DIMACS Technical Report 98-18
March 1998

Functional Dependencies in Horn Theories

by

Toshihide Ibaraki[1] Alexander Kogan[2] Kazuhisa Makino[3]

[1]Department of Applied Mathematics and Physics, Graduate School of Engineering, Kyoto University, Kyoto, Japan 606, e-mail: `ibaraki@kuamp.kyoto-u.ac.jp`
[2]Department of Accounting and Information Systems, Faculty of Management, Rutgers University, Newark, NJ 07102, and RUTCOR, Rutgers University, 640 Bartholomew Road, Piscataway, NJ 08854-8003, U.S.A., e-mail: `kogan@rutcor.rutgers.edu`
[3]Department of Systems and Human Science, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, Japan, e-mail: `makino@sys.es.osaka-u.ac.jp`

DIMACS is a partnership of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, Bellcore and NEC Research Institute.

DIMACS is an NSF Science and Technology Center, funded under contract STC–91–19999; and also receives support from the New Jersey Commission on Science and Technology.

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 that hold in a given Horn theory. We show that some functional dependencies inference problems 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 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

Contents

1 Introduction

Relational databases have been invented, studied and deployed as essential tools of information storage and retrieval (see [9, 34, 43, 35]). Functional dependencies have 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 [36, 30]. Thorough theoretical studies of functional dependencies in relational databases (see [11, 14, 42, 15, 28]) have established a close connection between them and Horn clauses.

Horn clauses were introduced in formal logic (see [37, 22]), and gained prominence in logic programming (see [12]) and artificial intelligence (see [8, 10, 25]). 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 [12, 38]) 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 [25, 29, 27]). 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.

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, may have much fewer variables than the original theory, and may 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.

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. 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 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 that hold in a given Horn theory. We show that some functional dependencies inference problems 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, and the condensation of a Horn theory is computationally advantageous. 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.

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 (or points) 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 $\alpha \leq \beta$ the condition that $\alpha_i \leq \beta_i$ for all $i = 1, 2, \ldots, n$.

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 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 (Boolean function) 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 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 implicates, then every quadratic implicate of such a theory is prime.

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 with a single variable in their right-hand sides. A functional dependency $X \to y$ in Σ states that in all the models of Σ the variable y is a Boolean function of the variables in X (i.e., $y = f(X)$).

If a theory Σ is given as a set of models, then one can check in $O(|V||\Sigma|)$ time[1] whether a functional dependency $X \to y$ holds in Σ, simply by building a binary decision tree on the points in Σ using all the variables in X one by one at the decision nodes, and then checking if there exists a leaf in the resulting tree which contains points in Σ that have the opposite values in y. On the other hand, if a theory is represented by a CNF, the problem becomes difficult.

Theorem 2.1 *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. Given a 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, except for the case when $\mathcal{F}'$ has no models. 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 the monotonicity, it is essential to know only the set of minimal functional dependencies in Σ, which will be denoted by $\mathcal{M}(\Sigma)$.

---

[1] In the following we use the notation $\phi = O(\psi)$ to denote that there exists a constant C such that $\phi \leq C\psi$.

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 2.2 *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 proof of Theorem 2.1 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. For example, if $X \to Y$ and $Y \to Z$ hold in Σ, then $X \to Z$ must also hold in Σ. 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, 35]). It is known that for any set of functional dependencies 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, it follows from Lemma 2.2 that $\mathcal{D} = \{x \to y\}$ has no Armstrong relation among Boolean theories (since the dependency $y \to x$ would also hold in any such theory).

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 this value can be determined from the values of the variables in X, since $y = f(X)$, where f is a Boolean function. It therefore may be beneficial to "reduce" Σ by eliminating y and considering instead the theory $\Sigma[V \setminus y]$. If the description of 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.

The reduction procedure described in the previous paragraph is similar to the one which is routinely used in the logical design of relational databases in the process called *normalization*. 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.

If 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 will be called a *condensation* of Σ, and 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 randomly. Therefore, the result of the condensation procedure is generally non-deterministic, and a theory may be condensed to many different ones.

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 [18, 19]) 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 [12]), 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 [37, 10]) 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. The discussion in the previous paragraphs indicates two common ways of representing Horn theories: as a CNF and as the set of all models. The closedness of Horn theories under intersection suggests that 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. 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 $H(\Sigma) \subseteq \Sigma'$.

It is natural to establish a correspondence between functional dependencies and Horn clauses by introducing for a functional dependency $X \rightarrow y$ the definite Horn clause $y \vee \bigvee_{x \in X} \overline{x}$. This correspondence has been well studied (see [11, 14, 15, 42]), 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 $X \rightarrow y$ 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 [18, 19]), 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 a set of row vectors in the following matrix Γ:

$$\Gamma = \left(\begin{array}{cccccccc} x & y & z & w & u & p & q & t \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \end{array} \right).$$

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

$$(\overline{x} \vee u)(p \vee q)(\overline{t} \vee q)(\overline{z} \vee \overline{w} \vee q)(z \vee w)(\overline{q} \vee z)(\overline{q} \vee w)(\overline{t} \vee p)(x \vee y)$$

$$\wedge(\overline{p} \vee x)(\overline{p} \vee y)(\overline{u} \vee \overline{y} \vee \overline{q} \vee t).$$

Furthermore, it can be seen that the following set of functional dependencies is an FD-cover of Σ:

$$\mathcal{D} = \{xy \rightarrow p, zw \rightarrow q, pq \rightarrow t, x \rightarrow u, u \rightarrow x, pt \rightarrow q, qt \rightarrow p,$$

$$wq \rightarrow z, zq \rightarrow w, xp \rightarrow y, yp \rightarrow x\}.$$

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

$$\mathcal{M}(\Sigma) = \mathcal{D} \bigcup \{wxyt \rightarrow z, xyq \rightarrow t, uyq \rightarrow t, pzw \rightarrow t, xyzw \rightarrow t, uyzw \rightarrow t, xyt \rightarrow q,$$

$$uy \rightarrow p, uyt \rightarrow q, zwt \rightarrow p, wpt \rightarrow z, wuyt \rightarrow z, zpt \rightarrow w, zxyt \rightarrow w, zuyt \rightarrow w, xqt \rightarrow y,$$

$$uqt \rightarrow y, xzwt \rightarrow y, uzwt \rightarrow y, up \rightarrow y, yp \rightarrow u, yqt \rightarrow x, yqt \rightarrow u, yzwt \rightarrow x, yzwt \rightarrow u\}.$$

The condensation of Σ using the sequence of functional dependencies $\{pq \to t, zw \to q, xy \to p, x \to u\}$ results in the condensed theory Σ_1^c, while the condensation of Σ using the sequence of functional dependencies $\{x \to u, qt \to p, zq \to w, yp \to x\}$ results in the condensed theory Σ_2^c:

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

In the condensation of Σ_1^c the eliminated variables have the following expressions: $u = x$, $p = xy$, $q = zw$, $t = pq = xyzw$, while in the condensation of Σ_1^c the eliminated variables have the following expressions: $p = \overline{q} \vee t$, $w = \overline{z} \vee q$, $u = x = \overline{y} \vee p = \overline{y} \vee \overline{q} \vee t$.

