Available online at www.sciencedirect.com

ScienceDirect

Journal of Computer and System Sciences 74 (2008) 1103–1115

JOURNAL OF COMPUTER AND SYSTEM SCIENCES

www.elsevier.com/locate/jcss

Structure identification of Boolean relations and plain bases for co-clones

Nadia Creignou [a], Phokion Kolaitis [b,1], Bruno Zanuttini [c,*]

[a] *LIF, UMR CNRS 6166, Université de la Méditerranée, 163, avenue de Luminy, 13 288 Marseille, France*
[b] *IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120-6099, USA*
[c] *GREYC, UMR CNRS 6072, Université de Caen, Boulevard du Maréchal Juin, 14032 Caen, France*

Received 4 July 2006; received in revised form 26 March 2007

Available online 4 March 2008

Abstract

We give a quadratic algorithm for the following structure identification problem: given a Boolean relation R and a finite set S of Boolean relations, can the relation R be expressed as a conjunctive query over the relations in the set S? Our algorithm is derived by first introducing the concept of a plain basis for a co-clone and then identifying natural plain bases for every co-clone in Post's lattice. In the process, we also give a quadratic algorithm for the problem of finding the smallest co-clone containing a Boolean relation.

Keywords: Computational complexity; Boolean constraints; Closure properties; Boolean co-clones; Inverse satisfiability

1. Introduction and summary of results

The *structure identification problem* [10] has been recognized as a basic algorithmic problem arising in several different areas of artificial intelligence and computer science, such as knowledge representation and computational learning theory. Informally, structure identification is the problem of determining whether a given relation can be "represented" by a formula in some logical formalism. The given relation can be thought of as a set of *observations* or a *state of knowledge*; thus, the structure identification problems asks whether a given set of observations coincides with the set of *models* of some formula in the logical formalism under consideration.

The structure identification problem can be formalized in different ways by considering different logical formalisms of interest. The most well-studied formalization of this problem, originally articulated by Dechter and Pearl [10], has become known as the Inverse Satisfiability Problem [8,13], which here will be denoted as InvSat. The input to an instance of the InvSat problem is a relation R and a finite set S of relations over the same domain as R; the question is whether R is the set of models of some CNF(S)-formula, i.e., a formula that is the conjunction of atomic

* Corresponding author.
E-mail addresses: creignou@lif.univ-mrs.fr (N. Creignou), kolaitis@almaden.ibm.com (P. Kolaitis), bruno.zanuttini@info.unicaen.fr (B. Zanuttini).
[1] On leave from UC Santa Cruz.

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2008.02.005

formulas of the form $T(x_1, \ldots, x_n)$, where T is a relation in the set S. In other words, the question is whether R can be obtained from the relations in S using finite Cartesian products and identification of variables. Clearly, CNF(S) formulas generalize Boolean formulas in conjunctive normal form. Note that the InvSat problem has connections to both constraint satisfaction and database theory. Indeed, from a constraint-satisfaction perspective, InvSat asks whether the relation R is the set of solutions of a *constraint network* [9] built from relations in S, while, from a database-theoretic perspective, it asks whether R can be expressed as a *relational join* [1] involving relations from S. Over the Boolean domain, InvSat is known to be a coNP-complete problem [13]. As defined above, InvSat is a *uniform* problem, in the sense that both a relation R and a finite set S of relations are part of the input. By keeping the set S fixed, we obtain a family of *non-uniform* decision problems InvSat(S) (one for each fixed set S) in which the input is just a relation R. Kavvadias and Sideri [13] proved a *Dichotomy Theorem* that completely characterizes the computational complexity of all non-uniform InvSat(S) problems over the Boolean domain, provided the set S contains the singletons $\{0\}$ and $\{1\}$ as members (equivalently, the constants 0 and 1 are allowed in CNF(S)-formulas). Specifically, they showed that if the set S consists of Boolean relations all of which are Horn,[2] or all of which are dual Horn, or all of which are bijunctive, or all of which are affine, then InvSat(S) is in P; in all other cases InvSat(S) is coNP-complete.

