ldots, x_n$.
Question: Is there a negative prime implicate of $\mathcal{F}' = \mathcal{F} \wedge (\bigvee_{i=1}^{n} \overline{x}_i)$ containing x_1?

An instance of the Prime-Attribute-Name problem can be transformed to an instance of the Horn-CNF-Aggregate-Set problem in the following way. Let us consider the definite Horn CNF

$$\mathcal{F}'' = \mathcal{F} \wedge \left(y \vee \bigvee_{i=1}^{n} \overline{x}_i \right) \wedge \bigwedge_{i=1}^{n} (\overline{y} \vee x_i),$$

where y is a new variable. We argue that the Horn-CNF-Aggregate-Set problem for the input CNF $\mathcal{F}''$ representing a Horn theory Σ, and variables x_1 and y, is equivalent to the original instance of the Prime-Attribute-Name problem. Indeed, since $\mathcal{F}$ has no unit implicates, each $\overline{y} \vee x_i$ is prime. Therefore, by Theorem 4.1, the variable x_1 belongs to $F_\Sigma(y)$ if and only if Σ has a prime implicate of the form $y \vee \overline{x}_1 \vee \bigvee_{x \in X} \overline{x}$ for some X. Obviously, a clause $y \vee \overline{x}_1 \vee \bigvee_{x \in X} \overline{x}$ is an implicate of $\mathcal{F}''$ if and only if the clause $\overline{x}_1 \vee \bigvee_{x \in X} \overline{x}$ is an implicate of $\mathcal{F}'$. Therefore, a clause $y \vee \overline{x}_1 \vee \bigvee_{x \in X} \overline{x}$ is a prime implicate of $\mathcal{F}''$ if and only if the clause $\overline{x}_1 \vee \bigvee_{x \in X} \overline{x}$ is a prime implicate of $\mathcal{F}'$. This establishes the equivalence and completes the reduction. □

Note that the Horn-CNF-Aggregate-Set problem is closely related to the abduction problem of determining whether a given variable occurs in a minimal explanation (see [41] and [28]).

6. Condensation of Horn theories

The procedure of condensation introduced in Section 2.3 aims at simplifying a given theory by eliminating variables that are functionally dependent on other variables. In the case of general Boolean theories the simplification provided by condensation may come at a price. First of all, the functional dependencies used in condensation may have complicated structure which can make their storage and manipulation very expensive computationally. Second, the resulting condensed theory may depend on the choice of functional dependencies to be used in condensation. We will show in this section that the condensation of Horn theories does not present these problems.

The computational feasibility and benefits of condensing Horn theories stem from the fact that functional dependencies in any Horn theory always have a very simple structure. Corollary 4.4 states that a minimal functional dependency $X \to y$ is actually a single positive term Boolean function: $y = \bigwedge_{x \in X} x$. This functional description can be easily stored and manipulated computationally.

Since a set of points closed under intersection will remain closed under intersection after removing any variables, a condensation of any Horn theory will be Horn. Therefore, the procedure of condensation preserves the computationally advantageous Horn structure, and simplifies the theory by removing the variables whose values are essentially superfluous.

If a Horn theory is represented by a set of models, the representation of its condensation is obtained by removing the corresponding columns from the matrix, as was demonstrated in Section 2.5. As a result, for any theory Σ, its condensation using those functional dependencies that hold in the Horn envelope $H(\Sigma)$ will result in the condensed theory Σ^c such that $H(\Sigma^c) = H(\Sigma)^c$.

The condensation of a Horn theory represented by a Horn CNF is more involved. If a minimal functional dependency $X \to y$ is used in condensation, then $\bigwedge_{x \in X} x$ has to be substituted for y in all the clauses involving y. As a result, a Horn clause $\overline{y} \vee C$ will be transformed to the Horn clause $\bigvee_{x \in X} \overline{x} \vee C$, while a Horn clause $y \vee C$ will be transformed to the non-clausal expression $\bigwedge_{x \in X} x \vee C$. This expression, however, is equivalent to the Horn CNF:

$$\bigwedge_{x \in X} x \vee C = \bigwedge_{x \in X} (x \vee C). \tag{5}$$

Therefore, the resulting CNF will remain Horn.