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

$$H(\Sigma) = \begin{array}{c} \begin{array}{cccccccc} x & y & z & w & u & p & q & t \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \end{array}, \qquad H(\Sigma)^c = \begin{array}{c} \begin{array}{cccc} x & y & z & w \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{array}$$

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

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

$$(\overline{x} \vee \overline{y} \vee p)(\overline{z} \vee \overline{w} \vee q)(\overline{p} \vee \overline{q} \vee t)(\overline{x} \vee u)(\overline{u} \vee x)(\overline{p} \vee x)(\overline{p} \vee y)(\overline{q} \vee z)(\overline{q} \vee w)(\overline{t} \vee p)(\overline{t} \vee q).$$

The following set of functional dependencies is now an FD-cover of $H(\Sigma)$,

$$\mathcal{D} = \{xy \to p, zw \to q, pq \to t, x \to u, u \to x\},$$

and the set of all minimal functional dependencies that hold in $H(\Sigma)$ is

$$\mathcal{M}(H(\Sigma)) = \mathcal{D} \bigcup \{uy \to p, xyq \to t, uyq \to t, pzw \to t, xyzw \to t, uyzw \to t\}.$$

The theory $H(\Sigma)$ has a unique condensation which is shown in Figure 1. The condensation $H(\Sigma)^c$ happens to be the trivial truth theory: every Boolean vector is its model. The eliminated variables have the following expressions: $u = x$, $p = xy$, $q = zw$, $t = pq = xyzw$.

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 Subsection 2.2 (see Theorem 2.1) 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 linear 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 $z \in V \setminus (X \cup y)$. Then 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 linear time by substituting $y = 1$ in (1) and solving the resulting Horn satisfiability problem. 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 [12]) can be employed to construct a linear time algorithm for checking whether a functional dependency holds in a theory represented by a Horn CNF. $\square$

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 $O(|X||\mathcal{F}|)$ 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. $\square$

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.,

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

Similarly,

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

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 in Σ_1^y have the value* 1 *in all the coordinates in X', 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 $\alpha_1[X] \leq \beta_0[X]$.

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 $\alpha_1 \leq \alpha$. 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, $\beta \leq \beta_0$. Then

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

Conversely, if $\alpha_1[X] \leq \beta_0[X]$, 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 $\alpha_1[X] \leq \beta_0[X]$ if and only if for every $\beta \in \Sigma_0^y$ there exists a coordinate $x \in X$ such that $\beta[x] = 0$ and $\alpha_1[x] = 1$. By construction, if $\alpha_1[x] = 1$ then for every point $\alpha \in \Sigma_1^y$ we have $\alpha[x] = 1$. Finally, let $X' = \{x \in X | \alpha_1[x] = 1\}$. Then it is easy to see that $\alpha_1[X] \leq \beta_0[X]$ holds if and only if conditions 1 and 2 of the lemma hold. $\square$

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 in Σ_1^y have the value* 1 *in all the coordinates in X,*
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 $\alpha[X \backslash x] = (11 \cdots 1)$.*

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. □

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 Subsection 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. Since the Horn envelope $H(\Sigma)$ may contain the 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 linear 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 $\alpha[X'] = (11\cdots 1)$ 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 $\beta[X'] = (11\cdots 1)$. □

Corollary 3.6 *Given a theory Σ and a functional dependency $X \to y$, it can be checked in linear 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 $\beta[X \setminus x] = (11\cdots 1)$. □

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 $X \to y$ is minimal in a Horn theory Σ if and only if, for all $x \in X$, clauses $\overline{y} \vee x$ and $y \vee \bigvee_{x \in X} \overline{x}$ 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 $\alpha[X] = (11 \cdots 1)$;
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 $\alpha[X \setminus x] = (11 \cdots 1)$.

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. □

Note that Theorem 4.1 provides a slightly different (from Theorem 3.1 and Corollary 3.2) way of checking whether $X \to y$ is a minimal functional dependency in the theory represented by a Horn CNF $\mathcal{F}$. It consists in simply checking (e.g. as described in [18, 19]) 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 4.2 *For a Horn theory Σ, the set of models of $\mathcal{M}(\Sigma)$ is a superset of Σ.*

Corollary 4.3 *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[2] 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.

[2] 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.

Corollary 4.4 *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 $\alpha \in \Sigma$.*

To analyze the structure of functional dependencies in Horn theories, we shall associate to every set of of functional dependencies $\mathcal{D}$ a directed graph $G(\mathcal{D})$ [34] 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 [39] in the study of unique Horn satisfiability, and in [20] 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.

Theorem 4.5 *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. Since every arc $x' \to y'$ in $G(\mathcal{M}(\Sigma))$ comes from a minimal functional dependency having x' in the left-hand side and y' in the right-hand side, 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 $y \to x$. □

Theorem 4.5 and Lemma 2.2 imply the following corollary.

Corollary 4.6 *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 4.5 and Corollary 4.4 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 [20] 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.

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 [20]). 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 all the rows of Σ' one by one at the decision nodes, and then removing from Σ' all but one columns 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 Subsection 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 4.4 implies the following statement.

Corollary 4.7 *No simple functional dependency holds in a 2-condensed Horn theory.*

For a variable $x \in V$, let us denote by $r(x)$ the representative of x in Σ^{2c}. Similarly, let $r(X)$ denote the set of representatives of X . The following statement was proven in [20].

Proposition 4.8 *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 $X \rightarrow y$ is minimal in a Horn theory Σ if and only if either $r(X) \rightarrow r(y)$ 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.

Theorem 4.5 and Corollary 4.7 imply the following important structural property of functional dependencies in 2-condensed Horn theories.

Theorem 4.10 *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 [20], where it was proven that any acyclic Horn theory has a unique irredundant and prime CNF, thus establishing the next theorem.

Theorem 4.11 *The set of functional dependencies holding in a 2-condensed Horn theory has a unique irredundant FD-cover.*

We will call a set of functional dependencies *acyclic* if the Horn theory represented by the corresponding definite Horn CNF is acyclic. Theorem 4.10 and Corollary 4.7 imply that the set of functional dependencies holding in a 2-condensed Horn theory is acyclic, and does not contain any simple dependency. The next result provides a more subtle structural property of the set of functional dependencies holding in a Horn theory.

Theorem 4.12 *Let Σ be a Horn theory and $X \rightarrow y$ be a functional dependency in Σ. If $X \rightarrow y$ is minimal, then the graph $G(\mathcal{M}(\Sigma))$ contains no oriented path from any node $x' \in X$ to any node $x'' \in X \setminus x'$.*

Proof. We will prove this theorem by contradiction. It was argued in the proof of Theorem 4.5 that if $G(\mathcal{M}(\Sigma))$ contains an oriented path from x' to x'', then $\overline{x''} \vee x'$ is a prime implicate of Σ. Then the clause $y \vee \bigvee_{x \in X} \overline{x}$ cannot be a prime implicate of Σ, since this clause together with the clause $\overline{x''} \vee x'$ imply that the clause $y \vee \bigvee_{x \in X \setminus x'} \overline{x}$ is an implicate of Σ. Therefore, by Theorem 4.1, the functional dependency $X \rightarrow y$ is not minimal. □

The converse statement does not hold, i.e., there exists an acyclic set of functional dependencies $\mathcal{D}$ (which does not contain simple dependencies), and for every dependency $X \rightarrow y$ and every $x', x'' \in X$, the graph $G(\mathcal{D})$ does not contain an oriented path from x' to x'', and nevertheless, no Horn theory is $\mathcal{D}$'s Armstrong relation. Consider for example the following set of functional dependencies:

$$\mathcal{D} = \{xy \rightarrow z, xy \rightarrow w\}.$$

Clearly, $\mathcal{D}$ satisfies all the conditions mentioned above, and $\mathcal{D} = \widehat{\mathcal{D}}$. It follows from Theorem 4.1 that any Horn theory which is $\mathcal{D}$'s Armstrong relation must be a subset of the Horn theory represented by the following CNF:

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

Since both $\overline{z} \vee w$ and $\overline{w} \vee z$ are prime implicates of $\mathcal{F}$, Theorem 4.1 tells that the functional dependencies $z \rightarrow w$ and $w \rightarrow z$ will be minimal functional dependencies in every Horn theory which is a subset of $\mathcal{F}$. Since these dependencies are not in $\widehat{\mathcal{D}}$, no Horn theory is $\mathcal{D}$'s Armstrong relation.