In view of the intractability of InvSat, it is natural to ask: are there are tractable variants of the structure identification problem in which formulas from more powerful formalisms are used? If S is a set of relations, then the class of $\exists$CNF(S)-formulas consists of all expressions of the form $\exists \mathbf{y} \varphi(\mathbf{x}, \mathbf{y})$, where $\varphi(\mathbf{x}, \mathbf{y})$ is a CNF(S)-formula and $\mathbf{x}$, $\mathbf{y}$ are tuples of variables. This means that a relation R is the set of models of some $\exists$CNF(S)-formula if and only if R can be obtained from the relations in S using finite Cartesian products, identification of variables, and projections. In universal algebra, $\exists$CNF(S)-formulas are known as *primitive positive* formulas, and they play an important role in the Galois connection between clones of functions and co-clones of relations [15,16]. They also play a crucial role in the proof of Schaefer's Dichotomy Theorem for generalized satisfiability Sat(S) problems [18], which is the first, and arguably the most influential, Dichotomy Theorem in computational complexity. In database theory, $\exists$CNF(S)-formulas are known as *conjunctive queries* or *select-project-join queries*, and they constitute the most frequently asked queries in relational database systems. In fact, conjunctive queries are directly expressed in SQL through the `Select-From-Where` construct, the main building block of SQL [12].

The $\exists$-InvSat problem asks: given a relation R and a finite set S of relations over the same domain, is R is the set of models of some $\exists$CNF(S)-formula? In other words, is R definable by a conjunctive query with relations from S? Dalmau [8, Lemma 42] showed that, over arbitrary finite domains, $\exists$-InvSat is a decidable problem. Since the algorithm given in [8] has a running time of several exponentials, Dalmau raised the question of designing more efficient algorithms for $\exists$-InvSat or establishing lower bounds for the complexity of this problem. He also considered the non-uniform version of this problem, that is, the family $\exists$-InvSat(S) of decision problems obtained by fixing the set S of relations (thus, the input to $\exists$-InvSat(S) is just a relation R). Using the fact that every Boolean clone is finitely generated, Dalmau [8, Corollary 11] pointed out that, for each fixed finite set S of Boolean relations, $\exists$-InvSat(S) is a polynomial-time solvable problem: the algorithm simply checks that the relation R is closed under every function in the basis for the clone associated with the smallest co-clone containing the relations in S. The running time, however, is bounded by a polynomial whose degree depends on the set S, and can be arbitrarily high. The reason is that if S is a set of relations in one of the co-clones in the infinite part of Post's lattice, then the bases for the associated clones contain functions of arbitrarily large arity.

In this paper, we show that, over the Boolean domain, the (uniform) $\exists$-InvSat problem is solvable in time quadratic in the size of the relation R and the set S. As an immediate consequence, we have that, over the Boolean domain, each non-uniform $\exists$-InvSat(S) problem is also solvable in quadratic time. This result contrasts sharply with the intractability of the InvSat problem; it also reveals the difference that the choice of the logical formalism can make on the complexity of the structure identification problem.

Our quadratic algorithm for the $\exists$-InvSat problem is designed in two stages. In the first stage, we obtain a quadratic algorithm for the restriction of $\exists$-InvSat to sets S of relations in one of the co-clones in the infinite part of Post's lattice. This is achieved by introducing the concept of a *plain basis* for a co-clone, making use of *prime* CNF representations of relations, and exhibiting natural plain bases for the co-clones in the infinite part of Post's lattice. By

[2] Dechter and Pearl [10] had already shown that if S is a set of Horn relations, then InvSat(S) is in P.