The clause $\bigvee_{x \in X} \overline{x} \vee C$ may contain up to $|X| - 1$ more literals than the original clause $\overline{y} \vee C$, while the CNF $\bigwedge_{x \in X}(x \vee C)$ may have up to $|X|$ times as many literals as the original clause $y \vee C$. This observation might hint that the length of a Horn CNF could explode in the condensation procedure.

Let us denote by V_i the set of variables of the Horn theory Σ_i produced at the ith step of the condensation procedure.

Lemma 6.1. *For a Horn CNF $\mathcal{F}$, the Horn theory Σ_i produced at the ith step of the condensation procedure can be represented by a Horn CNF $\mathcal{F}_i$ whose length is limited by* $\mathrm{O}(|V_i|^2|\mathcal{F}|)$*, and this bound is sharp.*

Proof. Note that every variable in $V \setminus V_i$ can be expressed as a single positive term Boolean function of a subset of variables of V_i, since the superposition of positive terms is again a positive term:

$$\text{if } y_i = \bigwedge_{x \in X_i} x, \quad \text{then } z = \bigwedge_{i \in I} y_i = \bigwedge_{x \in \bigcup_{i \in I} X_i} x.$$

Therefore, instead of carrying out the condensation procedure step by step, we can achieve the same result if we first derive the expressions of all the variables of $V \setminus V_i$ through the variables of V_i, and only then substitute all these expressions directly in the original CNF $\mathcal{F}$ and carry out the expansion (5) to obtain the Horn CNF $\mathcal{F}_i$. Then one can see that the number of literals in each clause can increase by at most $\mathrm{O}(|V_i|)$ and the number of clauses can increase at most by a factor of $\mathrm{O}(|V_i|)$, resulting in the bound of the lemma.

To demonstrate that the length of a CNF can actually increase proportionally to the square of the number of variables, consider for example the following Horn CNF:

$$\mathcal{F} = \bigwedge_{i=1}^{n} (\overline{x}_0 \vee \overline{y}_i \vee z_0) \wedge \left(\bigvee_{i=1}^{n} \overline{z}_i \vee z_0 \right) \wedge \bigwedge_{i=1}^{n} (\overline{z}_0 \vee z_i) \wedge \left(\bigvee_{i=1}^{n} \overline{x}_i \vee x_0 \right) \wedge \bigwedge_{i=1}^{n} (\overline{x}_0 \vee x_i).$$

This CNF depends on $3n+2$ variables, has $9n+2$ literals, and can be seen to be irredundant and prime. It follows from Theorem 4.1 that $\bigcup_{i=1}^{n} z_i \to z_0$ and $\bigcup_{i=1}^{n} x_i \to x_0$ are minimal functional dependencies in the theory represented by $\mathcal{F}$. Using these dependencies to eliminate the variables z_0 and x_0 by condensation results in the following Horn CNF:

$$\mathcal{F}^c = \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{n} \left(\bigvee_{k=1}^{n} \overline{x}_k \vee \overline{y}_i \vee z_j \right).$$

This CNF depends on $3n$ variables, and its length is $n^3 + 2n^2$. One can see that this CNF is the unique irredundant prime CNF of the Horn theory it represents. □

It follows from Lemma 6.1 that the condensation procedure can be implemented to produce a Horn CNF of length at most $\mathrm{O}(|V|^2|\mathcal{F}|)$. Although there is a possibility (in the worst case) of a moderate polynomial increase in the length of a Horn CNF resulting from condensation, the next result proves that the potential reduction of the length of the CNF can be exponential.

Theorem 6.2. *For every $n \geqslant 2$, there exists a Horn theory of 2^n variables, whose minimum CNF representation is of size $\Omega(2^n)$ but whose condensation has a CNF representation of size $\mathrm{O}(n)$.*

Proof. For a Boolean vector $\alpha \in \{0,1\}^n$, let $||\alpha|| = \sum_{i=1}^{n} \alpha_i$. Let us now consider the following Horn CNF depending on 2^n variables $z, x_1, \ldots, x_n$, and all y_α, where $\alpha \in \{0,1\}^n$ and $||\alpha|| \geqslant 2$:

$$\mathcal{F} = \bigwedge_{\alpha \in \{0,1\}^n:\ ||\alpha|| \geqslant 2} \left(\bigvee_{i:\ \alpha_i = 1} \overline{x}_i \vee y_\alpha \right) \wedge \bigwedge_{\alpha \in \{0,1\}^n:\ ||\alpha|| \geqslant 2} \left(\bigwedge_{i:\ \alpha_i = 1} (\overline{y}_\alpha \vee x_i) \right) \wedge \left(\bigvee_{i=1}^{n} \overline{x}_i \vee z \right).$$