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 (following the terminology of [20]). In this general case, the irredundant FD-cover will not be unique any more, but the results of [20] 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 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. Otherwise, we can 2-condense it, infer the functional dependencies holding in the 2-condensation, and then describe all the functional dependencies in the original theory.

5.1 Horn Theories with Large FD-Cover

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 *There exists a 2-condensed Horn theory which has the CNF representation of size $O(n)$ and the irredundant FD-cover of size*[3] $\Omega(2^n)$.

Proof. Consider the following Horn CNF:

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

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 $\bigwedge_{i=1}^{n} z_i \rightarrow x_0$, 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. □

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

Theorem 5.2 *There exists a theory Σ having size $|\Sigma| = O(n)$ and depending on $O(n)$ 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 a single model $(1111\cdots11)$. Among the remaining $3n$ models, there is a group of n models such that in the i-th 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 i-th model of this group the only variable that has the value 1 is x_i. Informally, Σ can be given by

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

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

Since all the columns of Γ 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 $i \leq 2n$ 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 $i \leq n$ 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 $\bigwedge_{i=1}^{n} y_i \rightarrow x_{2n+1}$, 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. $\square$

5.2 Recognizing Irredundant FD-Cover Dependencies

Theorem 4.11 implies that a minimal functional dependency in a 2-condensed Horn theory either belongs to the unique irredundant FD-cover, and therefore is not implied by all the other functional dependencies, or does not belong to the irredundant FD-cover, and therefore is implied by the FD-cover. The former dependencies are clearly more important than the latter. One could immediately tell the former from the latter if the irredundant FD-cover were available. However, the results of the previous subsection state that the irredundant FD-cover of a 2-condensed Horn theory may include exponentially many (in the size of Horn CNF or Horn envelope representation) functional dependencies, and therefore the construction of the FD-cover is computationally intractable.

We will develop below polynomial procedures for checking whether a minimal functional dependency in a 2-condensed Horn theory belongs to its unique irredundant FD-cover. These procedures are based on a polynomial algorithm developed in [21] for checking whether a definite Horn prime implicate of a Horn CNF is *essential*, i.e., whether it belongs to every irredundant prime CNF representing the same theory. Using the correspondence between functional dependencies and Horn clauses described in Subsection 2.4, we can paraphrase a result obtained in [21] in the following way.

Let $I_\Sigma(X)$ denote the set of all variables $y \in V$ such that $X \rightarrow y$ is a functional dependency in a theory Σ. Obviously, $X \subseteq I_\Sigma(X)$ for any Σ.

Proposition 5.3 *A minimal functional dependency $X \rightarrow y$ in a theory Σ belongs to every irredundant FD-cover of Σ if and only if for every $z \in X$ the functional dependency $I_\Sigma(X) \setminus \{y, z\} \rightarrow y$ does not hold in Σ.*

It follows from Proposition 5.3 that to check whether $X \rightarrow y$ belongs to every irredundant FD-cover of Σ, one can carry out the following two steps.

1. Construct $I_\Sigma(X)$ by checking for every $z \in V \setminus X$ whether $X \rightarrow z$ is a functional dependency in Σ.

2. For every $z \in X$, check whether $I_\Sigma(X) \setminus \{y, z\} \rightarrow y$ is a functional dependency in Σ.

Theorem 5.4 *Given a theory Σ whose Horn envelope $H(\Sigma)$ is 2-condensed, and a minimal functional dependency $X \to y$ in $H(\Sigma)$, it can be checked in $O(|V|^2|\Sigma|)$ time whether $X \to y$ belongs to the unique irredundant FD-cover of $H(\Sigma)$.*

Proof. It follows from Theorem 3.5 that the set $I_{H(\Sigma)}(X)$ can be constructed in $O(|V|^2|\Sigma|)$ time, and then it can be checked in $O(|X||V||\Sigma|)$ time whether there exists $z \in X$ such that $I_{H(\Sigma)}(X) \setminus \{y, z\} \to y$ is a functional dependency in $H(\Sigma)$. □

Theorem 5.5 *Given a Horn CNF $\mathcal{F}$ representing a 2-condensed Horn theory Σ, and a minimal functional dependency $X \to y$ in Σ, it can be checked in $O(|V||\mathcal{F}|)$ time whether $X \to y$ belongs to the unique irredundant FD-cover of Σ.*

Proof. It follows from Theorem 3.1 that the set $I_\Sigma(X)$ can be constructed in $O(|V||\mathcal{F}|)$ time, and then it can be checked in $O(|X||\mathcal{F}|)$ time whether there exists $z \in X$ such that $I_\Sigma(X) \setminus \{y, z\} \to y$ is a functional dependency in Σ. □

As was discussed at the end of Section 4, if a Horn theory Σ is not 2-condensed, then it can have many irredundant FD-covers, but they all have essentially the same structure determined in large part by the unique irredundant FD-cover of Σ^{2c}. Therefore, Theorems 5.4 and 5.5 can be applied to Σ^{2c} to characterize the relative importance of minimal functional dependencies in Σ. This allows to recognize in polynomial time whether a minimal functional dependency in Σ belongs to all irredundant FD-covers of Σ, or only to some irredundant FD-covers of Σ, or does not belong at all to any irredundant FD-covers of Σ.

5.3 Aggregate Description of 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:

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

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 Horn theories are represented by sets of models and by CNFs. We first consider the Horn envelope representation.

Theorem 5.6 *Given a theory Σ, the set $F_{H(\Sigma)}(y)$ can be constructed in $O(|V||\Sigma|^2)$ time for every variable $y \in V$.*

Proof. 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), $F_{H(\Sigma)}(y) \subseteq X$. 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.

We say that a Boolean vector α *dominates* a Boolean vector β if $\alpha[x] = 0$ implies $\beta[x] = 0$ for all x. 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 $x \in X'$ such that there exists an $\alpha \in \Sigma_0^y$ such that $\alpha[x] = 0$ and $\alpha[X' \setminus x] = (11 \cdots 1)$, the set X'' must contain this x. Therefore, 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 for every $\beta \in \Sigma_0^y \setminus \alpha$ there exists a variable $z \in X_y$ such that $\beta[z] = 0$ and $\alpha[z] = 1$. This means that $\alpha[X_y]$ is not dominated by any other point in $\Sigma_0^y[X_y]$. Therefore, 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 which is dominated by another point 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 $O(|V||\Sigma|^2)$ time. □

Theorem 5.6 shows that the sets $F_\Sigma(y)$, and therefore the graph $G(\mathcal{M}(\Sigma))$, are easily constructible if a Horn envelope representation of a Horn theory is available. 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 5.7 *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 [33]) 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

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

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. Properties of the forward chaining procedure (see [19, 20]) then imply that the 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. □

5.4 Inferring All Minimal Functional Dependencies

In the developments that follow we investigate the computational complexity of the following problem.

Problem: FD-Inference

Input: A Horn theory represented either by a Horn CNF or as the Horn envelope of a set of models.

Output: All minimal functional dependencies of the Horn theory.

Theorems 5.1 and 5.2 show that the set of all minimal functional dependencies may be exponential in the input size. This is the reason why the complexity of the FD-Inference problem should be evaluated in terms of both the input and the output lengths.