definition, a *plain basis* for a co-clone $\mathcal{I}$ is a set B of relations in $\mathcal{I}$ such that every relation in $\mathcal{I}$ is the set of models of a CNF($\mathcal{I}$)-formula. Thus, the notion of a plain basis is a strengthening of the notion of a *basis* for a co-clone $\mathcal{I}$, which, by definition, is a set B of relations in $\mathcal{I}$ such that every relation in $\mathcal{I}$ is definable by a ∃CNF($\mathcal{I}$)-formula; natural bases for all Boolean co-clones have been exhibited in [5]. Note that our quadratic algorithm for the ∃-INVSAT problem restricted to the co-clones in the infinite part of Post's lattice easily yields a cubic algorithm for the full ∃-INVSAT over the Boolean domain; this is so because the clones in the finite part of Post's lattice have bases consisting of functions of arities at most three. As it turns out, however, we can do better than this. Indeed, in the second stage, we exhibit natural plain bases for all Boolean co-clones, and then use these plain bases to derive a quadratic algorithm for the full ∃-INVSAT problem over the Boolean domain.

In the process of solving the ∃-INVSAT problem over the Boolean domain, we also use plain bases to give a quadratic algorithm for the following problem, which is of independent interest: given a Boolean relation, find the smallest co-clone to which it belongs.

2. Basic notions and background

This section contains the definitions of the basic notions and a minimum amount of the necessary background material.

2.1. Boolean formulas in conjunctive normal form and prime implicates

A *literal* is either a variable x (*positive* literal) or a negated variable $\neg x$ (*negative* literal). A *clause* is a finite disjunction $(\ell_1 \vee \cdots \vee \ell_k)$ of literals. A Boolean formula is said to be in *Conjunctive Normal Form* (CNF) if it is a conjunction of clauses. We refer to formulas in conjunctive normal form as CNF-formulas.

If V is a set of variables, then an *assignment on* V is a mapping from V to $\{0,1\}$. If V is a set of variables and φ is a CNF-formula over a subset of V, then a *model of* φ *over* V is an assignment on V that satisfies φ. A formula is *satisfiable* if it has at least one model. If φ_1 and φ_2 are two propositional formulas over sets of variables V_1 and V_2, respectively, then we say that φ_1 (*logically*) *entails* φ_2 if every model of φ_1 over $V_1 \cup V_2$ is a model of φ_2. We also say that φ_1 and φ_2 are (*logically*) *equivalent*, denoted $\varphi \equiv \varphi'$, if their sets of models over $V_1 \cup V_2$ coincide.

An *n-ary Boolean relation* is a set $R \subseteq \{0,1\}^n$. We will be interested in the following correspondence between Boolean relations and propositional formulas. Every n-ary Boolean relation R can be viewed as a set of assignments to the variables $x_1, x_2, \ldots, x_n$, i.e., we view every vector $m = (m_1, \ldots, m_n) \in R$ as the assignment of value m_i to variable x_i, for $i \in \{1, \ldots, n\}$. We say that a propositional formula φ over the variables $x_1, \ldots, x_n$ *represents* R if R is the set of models of φ. A relation is *Horn* (respectively, *dual Horn*) if it is the set of models of some Horn formula (respectively, dual Horn formula). A relation is *bijunctive* if it is the set of models of some 2CNF formula.

An important notion that we will use repeatedly in what follows is that of a *prime implicate*. Let φ be a propositional formula. A clause $C = (\ell_1 \vee \cdots \vee \ell_k)$ is said to be a *prime implicate* of φ if φ entails C, but it does not entail any proper subclause of C. This means that φ entails C, but there is no $i \in \{1, \ldots, k\}$ such that φ entails the clause $(\ell_1 \vee \cdots \vee \ell_{i-1} \vee \ell_{i+1} \vee \cdots \vee \ell_k)$. A CNF-formula φ is said to be *prime* if all its clauses are prime implicates of it. For example, the CNF-formula $(x_1 \vee x_2) \wedge (\neg x_2 \vee \neg x_3)$ is prime. In contrast, the CNF-formula $(x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_4) \wedge (x_3 \vee \neg x_4)$ is not prime because it entails $(x_2 \vee x_3)$, which is a proper subclause of $(x_1 \vee x_2 \vee x_3)$. It is easy to see that every Boolean relation is represented by some prime CNF formula.