This CNF has $n2^{n-1} + 2^n - 2n$ clauses and $3n2^{n-1} + 2^n - 3n$ literals. It can be checked that $\mathcal{F}$ is irredundant and prime. Moreover, all the variables of the Horn theory Σ represented by $\mathcal{F}$ are irredundant, and therefore the minimum CNF representation of Σ is of size $\Omega(2^n)$.

It follows from Theorem 4.1 that all minimal functional dependencies of the form

$$\bigcup_{i:\ \alpha_i = 1} x_i \to y_\alpha,$$

where $\alpha \in \{0,1\}^n$ with $||\alpha|| \geqslant 2$, hold in Σ. The condensation of Σ using these functional dependencies results in the following Horn CNF:

$$\mathcal{F}^c = \bigvee_{i=1}^{n} \overline{x}_i \vee z,$$

which consists of a single clause having $n+1$ literals. $\square$

As was mentioned in Section 2.3 and demonstrated in Section 2.5, for general Boolean theories the result of the condensation procedure may depend on the order in which functional dependencies are used. We will show next that the quasi-acyclicity of structure of functional dependencies in Horn theories makes the condensation procedure essentially deterministic.

Simple functional dependencies, if hold at all in Σ, correspond to logically equivalent variables (see Corollary 4.5). The order of elimination of these variables may affect only which variable will be kept from each group of logically equivalent variables, and will not affect the results of condensation using non-simple functional dependencies, as obvious from Lemma 2.3. We can therefore assume without loss of generality that the condensation procedure uses simple functional dependencies first, i.e., the procedure of 2-condensation described in Section 4 is finished first, and in what follows, Σ is assumed to be 2-condensed.

Theorem 6.3. *The condensation of a Horn theory Σ does not depend on the order of usage of functional dependencies, and the resulting theory Σ^c is unique up to the names of representatives of logically equivalent variables of Σ.*

Proof. Let us consider the graph $G(\mathcal{M}(\Sigma))$. Every vertex of this graph which has incoming arcs corresponds to a variable which appears as the right-hand side of some non-simple minimal functional dependencies. To show that the order of usage of functional dependencies does not affect which variables are eliminated, it is sufficient to show that if a variable x can be eliminated by condensation from Σ, then x can still be eliminated by condensation from Σ' obtained by eliminating another variable y. In other words, it is sufficient to show that if a variable x had an incoming arc before a minimal functional dependency $C \to y$ (where $x \neq y$) was used for condensation, then x would still have an incoming arc in the resulting graph of the reduced theory.

Since the functional dependencies of the reduced theory are exactly those not involving y, an incoming arc of x might disappear only if that arc was produced by the minimal functional dependencies of the form $yC' \to x$. By Theorem 4.11, the graph $G(\mathcal{M}(\Sigma))$ does not have oriented cycles, and therefore $x \notin C$. Then $CC' \to x$ is a nontrivial functional dependency which holds in Σ, and there exists a set $C'' \subseteq C \cup C'$ such that $C'' \to x$ is a minimal functional dependency holding in Σ. Since $y \notin C''$, this functional dependency remains in the reduced theory, and x will have an incoming arc in the resulting graph. $\square$

The condensation of a Horn theory requires the knowledge of its functional dependencies. It was shown in Section 5 that the inference of all the minimal functional dependencies may be very expensive computationally, and even the construction of the graph $G(\mathcal{M}(\Sigma))$ may be difficult if a Horn theory is represented by a Horn CNF. However, we show now that any Horn theory can be condensed in polynomial time.

Let V^c denote the variables remaining after the condensation procedure. It follows from Theorem 6.3 that this set V^c is uniquely defined.

Theorem 6.4.