An algorithm enumerating items $a_1, a_2, \ldots, a_p$ is called *incrementally polynomial* [24, 31] if

1. it iteratively outputs the i-th item a_i (for $i = 1, 2, \ldots, p$) using the input and the items $a_1, a_2, \ldots, a_{i-1}$ generated before, and
2. the time required for the i-th iteration is polynomial in the input length and the sizes of $a_1, a_2, \ldots, a_{i-1}$.

If an algorithm is incrementally polynomial, it also satisfies the criterion of *polynomial total time* [24] (i.e., polynomial time in the lengths of both input and output). These concepts take into account the fact that the output can be much longer (in particular, exponentially longer) than the input.

We shall see below that the complexity of the FD-Inference problem may be different depending upon whether a Horn theory is represented by a Horn CNF or as the Horn envelope of a set of models. After proving several necessary lemmas, we first consider the case in which a Horn theory is represented by a Horn CNF, and develop an incrementally polynomial algorithm for the problem. As discussed in Section 2, we continue to assume that our Horn theory has no unit implicates.

For a Horn theory Σ depending on the set of variables $V = \{x_1, x_2, \ldots, x_n\}$ and for a variable $x_i \in V$, let us introduce the following subset of variables:

$$V_i = \{x_j \in V \mid \Sigma \models (\overline{x}_i \vee x_j)\}. \tag{5}$$

Clearly, $x_i \notin V_i$, since Σ is not a tautology (it has no unit implicates).

Additionally, let us introduce a theory Σ_i depending only on the variables in V_i in the following way:

$$\Sigma_i = \{\alpha \in \Sigma \mid \alpha[x_l] = 0 \text{ for all } x_l \in V \setminus V_i\}[V_i]. \tag{6}$$

This theory Σ_i is Horn, since the intersection of any two of its models is again a model, as obvious from the assumption that Σ is Horn.

Lemma 5.8 *For a Horn theory Σ and its variable $x_i \in V$, a negative clause $C = \bigvee_{x_j \in S} \overline{x}_j$ is a prime implicate of Σ_i if and only if $S \rightarrow x_i$ is a minimal functional dependency in Σ.*

Proof. Let us first prove the "only if" part. Let a negative clause $C = \bigvee_{x_j \in S} \overline{x}_j$ be a prime implicate of Σ_i. It follows from the definition of Σ_i that the clause $C' = (C \vee \bigvee_{x_j \in V \setminus V_i} x_j)$ is an implicate of Σ. Since Σ is Horn, there is a Horn prime implicate C^* subsuming C'. Then C^* has the following properties:

(i) C^* is not negative (i.e., it is definite Horn). To prove this, assume to the contrary that $C^* = \bigvee_{x_j \in S^*} \overline{x}_j$, where $S^* \subseteq S$. Since C is an implicate of Σ_i and therefore $S \subseteq V_i$, the inclusion $S^* \subseteq V_i$ holds. By the definition of V_i, the clause $\overline{x}_i \vee x_j$ is an implicate of Σ for all $x_j \in S^*$. Therefore, we have

$$\Sigma \;\models\; C^* \bigwedge_{x_j \in S^*} (\overline{x}_i \vee x_j) \;\models\; \overline{x}_i, \tag{7}$$

which contradicts our assumption that Σ has no unit implicates.

(ii) C^* can be represented as $\bigvee_{x_j \in S^*} \overline{x}_j \vee x_i$, where $S^* \subseteq S$. Assume to the contrary that $C^* = \bigvee_{x_j \in S^*} \overline{x}_j \vee x_l$ for some $l \in V \setminus (V_i \cup \{x_i\})$. Then, similarly to (i), the clause $\overline{x}_i \vee x_j$ is an implicate of Σ for all $x_j \in S^*$, and furthermore we have

$$\Sigma \;\models\; C^* \bigwedge_{x_j \in S^*} (\overline{x}_i \vee x_j) \;\models\; \overline{x}_i \vee x_l, \tag{8}$$

which contradicts the definition of V_i.

(iii) C^* is in fact $\bigvee_{x_j \in S} \overline{x}_j \vee x_i$ (i.e., $S^* = S$), since otherwise C would not be a prime implicate of Σ_i.

It follows from (i) and (iii) that $C^* = \bigvee_{x_j \in S} \overline{x}_j \vee x_i$ is a prime implicate of Σ. Furthermore, since $S \subseteq V_i$, all the quadratic clauses $\overline{x}_i \vee x_j$, for all $x_j \in S$, are prime implicates of Σ (there no unit implicates). Therefore, by Theorem 4.1, the "only if" part follows.

Now to prove the "if" part, let us assume that $S \rightarrow x_i$ is a minimal functional dependency in Σ. Then, by Theorem 4.1, the clause $\bigvee_{x_j \in S} \overline{x}_j \vee x_i$ and all the quadratic clauses $\overline{x}_j \vee x_i$, for all $x_j \in S$, are prime implicates of Σ. Hence $V_i \supseteq S$, and $\bigvee_{x_j \in S} \overline{x}_j$ is a negative implicate of Σ_i. Moreover, this implicate $\bigvee_{x_j \in S} \overline{x}_j$ must be prime, since otherwise $S \rightarrow x_i$ would not be a minimal functional dependency of Σ (by the "only if" part). □

If a Horn theory Σ is represented by a Horn CNF $\mathcal{F}$, then a Horn CNF of Σ_i can be obtained from $\mathcal{F}$ by fixing all the variables in $V \setminus V_i$ to 0: $\mathcal{F}_{(x_l \leftarrow 0 \,|\, l \in V \setminus V_i)}$. This fact together with Lemma 5.8 shows that the FD-Inference problem for the Horn CNF representation is reduced to the generation of all negative prime implicates of a Horn CNF. We will describe below an incrementally polynomial algorithm for this problem, which is a modification of the input resolution method discussed in [4, 8].

Given two clauses $C_1 = \bigvee_{i \in P_1} x_i \vee \bigvee_{j \in N_1} \overline{x}_j$ and $C_2 = \bigvee_{i \in P_2} x_i \vee \bigvee_{j \in N_2} \overline{x}_j$ such that $P_1 \cap N_2 = \{l\}$ and $N_1 \cap P_2 = \emptyset$, the clause C_3:

$$C_3 = \bigvee_{j \in (P_1 \setminus \{l\}) \cup P_2} x_j \quad \bigvee_{j \in N_1 \cup (N_2 \setminus \{l\})} \overline{x}_j, \tag{9}$$

is called the *resolvent* of C_1 and C_2. For example, $x_2 \vee \overline{x}_3 \vee x_4 \vee \overline{x}_5 \vee x_6$ is the resolvent of $x_1 \vee \overline{x}_3 \vee x_4 \vee \overline{x}_5$ and $\overline{x}_1 \vee x_2 \vee \overline{x}_3 \vee x_4 \vee x_6$. Let $\mathcal{F} = \bigwedge_i C_i$ be an arbitrary CNF of a theory Σ. It is known (e.g., [41]) that every prime implicate C of Σ can be derived from the clauses in $\mathcal{F}$ by applying a *resolution procedure* which iteratively generates resolvents. In other words, there is a sequence $C^{(1)}, C^{(2)}, \ldots, C^{(m)} = C$ of clauses such that each $C^{(k)}$ is either in $\mathcal{F}$ or the resolvent of two clauses $C^{(k_1)}$ and $C^{(k_2)}$ such that $k_1, k_2 < k$.

For a Horn CNF $\mathcal{F}$, let Horn$(\mathcal{F})$ and Neg$(\mathcal{F})$ denote the sets of all definite Horn and all negative clauses in $\mathcal{F}$, respectively.