2.2. CNF(S)-*formulas,* ∃CNF(S)-*formulas,* INVSAT(S)*, and* ∃-INVSAT(S)

Let S be a (possibly infinite) set of Boolean relations. For every relation R in S, let R' be a relation symbol of the same arity as R.

- A CNF(S)-formula is a finite conjunction of expressions (sometimes called *generalized clauses*) of the form $T'(x_1, \ldots, x_k)$, where each T' is the relation symbol representing a relation T in S, and $x_1, \ldots, x_k$ are Boolean variables that need not be distinct from each other (i.e., identification of variables is allowed).
- A ∃CNF(S)-formula is an expression of the form $\exists y_1 \ldots y_m \varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$, where $\varphi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ is a CNF(S)-formula.

The concepts of a *model of a* CNF(S)*-formula* and of a *model of a* ∃CNF(S)*-formula* are defined in a standard way by assuming that the variables range over the set $\{0,1\}$ and each relation symbol T' is interpreted by the corresponding relation T in S. For notational simplicity, in what follows we will use the same symbol, say T, for both a Boolean relation T and the relation symbol T' representing it.

- SAT(S) is the following decision problem: given a CNF(S)-formula φ, is it satisfiable (i.e., does it have at least one model)?

Numerous well-known variants of Boolean satisfiability can be cast as SAT(S) problems, for appropriately chosen sets S of logical relations. For example, the prototypical NP-complete problem 3-SAT coincides with the problem SAT(S), where $S = \{R_0, R_1, R_2, R_3\}$ and $R_0 = \{0,1\}^3 - \{(0,0,0)\}$ (expressing the clause $(x \vee y \vee z)$), $R_1 = \{0,1\}^3 - \{(1,0,0)\}$, $R_2 = \{0,1\}^3 - \{(1,1,0)\}$, and $R_3 = \{0,1\}^3 - \{(1,1,1)\}$. Similarly, the well-known NP-complete problem POSITIVE-1-IN-3-SAT is precisely SAT(S), where $S = \{R_{1/3}\}$ is the singleton consisting of the relation $R_{1/3} = \{(1,0,0), (0,1,0), (0,0,1)\}$.

Schaefer [18] completely characterized the computational complexity of all SAT(S) problems, as S ranges over finite sets of Boolean relations. Specifically, he identified all finite sets S for which SAT(S) is in P, and showed that SAT(S) is NP-complete for all other finite sets S. In particular, assume that S is a finite set of Boolean relations containing the singletons $\{0\}$ and $\{1\}$ as members. Schaefer showed that if S is Horn,[3] or dual Horn, or bijunctive, or affine, then SAT(S) is in P; in all other cases, SAT(S) is NP-complete. This is called a *Dichotomy Theorem* because Ladner [14] has shown that, assuming P $\neq$ NP, there are decision problems in NP that are neither NP-complete nor in P. Thus, Schaefer's Theorem implies that no SAT(S) problem is of the kind discovered by Ladner.

- INVSAT(S) is the following decision problem: given a Boolean relation R, is R the set of models of some CNF(S)-formula?

INVSAT(S) is a structure identification problem as it asks whether a given Boolean relation (a set of observations) can be represented by a formula in some particular formalism. As mentioned in Section 1, Kavvadias and Sideri [13] proved a Dichotomy Theory for SAT(S) that parallels Schaefer's Dichotomy Theorem. Specifically, assuming S is a finite set of Boolean relations containing the singletons $\{0\}$ and $\{1\}$ as members, the polynomial-time cases of INVSAT(S) coincide with the polynomial-time cases of SAT(S); in all other cases, INVSAT(S) is coNP-complete.

- ∃-INVSAT(S) is the following decision problem: given a Boolean relation R, is R the set of models of some ∃CNF(S)-formula?

∃-INVSAT(S) is a structure identification problem that asks whether a given Boolean relation can be represented by a formula in a certain logical formalism that is more expressive than the logical formalism in INVSAT(S). As mentioned in Section 1, Dalmau [8] showed that, for every finite set S of Boolean relations, ∃-INVSAT(S) is solvable in polynomial time via an algorithm whose running time depends on the set S. Here, we shall show that every ∃-INVSAT(S) problem is solvable in quadratic time. In fact, we shall give a quadratic algorithm for the *uniform* structure identification problem ∃-INVSAT, which contains all ∃-INVSAT(S) problems as special cases.