(1) *Given a theory Σ, the theory Σ^c such that $H(\Sigma^c) = H(\Sigma)^c$, and the terms representing all the variables in $V \setminus V^c$ through the variables in V^c, can be constructed in* $\mathrm{O}(|V|^3|\Sigma|)$ *time.*

(2) *Given a Horn CNF $\mathcal{F}$, a Horn CNF representing the condensation of the theory represented by $\mathcal{F}$, and the terms representing all the variables in $V \setminus V^c$ through the variables in V^c, can be constructed in* $\mathrm{O}(|V|^2|\mathcal{F}|)$ *time.*

Proof. The underlying reason for this theorem is the fact that condensation can be carried out using a very limited number of functional dependencies. More precisely, by the proof of Theorem 6.3, it is sufficient to construct a single minimal functional dependency $X \to y$ for every variable y to be eliminated. For every variable $y \in V$ such a minimal functional dependency can be easily found, if it exists at all. The procedure consists in checking whether $V \setminus y \to y$ is a functional dependency, and if yes, then deriving a minimal functional dependency $X \to y$. It follows from Theorem 3.1 that, for the CNF representation, this can be done for all the variables in $\mathrm{O}(|V|^2|\mathcal{F}|)$ time. Theorem 3.5 shows that, for the model representation, this will take $\mathrm{O}(|V|^3|\Sigma|)$ time.

The next step is to use the inferred functional dependencies to construct a subgraph G' of the graph $G(\mathcal{M}(\Sigma))$. Clearly, the variables V^c that will remain after the condensation procedure correspond to the vertices in G' (and hence in $G(\mathcal{M}(\Sigma))$) that have no incoming arcs in $G(\mathcal{M}(\Sigma))$. Then we can start from V^c, follow the arcs in G', and superpose positive terms to express every variable in $V \setminus V^c$ as a single positive term Boolean function of variables in V^c. Since the number of incoming arcs in every vertex cannot exceed $|V|$, the Boolean function in every vertex can be computed in $\mathrm{O}(|V^c||V|)$ time, and all the vertices in the graph can be processed in $\mathrm{O}(|V|^2|V^c|)$ time.

For the model representation, the only remaining step is the elimination of columns corresponding to the variables in $V \setminus V^c$. Since this can be done in linear time, and since $\mathrm{O}(|V|^2|V^c|) \leqslant \mathrm{O}(|V|^3)$, we have proven the bound of the first statement.

For the CNF representation, we now have to substitute in $\mathcal{F}$ the functional expressions for all the variables in $V \setminus V^c$. Since every clause contains at most $|V|$ literals, the substitution itself (without carrying out the expansion (5)) can be done in $\mathrm{O}(|V^c||\mathcal{F}|)$ time. Finally, the expansion (5) can be done in $\mathrm{O}(|V^c|^2|\mathcal{F}|)$ time, thus proving the bound of the second statement. $\square$

It follows from Lemma 6.1 that the bound of Theorem 6.4(2) cannot be improved.

While Theorem 6.3 states that the condensed theory is unique, the terms representing the variables in $V \setminus V^c$ may not be unique. Consider, for example, the following irredundant and prime Horn CNF:

$$\mathcal{F} = (\overline{x} \vee \overline{y} \vee \overline{z} \vee t)(\overline{t} \vee x)(\overline{t} \vee y)(\overline{t} \vee z)(\overline{u} \vee \overline{w} \vee t)(\overline{t} \vee u)(\overline{t} \vee w).$$

Since $uw \to t$ is a minimal functional dependency (see Theorem 4.1), the variable t can be eliminated by condensation, which results in the following condensed CNF:

$$\mathcal{F}^c = (\overline{x} \vee \overline{y} \vee \overline{z} \vee u)(\overline{x} \vee \overline{y} \vee \overline{z} \vee w)(\overline{u} \vee \overline{w} \vee x)(\overline{u} \vee \overline{w} \vee y)(\overline{u} \vee \overline{w} \vee z).$$

The eliminated variable t, however, can be expressed either as $t = uw$ or as $t = xyz$. The first term is shorter than the second one, and therefore is more efficient to use. It would be advantageous to find the shortest possible terms for representing the variables in $V \setminus V^c$. Therefore, we shall study the computational complexity of the following two problems.

Problem: *Shortest-Term* (*CNF*)
Instance: A Horn CNF $\mathcal{F}$, a variable $x \in V \setminus V^c$, and a number k.
Question: Can x be expressed through no more than k variables of V^c?