Algorithm: ALL-NPIS

Input: A Horn CNF $\mathcal{F}$ representing a theory $\Sigma \subseteq \{0,1\}^n$.

Output: All negative prime implicates of Σ.

Step 0. $Q := \emptyset$.

Step 1. For each clause $C \in$ Neg$(\mathcal{F})$, find a prime implicate C^* of $\mathcal{F}$ such that $C^* \models C$. If $C^* \notin Q$, then output C^* and let $Q := Q \cup \{C^*\}$. All clauses in Q are unmarked.

Step 2. If every C in Q is marked, then stop. Otherwise, take an arbitrary unmarked clause C in Q, mark it, and go to Step 3.

Step 3. For each $C' \in$ Horn$(\mathcal{F})$, if C and C' have a resolvent C'', then find a prime implicate C^* of Σ such that $C^* \models C''$. If $C^* \notin Q$, then output C^* and let $Q := Q \cup \{C^*\}$, where C^* is unmarked. Go to Step 2.

Let us first show the correctness of this algorithm. Let NPI(Σ) denote the set of all negative prime implicates of Σ, and let ALL-NPIS output the set of clauses Q. We show that $Q =$ NPI(Σ) holds. Step 1 of the algorithm outputs only negative prime implicates of Σ, and since the resolvent of a negative clause and a definite Horn clause is negative, Step 3 also outputs only negative prime implicates of Σ. Hence, $Q \subseteq$ NPI(Σ) holds. Assume

$C_1 \in \mathrm{NPI}(\Sigma) \setminus Q$. Let Σ_Q and $\Sigma_{Q\cup\{C_1\}}$ denote the theories represented by $\mathcal{F}_Q = \bigwedge_{C\in Q} C$ and $\mathcal{F}_{Q\cup\{C_1\}} = \bigwedge_{C\in Q\cup\{C_1\}} C$, respectively. Then the following implications hold:

$$\Sigma \;\models\; \Sigma_{Q\cup\{C_1\}} \;\models\; \Sigma_Q. \tag{10}$$

Since Σ_Q and $\Sigma_{Q\cup\{C_1\}}$ are negative, Q and $Q\cup\{C_1\}$ are exactly the sets of all prime implicates of Σ_Q and $\Sigma_{Q\cup\{C_1\}}$, respectively. Thus,

$$\Sigma_Q \;\not\models\; \Sigma_{Q\cup\{C_1\}} \tag{11}$$

holds. Note that a theory Σ' is negative if and only if, for all models α, $\alpha \in \Sigma'$ implies that each β such that $\beta \leq \alpha$ is also a model ($\beta \in \Sigma'$). Therefore, by (10) and (11), we have a maximal model α of Σ_Q such that $\alpha \notin \Sigma_{Q\cup\{C_1\}}$, i.e., $\alpha \in \Sigma_Q \setminus \Sigma_{Q\cup\{C_1\}}$, and $\alpha \vee e^{(j)} \notin \Sigma_Q$ for all j with $\alpha[x_j] = 0$, where $e^{(j)}$ denotes the unit vector such that $e^{(j)}[x_j] = 1$ and $e^{(j)}[x_i] = 0$ for all $i \neq j$. By (10), $\alpha \notin \Sigma_{Q\cup\{C_1\}}$ implies $\alpha \notin \Sigma$, and hence there is a clause C_2 in $\mathcal{F}$ not satisfied by α. Since $\Sigma_Q \models C$ holds for all negative clauses C in $\mathcal{F}$ (Step 1 of the algorithm guarantees this), C_2 is in $\mathrm{Horn}(\mathcal{F})$. Since α does not satisfy C_2, this clause must be of the following form:

$$C_2 \;=\; \bigvee_{x_j \in J_2} \overline{x}_j \vee x_l, \tag{12}$$

where $\alpha[x_j] = 1$ for all $j \in J_2$ and $\alpha[x_l] = 0$. Since $\alpha[x_l] = 0$ and α is maximal in Σ_Q, we have $\alpha \vee e^{(l)} \notin \Sigma_Q$; i.e., there must exist a negative clause $C_3 \in Q$ not satisfied by $\alpha \vee e^{(l)}$. Since this clause C_3 is satisfied by α, it must be of the following form:

$$C_3 \;=\; \bigvee_{x_j \in J_3} \overline{x}_j \vee \overline{x}_l, \tag{13}$$

where $\alpha[x_j] = 1$ for all $j \in J_3$.

Consider now the resolvent $C_4 = \bigvee_{x_j \in J_2 \cup J_3} \overline{x}_j$ of C_2 and C_3. Since $\alpha[x_j] = 1$ for all $j \in J_2 \cup J_3$, C_4 is not satisfied by α. Thus, C_4 is a negative implicate of Σ such that $\Sigma_Q \not\models C_4$. However, this is a contradiction because Step 3 finds a negative prime implicate C^* of Σ such that $C^* \models C_4$, and therefore $\Sigma_Q \models C_4$.

Let us consider next the time complexity of the algorithm. Clearly, Step 0 can be done in $O(1)$ time. In Step 1, it is known [18] that, for each clause $C \in \mathrm{Neg}(\mathcal{F})$, C^* can be found in $O(n|\mathcal{F}|)$ time, and (see [40]) that both checking if $C^* \notin Q$ and updating Q can be done in $O(n)$ time by using a binary tree as the data structure. Therefore, Step 1 requires $O(n|\mathrm{Neg}(\mathcal{F})||\mathcal{F}|)$ time.

Steps 2 and 3 are iterated at most $|\mathrm{NPI}(\Sigma)|$ times. Step 2 obviously can be done in $O(1)$ time. In Step 3, for each $C' \in \mathrm{Horn}(\mathcal{F})$, we can construct a resolvent C'' in $O(n)$ time (if any). Furthermore, similarly to Step 1, C^* can be found in $O(n|\mathcal{F}|)$ time, and both checking if $C^* \notin Q$ and updating Q can be done in $O(n)$ time. Thus, each iteration of Step 3 takes $O(n|\mathrm{Horn}(\mathcal{F})||\mathcal{F}|)$ time. Therefore, Steps 2 and 3 together require $O(n|\mathrm{Horn}(\mathcal{F})||\mathcal{F}||\mathrm{NPI}(\Sigma)|)$ time. In total, algorithm ALL-NPIS can be executed in

$$O(n|\mathcal{F}|(|\mathrm{Neg}(\mathcal{F})| + |\mathrm{Horn}(\mathcal{F})||\mathrm{NPI}(\Sigma)|))$$

time. The incremental polynomiality is also obvious because, similarly to the discussion above, the time required to output the i-th negative prime implicate $C \in \mathrm{NPI}(\mathcal{F})$ is polynomial in the input length and the size of the output generated so far.

We have proven the following lemma.

Lemma 5.9 *Given a Horn CNF $\mathcal{F}$ representing a theory $\Sigma \subseteq \{0,1\}^n$, algorithm* ALL-NPIS *generates all negative prime implicates of Σ in $O(n|\mathcal{F}|(|Neg(\mathcal{F})|+|Horn(\mathcal{F})||NPI(\Sigma)|))$ time, and furthermore, it is incrementally polynomial.* □

Based on ALL-NPIS, we can describe now an incrementally polynomial algorithm for FD-Inference when a Horn theory is represented by a Horn CNF.

Algorithm: ALL-FDS-FROM-CNF

Input: A Horn CNF $\mathcal{F}$ representing a theory $\Sigma \subseteq \{0,1\}^n$.

Output: All minimal functional dependencies of Σ.