- ∃-INVSAT is the following decision problem: given a finite set S of Boolean relations and a Boolean relation R, is R the set of models of some ∃CNF(S)-formula?

2.3. Post's lattice

An *n-ary Boolean function* is a function $f:\{0,1\}^n \mapsto \{0,1\}$. If f is an n-ary Boolean function and $g_1,\dots,g_n$ are all m-ary Boolean functions, then their *composition* $f(g_1,\dots,g_n)$ is the m-ary Boolean function defined by $f(g_1,\dots,g_n)(a_1,\dots,a_m) = f(g_1(a_1,\dots,a_m),\dots,g_n(a_1,\dots,a_m))$, for every $(a_1,\dots,a_m) \in \{0,1\}^m$. For $n \geqslant$

[3] This means that every relation in S is the set of models of some Horn formula; the other cases are defined in a similar manner.

$m \geqslant 1$, the *projection function* $\pi_{n,m}$ is defined by $\pi_{n,m}(x_1, \ldots, x_n) = x_m$. If f is a Boolean function, then the *dual of* f is the Boolean function dual(f) defined by dual$(f)(a_1, \ldots a_n) = \overline{f(\overline{a_1}, \ldots, \overline{a_n})}$.

A (Boolean) *clone* is a set of Boolean functions closed under composition and containing all projections functions. Every clone has a *dual* clone whose members are the dual functions of the members of the clone. Since the roles of 0 and 1 are interchangeable, properties of clones can be transferred to their dual clones.

The clones form a lattice under set inclusion, which has become known as *Post's lattice* [17], since Post was the first to give a complete description of all clones and of the inclusions between them. Post's lattice is depicted in Fig. 1; note that we use the notation of clones developed in [3,4].[4] The *infinite part* of Post's lattice consists of the clones S_0^n, S_{00}^n, S_{01}^n, S_{02}^n, $n \geqslant 1$, and their duals S_1^n, S_{10}^n, S_{11}^n, S_{12}^n, $n \geqslant 1$. The remaining clones form the *finite part* of Post's lattice.

A *basis* for a clone *Cl* is a subset F of *Cl* such that every function in *Cl* can be obtained from members of F and from the projection functions via compositions. One of the main findings of Post [17] was that every (Boolean) clone has a finite basis. The clones in the finite part of Post's lattice have bases in which each function has arity at most 3. In contrast, the bases of the clones in the infinite part have members of arbitrarily large arity; these bases are depicted in Table 1.

A (Boolean) *co-clone* is a set of Boolean relations containing the *equality* relation $EQ = \{00, 11\}$ and closed under finite Cartesian products, projections, and identification of variables. It has been shown that a *Galois connection* holds between clones and co-clones so that each co-clone turns out to be a maximal class of relations *closed under* every function in some clone. More precisely, let R be an m-ary Boolean relation and let f be an n-ary Boolean function. We say that R is *closed under* f, or that f a *polymorphism* of R, if whenever f is applied coordinate-wise to n (not necessarily distinct) m-tuples in R, then the resulting m-tuple is also in R. For instance, a binary relation R is closed under a ternary function f if whenever (a_{11}, a_{12}), (a_{21}, a_{22}), and (a_{31}, a_{32}) are in R, then also $(f(a_{11}, a_{21}, a_{31}), f(a_{12}, a_{22}, a_{32}))$ is in R. We write Pol(R) to denote the set of all polymorphisms of R. If S is a set of Boolean relations, then we write Pol(S) to denote the set of all functions that are polymorphisms of every relation in S. Thus, $\mathrm{Pol}(S) = \bigcap_{R \in S} \mathrm{Pol}(R)$. It is easy to verify that every Pol(S) is a clone.