Problem: *Shortest-Term* (*Models*)
Instance: A theory Σ representing a Horn envelope $H(\Sigma)$, a variable $x \in V \setminus V^c$, and a number k.
Question: Can x be expressed through no more than k variables of V^c?

Clearly, both problems belong to NP, since if a term expressing y as a function of no more than k variables in V^c is given, one can easily check that all the variables in the term indeed belong to V^c and the term corresponds to a functional dependency.

We will show that both these problems are computationally difficult, using reductions from the following well known NP-complete problem (see, e.g., [16]).

Problem: *Set-Covering*
Instance: A 0–1 matrix $A = (a_{ij})_{l\times m}$ and a number k.
Question: Is there a 0–1 vector $y = (y_1, y_2, \ldots, y_m)$ such that $\sum_{j=1}^{m} y_j \leqslant k$ and the inequality

$$Ay \geqslant e \tag{6}$$

holds, where e is the l-dimensional all-one column vector?

In the following, we can assume without loss of generality that the matrix A does not have zero rows, and that no two columns of A are comparable, since the Set-Covering problem remains NP-complete under these conditions. Note that if no two columns of A are comparable, then A does not contain all-zero or all-one columns, and it also does not contain any identical columns.

Theorem 6.5. *The Shortest-Term* (*Models*) *problem is* NP-*complete.*

Proof. Let us transform an arbitrary instance of the Set-Covering problem into an instance of the Shortest-Term (Models) problem in the following way:

$$\Sigma = \left(\begin{array}{c|c} 1\ 1\ \ldots\ 1 & 1 \\ \hline J_{l\times m} - A & O_l \end{array}\right).$$

Here O_l is the $l \times 1$ zero matrix, and $J_{l\times m}$ is the $l \times m$ matrix whose elements are all 1's.

Let x be the variable corresponding to the last column of Σ. Since A does not have comparable columns, it follows from Lemma 3.3(1) that all functional dependencies in $H(\Sigma)$ have x in the right-hand side. Therefore, V^c includes all the variables of Σ except x. Lemma 3.3 (2) then implies that the left-hand side of every functional dependency of Σ corresponds to a solution of (6), and vice versa every solution of (6) corresponds to the left-hand side of a functional dependency of Σ. Therefore, $H(\Sigma)$ has a functional dependency with no more than k variables in the left-hand side if and only if the answer to the Set-Covering problem is "yes". □

Theorem 6.6. *The Shortest-Term* (*CNF*) *problem is* NP*-complete.*

Proof. We will use an instance of the Set-Covering problem to construct a Horn CNF depending on $l+m+1$ variables in the following way:

$$\mathcal{F} = \left(\bigvee_{i=1}^{l} \overline{y}_i \vee x\right) \wedge \bigwedge_{i=1}^{l} (\overline{x} \vee y_i) \wedge \bigwedge_{j=1}^{m} (\overline{x} \vee z_j) \wedge \bigwedge_{i,j:\, a_{ij}=1} (y_i \vee \overline{z}_j).$$

It can be checked that $\mathcal{F}$ is irredundant and prime, and it contains all the quadratic prime implicates of the Horn theory Σ it represents. In addition to the prime implicates in $\mathcal{F}$, Σ has prime implicates $\bigvee_{i=1}^{l} \overline{y}_i \vee z_j$, $j = 1, \ldots, m$, and implicates of the form

$$\bigvee_{i \in I} \overline{y}_i \vee \bigvee_{j \in J} \overline{z}_j \vee x,$$

where I and J satisfy the condition that for every $i \in \{1, \ldots, l\} \setminus I$ there exists a $j \in J$ such that $a_{ij} = 1$.

Since A does not have all-one columns, for every $j \in \{1, \ldots, m\}$ there exists an $i \in \{1, \ldots, l\}$ such that $y_i \vee \overline{z}_j$ is not an implicate of Σ. Then it follows from Theorem 4.2 that all functional dependencies in this Horn theory Σ have x in the right-hand side, and are of the following form:

$$\bigcup_{i \in I} y_i \cup \bigcup_{j \in J} z_j \to x, \tag{7}$$

where I and J satisfy the condition that for every $i \in \{1, \ldots, l\} \setminus I$ there exists a $j \in J$ such that $a_{ij} = 1$. Thus, the set V^c consists of $l+m$ variables: y_i, $i = 1, 2, \ldots, l$, and z_j, $j = 1, 2, \ldots, m$.