Step 1. For each $x_i \in V = \{x_1, x_2, \ldots, x_n\}$, compute V_i (defined by (5)) and the Horn CNF $\mathcal{F}_{(x_l \leftarrow 0 \,|\, l \in V \setminus V_i)}$.

Step 2. Call Algorithm ALL-NPIS for $\mathcal{F}_{(x_l \leftarrow 0 \,|\, l \in V \setminus V_i)}$, and output $S \to x_i$ for each negative prime implicate $C = \bigvee_{x_j \in S} \overline{x}_j$ obtained.

The correctness of this algorithm follows from Lemmas 5.8 and 5.9. Let us analyze now its time complexity. Since $\mathcal{F}$ is Horn, we can compute each V_i in $O(|\mathcal{F}|)$ time by applying the forward chaining procedure (see [19, 20]), which adds variables to V_i one by one as soon as an appropriate clause in $\mathcal{F}$ can be "fired". Once V_i is known, $\mathcal{F}_{(x_l \leftarrow 0 \,|\, l \in V \setminus V_i)}$ can be obtained from $\mathcal{F}$ in $O(|\mathcal{F}|)$ time. Hence, Step 1 takes $O(|\mathcal{F}|)$ time. Note that $|\mathcal{F}_{(x_l \leftarrow 0 \,|\, l \in V \setminus V_i)}| \leq |\mathcal{F}|$, $|\mathrm{Neg}(\mathcal{F}_{(x_l \leftarrow 0 \,|\, l \in V \setminus V_i)})| \leq |\mathrm{Neg}(\mathcal{F})| + |\mathrm{Horn}(\mathcal{F})|$ and $|\mathrm{Horn}(\mathcal{F}_{(x_l \leftarrow 0 \,|\, l \in V \setminus V_i)})| \leq |\mathrm{Horn}(\mathcal{F})|$. Therefore, for each $x_i \in V$, the amount of time

$$\begin{aligned} &O(|\mathcal{F}| + n|\mathcal{F}|(|\mathrm{Neg}(\mathcal{F})| + |\mathrm{Horn}(\mathcal{F})|) + n|\mathcal{F}||\mathrm{Horn}(\mathcal{F})||\mathrm{NPI}(\Sigma_i)|) \\ &\qquad = O(n|\mathcal{F}|(|\mathrm{Neg}(\mathcal{F})| + |\mathrm{Horn}(\mathcal{F})|) + n|\mathcal{F}||\mathrm{Horn}(\mathcal{F})||\mathrm{NPI}(\Sigma_i)|) \end{aligned}$$

is needed, where Σ_i is defined by (6). In total, Algorithm ALL-FDS-FROM-CNF requires

$$O(n^2|\mathcal{F}|(|\mathrm{Neg}(\mathcal{F})| + |\mathrm{Horn}(\mathcal{F})|) + n|\mathcal{F}||\mathrm{Horn}(\mathcal{F})||\mathcal{M}(\Sigma)|)$$

time. Furthermore, its incremental polynomiality is immediate from the incremental polynomiality of ALL-NPIS.

Theorem 5.10 *Given a Horn CNF $\mathcal{F}$ representing a theory $\Sigma \subseteq \{0,1\}^n$, algorithm* ALL-FDS-FROM-CNF *constructs $\mathcal{M}(\Sigma)$ in*

$$O(n^2|\mathcal{F}|(|Neg(\mathcal{F})| + |Horn(\mathcal{F})|) + n|\mathcal{F}||Horn(\mathcal{F})||\mathcal{M}(\Sigma)|)$$

time, and furthermore, this algorithm is incrementally polynomial. □

Finally, we consider the FD-Inference problem when a Horn theory is represented as the Horn envelope of a set of models. The following theorem shows that this problem appears to be harder than the case in which a Horn theory is represented by a Horn CNF. To formulate the theorem, we shall need the following notions. A model α of a theory Σ is called *minimal* if Σ does not have a model $\beta < \alpha$. A theory Σ is called positive if for any its model α every $\beta > \alpha$ is also a model. Let $T(\Sigma)$ denote the set of minimal models of Σ. The problem of dualizing a positive theory Σ consists in computing $T((\Sigma)^d)$ from $T(\Sigma)$, where $(\Sigma)^d = \{\alpha \,|\, \overline{\alpha} \notin \Sigma\}$.

Theorem 5.11 *There exists a polynomial total time algorithm for the FD-Inference problem when a Horn theory is represented as the Horn envelope of a set of models if and only if there exists a polynomial total time algorithm for dualizing a positive theory.*

Proof. It is known [13] that the problem of dualizing a positive theory is equivalent to the problem of generating all minimal solutions y subject to

$$Ay \;\geq\; e, \quad \text{and} \quad y_j \in \{0,1\} \;\text{ for all } j = 1,2,\ldots m, \tag{14}$$

where A denotes a 0-1 $l \times m$ matrix (i.e., $a_{ij} \in \{0,1\}$ for all i and j), and e is the l-dimensional column vector of 1's. We call this problem ALL-SET-COVER. We show below that the FD-Inference problem, when a Horn theory is represented as the Horn envelope of a set of models, has a polynomial total time algorithm if and only if ALL-SET-COVER has a polynomial total time algorithm.

Let us first show the "only if" part. Without loss of generality, we assume that A does not have zero rows, and that no two columns of it are comparable. Let us transform an arbitrary instance of ALL-SET-COVER into an instance of our problem in the following way:

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

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 $n = m + 1$, and let $V = \{x_1, x_2, \ldots, x_n\}$ denote the set of all variables of Σ, i.e., x_i is the variable corresponding to the i-th column.

Since A has no comparable columns, it follows from the first condition of Lemma 3.3 that all functional dependencies in $H(\Sigma)$ have x_n in the right-hand side. It follows from Corollary 3.4 that minimal functional dependencies in $H(\Sigma)$ are in one-to-one correspondence with the minimal solutions of (14). This proves the "only if" part.

For the "if" part, we provide the following algorithm:

Algorithm: ALL-FDS-FROM-MODELS

Input: A theory $\Sigma \subseteq \{0,1\}^n$ representing a Horn envelope $H(\Sigma)$.

Output: All minimal functional dependencies of $H(\Sigma)$.

Step 1. For each $y \in V = \{x_1, x_2, \ldots, x_n\}$, compute the maximum subset $X_y \subseteq V \setminus y$, such that $\alpha[X_y] = (11\cdots 1)$ for all $\alpha \in \Sigma_1^y$.

Step 2. For each $y \in V$, compute all $X' \subseteq X_y$ satisfying (2) and (3) of Corollary 3.4, and output $X' \to y$.

The correctness of this algorithm is immediate from Corollary 3.4. For each $y \in V$, we can compute $X_y \subseteq V \setminus y$ in $O(n|\Sigma|)$ time, and as discussed in the proof of the "only if" part, we can see that the problem of computing all $X' \subseteq X_y$ satisfying (2) and (3) of Corollary 3.4 is equivalent to ALL-SET-COVER for the $|\Sigma_0^y| \times |X_y|$ matrix whose row vectors are $(1, 1, \ldots, 1) - \alpha$ for $\alpha \in \Sigma_0^y[X_y]$. Hence, for each $y \in V$, the problem of computing all minimal functional dependencies $X' \to y$ is polynomially equivalent to ALL-SET-COVER. This proves the "if" part. □

The problem of dualizing a positive theory has been well studied. Although it is still unknown whether a polynomial total time algorithm exists for this problem, a wide variety of computationally equivalent problems have been discovered (see e.g., [3, 13]), and a quasi-polynomial algorithm has been developed (see [16]).