Conversely, if F is a set of Boolean functions, then we write Inv(F) to denote the set of all relations that are closed under every function in F. It is easy to verify that every Inv(F) is a co-clone. The functions Inv and Pol are inverse to each other on the lattice of clones and the lattice of co-clones; thus, if F is a clone, then $\mathrm{Pol}(\mathrm{Inv}(F)) = F$, while if S is a co-clone, then $\mathrm{Inv}(\mathrm{Pol}(S)) = S$ (for additional information, see [11,15,16,20] and also the more recent survey [4]). In what follows, we will write *ICl* to denote the co-clone corresponding to clone *Cl*, that is, $ICl = \mathrm{Inv}(Cl)$; for example, IE_2 denotes the co-clone corresponding to the clone E_2 in Post's lattice. Note that a relation R belongs to a co-clone *ICl* if and only if R is closed under every member of a basis for *Cl*.

A *basis* for a co-clone *ICl* is a subset B of *ICl* such that every member of *ICl* is the set of models of some $\exists$CNF(B)-formula. In other words, every member of *ICl* can be obtained from members of B using finite Cartesian products, identification of variables, and projections. A list of simple bases of all (Boolean) co-clones was given in [5].

We now introduce a new, stronger notion of a basis for a co-clone.

Definition 1. Let *ICl* be a co-clone in Post's lattice. A subset B of *ICl* is called a *plain basis* for *ICl* if every member of *ICl* is definable by a CNF(B)-formula. In other words, every member of *ICl* can be obtained from members of B using finite Cartesian products and identification of variables (but no projections).

Every plain basis for *ICl* is also a basis for *ICl*; the converse, however, need not be true.

3. The smallest co-clone problem, ∃-InvSat, and plain bases

Our goal is to give a quadratic algorithm for ∃-InvSat on the Boolean domain. This will be achieved by first giving a quadratic algorithm for a different computational problem about Post's lattice, which we introduce next.

If S is a set of Boolean relations, then there is a smallest co-clone M(S) containing S as a subset. This is so because an arbitrary intersection of co-clones is itself a co-clone, which implies that the intersection of all co-clones containing S is the smallest co-clone containing S (as a subset).

[4] The authors are grateful to Steffen Reith who provided them with the figure.