Let us consider a functional dependency of the form (7). Since, without loss of generality, A does not have zero rows, for every $i \in I$ there exists a $j(i) \in \{1, \ldots, m\}$ such that $y_i \vee \overline{z}_{j(i)}$ is a prime implicate of Σ. Therefore, if a functional dependency (7) holds in Σ, then the functional dependency

$$\bigcup_{j \in J \cup \bigcup_{i \in I} j(i)} z_j \to x$$

also holds in Σ, and the cardinality of its right-hand side is

$$\left| J \cup \bigcup_{i \in I} j(i) \right| \leqslant |J| + |I|.$$

Clearly, if $I = \emptyset$, then there is a one-to-one correspondence between the sets J of functional dependencies of the form (7) and the vectors y satisfying the inequality (6). Therefore, x can be expressed through no more than k variables of V^c if and only if the answer to the Set-Covering problem is "yes". □

7. Concluding remarks

We studied here functional dependencies in Horn theories, with the main emphasis on computational results and structural properties. We considered the representation of a Horn theory as a Horn CNF, and as the Horn envelope of a set of models. In both cases, one can recognize in polynomial time whether a given functional dependency holds in a given Horn theory, while it is computationally difficult for a general CNF.

We established a correspondence between minimal functional dependencies in a Horn theory and some prime implicates of the theory, and proved that the associated Horn theory (defined by all minimal functional dependencies) is a superset of the original Horn theory. It was also established that every functional dependency in a Horn theory has the functional form of a single positive term, while a functional dependency in a general Boolean theory can be an arbitrary Boolean function without redundant variables.

We associated a directed graph with a set of functional dependencies, and proved that such a graph associated with all minimal functional dependencies in a Horn theory is quasi-acyclic; i.e., all its cycles (if any) are created by the logically equivalent variables. It was shown that this graph can be constructed in polynomial time if a Horn theory is represented as the Horn envelope of a set of models, while this construction becomes computationally difficult if a theory is represented by a Horn CNF. We showed that the set of minimal functional dependencies (and even its minimum FD-cover) can be exponentially large as compared with the size of both the CNF and the Horn envelope representations of a theory.

We introduced the procedure of condensing a theory by eliminating those variables that are functionally dependent on other variables. In the case of general Boolean theories, the condensed theory may depend on the choice of functional dependencies to be used in the condensation process, and the functional expressions of eliminated variables through the remaining variables may be too complicated to make condensation computationally advantageous. However, the condensation of a Horn theory is unique, and can be constructed in polynomial time.

In the research report version of this paper (see [22]) we develop further results about functional dependencies in Horn theories. We show how to recognize in polynomial time whether a minimal functional dependency in a Horn theory belongs to all or only to some irredundant FD-covers of the theory, or does not belong at all to any irredundant FD-cover of the theory. We also consider the complexity of inferring all minimal functional dependencies as a function of the size of the output, develop an incrementally polynomial algorithm for inferring all minimal functional dependencies in a theory given as a Horn CNF, and show that the existence of a polynomial total time algorithm for inferring all minimal functional dependencies in a theory given as the Horn envelope of a set of models is equivalent to the existence of a polynomial total time algorithm for the well known problem of dualizing a positive theory (see, e.g., [3]).

The results presented in this paper can be extended in a straightforward way to a wider class of so-called *renamable* Horn theories. A theory Σ is called renamable Horn if the theory Σ' resulting after substituting some variables in Σ with their negations (i.e., renaming some x's as $\overline{x}$'s) is Horn. If a renaming to make a theory Horn is known, then the theory can be transformed into a Horn form, and can be worked with as essentially a Horn theory. It turns out that for some representations of a renamable Horn theory such renaming can be easily found. This is the case for the CNF representation: one can recognize in polynomial time whether a given CNF can be renamed as Horn, and if yes, a renaming making it Horn can also be determined in polynomial time (see [8,31]). The "envelope-type" representation of a renamable Horn theory is not well defined in the sense that the explicit knowledge of renaming is required to redefine the intersection closure. If all the models of a theory are given, then one can check in polynomial time if the theory is renamable Horn. To see this, it is sufficient to use a given set of models for constructing in polynomial time (as described in [44]) a prime CNF representing this theory. One can easily see that a theory is renamable Horn if and only if any of its prime CNFs can be renamed as a Horn CNF.