6 Condensation of Horn Theories

The procedure of condensation introduced in Subsection 2.3 aims at simplifying a given theory by eliminating variables that are functionally dependent on other variables in the theory. 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 4.3 states that a minimal functional dependency $X \rightarrow y$ 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 Subsection 2.5, i.e., 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 $X \rightarrow y$ 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:

$$\bigwedge_{x \in X} x \vee C = \bigwedge_{x \in X} (x \vee C). \tag{15}$$

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 size 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 i-th step of the condensation procedure.

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

Proof. By superposing the functions corresponding to the functional dependencies used up to the i-th step in the condensation procedure, 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. 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 (15) 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 $O(|V_i|)$ and the number of clauses can increase at most by a factor of $O(|V_i|)$, resulting in the estimate 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:

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

This CNF depends on $3n+2$ variables and its length is $9n+2$. It can be checked easily that the condensation of this CNF will result in the following Horn CNF:

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

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 $O(|V|^2|\mathcal{F}|)$. The proof of Lemma 6.1 shows that the size of a Horn CNF does not explode significantly in the condensation procedure because functional dependencies in Horn theories are simply single positive term Boolean functions, and a superposition of single positive term Boolean functions is again a single positive term Boolean function:

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

As was mentioned in Subsection 2.3 and demonstrated in Subsection 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 4.4). 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 follows from Lemma 2.2. We can therefore assume without loss of generality that the condensation procedure uses first simple functional dependencies, i.e., the procedure of 2-condensation described in Section 4 is finished first. We can therefore assume in what follows that Σ is 2-condensed.

Theorem 6.2 *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 $G(\mathcal{M}(\Sigma))$ introduced in Section 4. 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 holding in Σ. Any of these functional dependencies can be used by the condensation procedure to eliminate the variable.

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 had an incoming arc before a minimal functional dependency $C \to y$ was used (where $x \neq y$), then x would again have an incoming arc in the resulting graph. Since the set of functional dependencies of the reduced theory consists of those and only those functional dependencies of the original theory that do not involve y, the 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 4.10, the graph $G(\mathcal{M}(\Sigma))$ does not have oriented cycles, and therefore $x \notin C$. Then $CC' \to x$ is a non-trivial functional dependency which holds in Σ, and there exists $C'' \subseteq C \cup C'$ such that $C'' \to x$ is a minimal functional dependency holding in Σ. Since $y \notin C''$, this functional dependency will hold in the reduced theory, and x will have an incoming arc in the resulting graph. □

The condensation of a Horn theory requires the knowledge of functional dependencies holding in this theory. 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. In contrast with this fact, we will 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 6.2 that this set V^c is uniquely defined.

Theorem 6.3 *Given a theory Σ, a theory Σ^c whose Horn envelope $H(\Sigma^c)$ is the condensation of the Horn envelope $H(\Sigma)$, and the terms representing all the variables in $V \setminus V^c$ through the variables in V^c can be constructed in $O(|V|^3|\Sigma|)$ time.*

Theorem 6.4 *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 $O(|V|^2|\mathcal{F}|)$ time.*

Proof. The underlying reason for these two theorems to hold is the fact that condensation can be carried out using a very limited number of functional dependencies. More precisely, as follows from the proof of Theorem 6.2, it is sufficient to construct a single minimal functional dependency $X \to y$ for every variable y having incoming arcs in $G(\mathcal{M}(\Sigma))$, and use this dependency for eliminating y. For every variable $y \in V$ such a minimal functional dependency can be easily found, if exists at all. The procedure consists in checking whether

$V \setminus y \to y$ is a functional dependency, and if yes, then dropping variables from $V \setminus y$ as long as possible to arrive at a minimal functional dependency. It follows from Theorem 3.1 that for the CNF representation this can be done for all the variables in $O(|V|^2|\mathcal{F}|)$ time. Theorem 3.5 shows that for the model representation this will take $O(|V|^3|\Sigma|)$ 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. Then we can start from V^c, follow the arcs in G', and use superposition of 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 $O(|V^c||V|)$ time, and all the vertices in the graph can be processed in $O(|V|^2|V^c|)$ 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 $O(|V|^2|V^c|) \leq O(|V|^3)$, we have proven the estimate of Theorem 6.3.

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 (15)) can be done in $O(|V^c||\mathcal{F}|)$ time. Finally, the expansion (15) can be done in $O(|V^c|^2|\mathcal{F}|)$ time, thus proving the estimate of Theorem 6.4. $\square$

It follows from Lemma 6.1 that the estimate of Theorem 6.4 cannot be improved. It is important to note that although the process of condensation is described as a sequential one, where variables are eliminated one by one, the estimate of Theorem 6.4 is obtained for the implementation of the condensation procedure that avoids the possible increase in the size of intermediate CNFs by first deriving the final expressions of the variables in $V \setminus V^c$ through the variables in V^c, then substituting all these expressions in $\mathcal{F}$ without carrying out the expansion (15), and finally rewriting the formula as a Horn CNF using (15).

While, as shown in Theorem 6.2, the condensed theory is uniquely defined by the original theory, the expressions of the variables in $V \setminus V^c$ through the variables in V^c may not be unique. Consider, for example, the following Horn CNF:

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

It can be seen easily that the condensed CNF is

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

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 for the condensation procedure to find the shortest possible expressions of the variables in $V \setminus V^c$ through the variables in V^c, if this were computationally tractable. We therefore have to 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. [17]).

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 such that $\sum_{j=1}^{m} y_j \leq k$ and the inequality (14) is satisfied?

In the following, we shall assume that matrix A does not have zero rows, and that no two columns of it are comparable, since the Set Covering problem remains NP-complete even if these conditions hold.

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:

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

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, and let the number k remain as it is given. Since A does not have comparable columns, it follows from the first condition of Lemma 3.3 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. It follows from Corollary 3.4 that minimal functional dependencies in $H(\Sigma)$ are in one-to-one correspondence with the minimal solutions of (14). 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". $\square$

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:

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

Without loss of generality, A does not have a column of all ones. Then it can be seen that all functional dependencies in this Horn theory have x in the right-hand side, and are of the following form:

$$\bigcup_{i\in I} y_i \cup \bigcup_{j\in J} z_j \to x, \tag{16}$$

where I and J satisfy the condition that for every $i \in V \setminus I$ 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, $j = 1, 2, \ldots, m$.

Since, without loss of generality, A does not have zero rows, there exists a minimal functional dependency with the minimum cardinality right-hand side such that $I = \emptyset$. Clearly, if $I = \emptyset$, then there is a one-to-one correspondence between the sets J and the vectors y satisfying the inequality (14). 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 both the representation of a Horn theory as a Horn CNF, and as the Horn envelope of a set of models. It was shown that for both these representations one can recognize in polynomial time whether a given functional dependency holds in a given Horn theory, while it is computationally difficult to check whether a given functional dependency holds in a theory represented by a given general CNF. We also showed 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 established a correspondence between minimal functional dependencies in a Horn theory and some prime implicates of the theory, and proved that the set of models of the Horn theory defined by all minimal functional dependencies is a superset of the set of models 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 the set of all minimal functional dependencies in a Horn theory is quasi-acyclic, i.e., all its cycles (if any) are created by logically equivalent variables in the theory. 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 studied the problem of inferring functional dependencies in a given Horn theory, and 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 therefore considered the complexity of inferring all minimal functional dependencies as a function of the size of the output, developed an incrementally polynomial algorithm for inferring all minimal functional dependencies in a theory given as a Horn CNF, and showed 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 with the existence of a polynomial total time algorithm for the well known problem of dualizing a positive theory.

We introduced the procedure of condensing a theory by eliminating those variables that are functionally dependent on other variables, as long as no functional dependency holds in the resulting theory. In the case of general Boolean theories the resulting condensed theory may depend on the choice of functional dependencies to be used in condensation, and the functional expressions of eliminated variables through the remaining variables may be too complicated to make condensation computationally advantageous. We showed that the simplicity of functional form and the acyclicity of structure of functional dependencies in Horn theories make their condensation uniquely defined and computationally beneficial. Although the inference of functional dependencies in a Horn theory may be computationally difficult, it was shown that any Horn theory can be totally condensed in polynomial time whether it is represented as a Horn CNF or as the Horn envelope of a set of models.

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 making 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: it can be recognized in polynomial time whether a given CNF can be renamed as a Horn CNF, and if yes, a renaming making the CNF Horn can also be determined in polynomial time (see [32, 7]). 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 specify the set of models of the theory by appropriately redefining 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 its prime CNF 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 that hold in a theory. However, the functional form of the expression $y = f(X)$ will change in accordance with the renaming. It is easy to see that in the case of renamable Horn theories this functional form does not become significantly more complicated than in the case of Horn theories. More specifically, a functional dependency in a renamable Horn theory has the form of either a single term or a single clause, which is not necessarily positive any more.

Renamable Horn CNFs and 2-CNFs 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, 6]. 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 have the form similar to functional dependencies in a renamable Horn theory, i.e., they are either a single term or a single clause.

Acknowledgments

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 support is gratefully acknowledged. The authors are grateful to Thomas Eiter for his useful comments.

References

[1] W.W. Armstrong. Dependency structures of database relationships, *Proceedings IFIP'74*, North-Holland, Amsterdam, 1974, 580 - 583.

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

[3] J.C. Bioch and T. Ibaraki. Complexity of identification and dualization of positive Boolean functions, *Information and Computation*, **123** (1995), 51 - 75.

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

[5] E. Boros, Y. Crama, P.L. Hammer, M. Saks. A complexity index for satisfiability problems, *SIAM Journal on Computing*, **23** (1994), 45 - 49.

[6] E. Boros, P.L. Hammer and X. Sun. Recognition of q-Horn formulae in linear time, *Discrete Applied Mathematics*, **55** (1994), 1 - 13.

[7] V. Chandru, C.R. Coullard, P.L. Hammer, M. Montañez, X. Sun. On renamable Horn and generalized Horn functions, *Annals of Mathematics and Artificial Intelligence*, **1** (1990), 33 - 47.

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

[9] E.F. Codd. A relational model for large shared data banks, *Communications of the ACM*, Vol. 13, No. 6, June 1970, 377 - 387.

[10] R. Dechter and J. Pearl. Structure identification in relational data, *Artificial Intelligence*, **58** (1992), 237 - 270.

[11] C. Delobel and R.G. Casey. Decomposition of a data base and the theory of Boolean switching functions, *IBM Journal of Research and Development*, Vol. 17, No. 5, September 1973, 374 - 386.

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

[13] T. Eiter and G. Gottlob. Identifying the minimal transversals of a hypergraph and related problems, *SIAM Journal on Computing*, **24** (1995), 1278 - 1304.

[14] R. Fagin. Functional dependencies in a relational database and propositional logic, *IBM Journal of Research and Development*, Vol. 21, No. 6, November 1977, 534 - 544.

[15] R. Fagin. Horn clauses and database dependencies, *Journal of the ACM*, **29** (4) (1982), 952 - 985.

[16] M.L. Fredman and L. Khachiyan. On the complexity of dualization of monotone disjunctive normal forms, *Journal of Algorithms*, **21** (3) (1996), 618 - 628.

[17] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman & Co., San Francisco, 1979.

[18] P.L. Hammer and A. Kogan. Horn functions and their DNFs, *Information Processing Letters*, **44** (1992), 23 - 29.

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

[20] P.L. Hammer and A. Kogan. Quasi-acyclic propositional Horn knowledge bases: Optimal compression, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, No. 5, October 1995, 751 - 762.

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

[22] A. Horn. On sentences which are true of direct unions of algebras, *Journal of Symbolic Logic*, **16** (1951), 14 - 21.

[23] T. Ibaraki, A. Kogan and K. Makino. *Functional Dependencies in q-Horn Theories*, unpublished manuscript, 1998.

[24] D.S. Johnson, M. Yannakakis and C.H. Papadimitriou. On generating all maximal independent sets, *Information Processing Letters*, **27** (1988), 119 - 123.

[25] H. Kautz, M. Kearns and B. Selman. Horn approximations of empirical data, *Artificial Intelligence*, **74** (1995), 129 - 145.

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

[27] R. Khardon. Translating between Horn representations and their characteristic models, *Journal of Artificial Intelligence Research*, **3** (1995), 349 - 372.

[28] R. Khardon, H. Mannila and D. Roth. *Reasoning with Examples: Propositional Formulae and Database Dependencies*, Technical Report TR-15-95, Aiken Computation Laboratory, Harvard University, Cambridge, MA, 1995.

[29] R. Khardon and D. Roth. Reasoning with models, *Artificial Intelligence*, **87** (1996), 187 - 213.

[30] J. Kivinen and H. Mannila. Approximate inference of functional dependencies from relations, *Theoretical Computer Science*, **149** (1995), 129 - 149.

[31] E. Lawler, J.K. Lenstra and A.H.G. Rinnooy Kan. Generating all maximal independent sets: NP-hardness and polynomial-time algorithms, *SIAM Journal on Computing*, **9** (1980), 558 - 565.

[32] H.R. Lewis. Renaming a set of clauses as a Horn set, *Journal of the ACM*, **25** (1978), 134 - 135.

[33] C.L. Lucchesi and S.L. Osborn. Candidate keys for relations. *Journal of Computer and System Sciences*, **17** (1978), 270 - 279.

[34] D. Maier. *The Theory of Relational Databases*. Computer Science Press, Rockville, MD, 1983.

[35] H. Mannila and K.-J. Räihä. *Design of Relational Databases*. Addison-Wesley, Wokingham, 1992.

[36] H. Mannila and K.-J. Räihä. Algorithms for inferring functional dependencies, *Data and Knowledge Engineering*, **12** (1994), 83 - 99.

[37] J.C.C. McKinsey. The decision problem for some classes of sentences without quantifiers, *Journal of Symbolic Logic*, **8** (3) (1943), 61 - 77.

[38] M. Minoux. LTUR: A simplified linear-time unit resolution algorithm for Horn formulae and computer implementation, *Information Processing Letters*, **29** (1988), 1 - 12.

[39] M. Minoux. The unique Horn-Satisfiability problem and quadratic Boolean equations, *Annals of Mathematics and Artificial Intelligence*, **6** (1992), 253 - 266.

[40] J.S. Provan and M.O. Ball. Efficient recognition of matroids and 2-monotonic systems, *Applications of Discrete Mathematics*, edited by R. Ringeisen and F. Roberts, SIAM, Philadelphia, (1988), 122 - 134.

[41] W. Quine. A way to simplify truth functions, *American Mathematical Monthly*, **62** (1955), 627 - 631.

[42] Y. Sagiv, C. Delobel, D.S. Parker, R. Fagin. An equivalence between relational database dependencies and a fragment of propositional logic, *Journal of the ACM*, **28** (3) (1981), 435 - 453.

[43] J.D. Ullman. *Principles of Database and Knowledge-Base Systems*. Computer Science Press, New York, NY: Vol. I: Classical Database Systems, 1988; Vol. II: The New Technologies, 1989.

[44] Yu.I. Zhuravlev and 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.