It is obvious that if $X \to y$ is a functional dependency in a theory Σ, then this functional dependency will hold in any theory Σ' obtained from Σ by renaming some variables, i.e., no renaming changes the set of functional dependencies. However, the functional form of the expression $y = f(X)$ will change in accordance with the renaming. In the case of renamable Horn theories this functional form does not become significantly more complicated: it is either a single term or a single clause, which is not necessarily positive any more.

Renamable Horn CNFs and 2-CNFs (where each clause contains at most 2 literals) are well known classes of formulae for which the satisfiability problem can be solved in polynomial time. These two classes turn out to be special cases of the class of so-called *q-Horn* CNFs which were introduced and studied in [4,5,7]. It was shown that a q-Horn CNF can be characterized by a special linear programming problem associated to the CNF, can therefore be recognized in polynomial time, and the q-Horn satisfiability problem can be solved in polynomial time. In a forthcoming paper [23], we study q-Horn theories and show that functional dependencies in a q-Horn theory still have the form similar to functional dependencies in a renamable Horn theory, i.e., they are either a single term or a single clause.

Acknowledgements

The authors extend their thanks for the partial support provided by the Office of Naval Research (grant N00014-92-J1375). A part of this work was done during the second author's visit at Kyoto University, which was supported by a Scientific Grant in Aid by the Ministry of Education, Culture and Sports of Japan. The authors are grateful to Thomas Eiter for his useful comments. The authors also express their gratitude to the anonymous referees whose constructive comments helped significantly improve this paper.

References

[1] W.W. Armstrong, Dependency structures of database relationships, in: Proceedings IFIP-74, North-Holland, Amsterdam, 1974, pp. 580–583.

[2] C. Beeri, M. Dowd, R. Fagin, R. Statman, On the structure of Armstrong relations for functional dependencies, J. ACM 31 (1) (1984) 30–46.

[3] J.C. Bioch, T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, Inform. and Comput. 123 (1995) 51–75.

[4] E. Boros, Y. Crama, P.L. Hammer, Polynomial time inference of all valid implications for Horn and related formulae, Ann. Math. Artificial Intelligence 1 (1990) 21–32.

[5] E. Boros, Y. Crama, P.L. Hammer, M. Saks, A complexity index for satisfiability problems, SIAM J. Comput. 23 (1994) 45–49.

[6] E. Boros, O. Čepek, A. Kogan, Horn minimization by iterative decomposition, Ann. Math. and Artificial Intelligence 23 (3–4) (1998) 321–343.

[7] E. Boros, P.L. Hammer, X. Sun, Recognition of q-Horn formulae in linear time, Discrete Appl. Math. 55 (1994) 1–13.

[8] V. Chandru, C.R. Coullard, P.L. Hammer, M. Montañez, X. Sun, On renamable Horn and generalized Horn functions, Ann. Math. Artificial Intelligence 1 (1990) 33–47.

[9] C.L. Chang, R.C.T. Lee, Symbolic Logic and Mechanical Theorem Proving, Academic Press, New York, 1973.

[10] E.F. Codd, A relational model for large shared data banks, Comm. ACM 13 (1970) 377–387.

[11] R. Dechter, J. Pearl, Structure identification in relational data, Artificial Intelligence 58 (1992) 237–270.

[12] C. Delobel, R.G. Casey, Decomposition of a data base and the theory of Boolean switching functions, IBM J. Res. Development 17 (1973) 374–386.

[13] W.F. Dowling, J.H. Gallier, Linear time algorithms for testing the satisfiability of propositional Horn formulae, J. Logic Programming 3 (1984) 267–284.

[14] R. Fagin, Functional dependencies in a relational database and propositional logic, IBM J. Res. Development 21 (1977) 534–544.

[15] R. Fagin, Horn clauses and database dependencies, J. ACM 29 (4) (1982) 952–985.

[16] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.

[17] P.L. Hammer, A. Kogan, Horn functions and their DNFs, Inform. Process. Lett. 44 (1992) 23–29.

[18] P.L. Hammer, A. Kogan, Optimal compression of propositional Horn knowledge bases: Complexity and approximation, Artificial Intelligence 64 (1993) 131–145.

[19] P.L. Hammer, A. Kogan, Quasi-acyclic propositional Horn knowledge bases: Optimal compression, IEEE Trans. Knowledge and Data Engineering 7 (1995) 751–762.

[20] P.L. Hammer, A. Kogan, Essential and redundant rules in Horn knowledge bases, Decision Support Systems 16 (2) (1996) 119–130.

[21] A. Horn, On sentences which are true of direct unions of algebras, J. Symbolic Logic 16 (1951) 14–21.

[22] T. Ibaraki, A. Kogan, K. Makino, Functional dependencies in Horn theories, RUTCOR Research Report RRR 14–98, Rutgers University, New Brunswick, NJ, 1998.

[23] T. Ibaraki, A. Kogan, K. Makino, Functional dependencies in q-Horn theories, Manuscript, 1998.

[24] H. Kautz, M. Kearns, B. Selman, Horn approximations of empirical data, Artificial Intelligence 74 (1995) 129–145.

[25] H. Kautz, M. Kearns, B. Selman, Forming concepts for fast inference, in: Proc. AAAI-92, San Jose, CA, 1992, pp. 786–793.

[26] D. Kavvadias, C.H. Papadimitriou, M. Sideri, On Horn envelopes and hypergraph transversals, in: K.W. Ng et al. (Eds.), Algorithms and Computation—ISAAC-93, Lecture Notes in Computer Science, Vol. 762, Springer, Berlin, 1993, pp. 399–405.

[27] R. Khardon, Translating between Horn representations and their characteristic models, J. Artificial Intelligence Res. 3 (1995) 349–372.

[28] R. Khardon, H. Mannila, D. Roth, Reasoning with examples: Propositional formulae and database dependencies, Acta Informatica (1998) to appear. (Preliminary version: Technical Report TR-15-95, Aiken Computation Laboratory, Harvard University, Cambridge, MA, 1995.)

[29] R. Khardon, D. Roth, Reasoning with models, Artificial Intelligence 87 (1996) 187–213.
[30] J. Kivinen, H. Mannila, Approximate inference of functional dependencies from relations, Theoret. Comput. Sci. 149 (1995) 129–149.
[31] H.R. Lewis, Renaming a set of clauses as a Horn set, J. ACM 25 (1978) 134–135.
[32] C.L. Lucchesi, S.L. Osborn, Candidate keys for relations, J. Comput. System Sci. 17 (1978) 270–279.
[33] D. Maier, The Theory of Relational Databases, Computer Science Press, Rockville, MD, 1983.
[34] H. Mannila, K.-J. Räihä, Design of Relational Databases, Addison-Wesley, Wokingham, 1992.
[35] H. Mannila, K.-J. Räihä, Algorithms for inferring functional dependencies, Data and Knowledge Engineering 12 (1994) 83–99.
[36] J.C.C. McKinsey, The decision problem for some classes of sentences without quantifiers, J. Symbolic Logic 8 (3) (1943) 61–77.
[37] M. Minoux, LTUR: A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation, Inform. Process. Lett. 29 (1988) 1–12.
[38] M. Minoux, The unique Horn-Satisfiability problem and quadratic Boolean equations, Ann. Math. Artificial Intelligence 6 (1992) 253–266.
[39] Y. Sagiv, C. Delobel, D.S. Parker, R. Fagin, An equivalence between relational database dependencies and a fragment of propositional logic, J. ACM 28 (3) (1981) 435–453.
[40] B. Selman, H. Kautz, Knowledge compilation and theory approximation, J. ACM 43 (2) (1996) 193–224.
[41] B. Selman, H.J. Levesque, Support set selection for abductive and default reasoning, Artificial Intelligence 82 (1996) 259–272.
[42] J.D. Ullman, Principles of Database and Knowledge-Base Systems, Vol. I: Classical Database Systems, Computer Science Press, New York, 1988.
[43] J.D. Ullman, Principles of Database and Knowledge-Base Systems, Vol. II: The New Technologies, Computer Science Press, New York, 1989.
[44] Yu.I. Zhuravlev, A. Kogan, Realization of Boolean functions with a small number of zeros by disjunctive normal forms and related problems, Soviet Mathematics—Doklady 32 (3) (1985) 771–775.