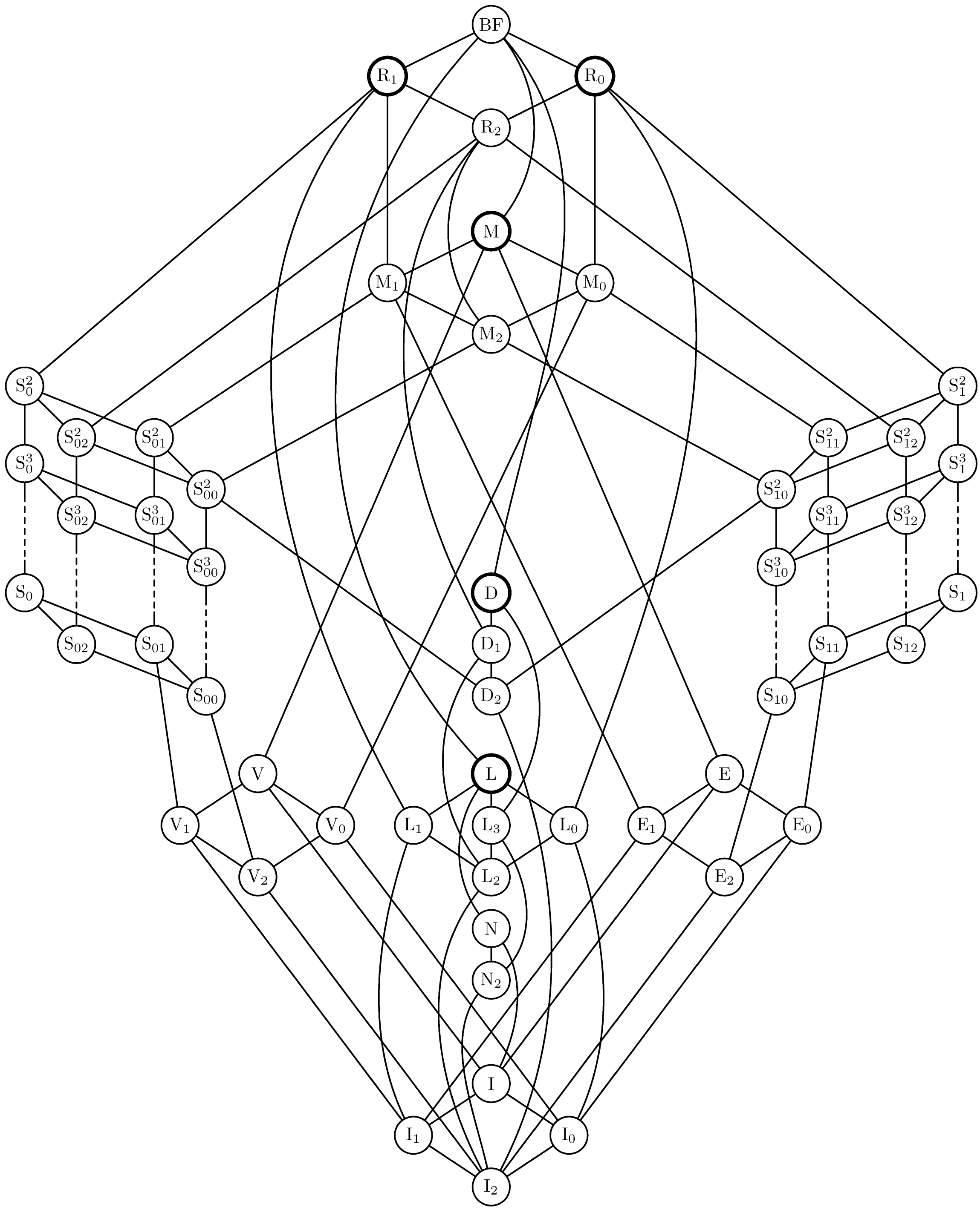

Fig. 1. Lattice of all Boolean clones.

- MIN CO-CLONE is the following algorithmic problem: Given a finite set S of Boolean relations, find the smallest co-clone $\mathrm{M}(S)$ containing S.

The following fact provides a connection between INVSAT and MIN CO-CLONE.

Table 1
Co-clones in the infinite part of Post's lattice and bases for the corresponding clones (where, for instance, $x \vee (y \wedge z)$ denotes the function $(x, y, z) \mapsto x \vee (y \wedge z)$)

Co-clone	Basis for corresponding clone	Co-clone	Basis for corresponding clone
IS_0^n	$\{x \to y,\ \mathrm{dual}(h_n)\}$	IS_1^n	$\{x \wedge \bar{y},\ h_n\}$
IS_0	$\{x \to y\}$	IS_1	$\{x \wedge \bar{y}\}$
IS_{02}^n	$\{x \vee (y \wedge \bar{z}),\ \mathrm{dual}(h_n)\}$	IS_{12}^n	$\{x \wedge (y \vee \bar{z}),\ h_n\}$
IS_{02}	$\{x \vee (y \wedge \bar{z})\}$	IS_{12}	$\{x \wedge (y \vee \bar{z})\}$
IS_{01}^n	$\{\mathrm{dual}(h_n),\ c_1\}$	IS_{11}^n	$\{h_n,\ c_0\}$
IS_{01}	$\{x \vee (y \wedge z),\ c_1\}$	IS_{11}	$\{x \wedge (y \vee z),\ c_0\}$
IS_{00}^n	$\{x \vee (y \wedge z),\ \mathrm{dual}(h_n)\}$	IS_{10}^n	$\{x \wedge (y \vee z),\ h_n\}$
IS_{00}	$\{x \vee (y \wedge z)\}$	IS_{10}	$\{x \wedge (y \vee z)\}$

Fact 1. *Assume that R is a Boolean relation and S is a set of Boolean relations. Then the following statements are equivalent.*

1. *R is the set of models of some $\exists\mathrm{CNF}(S)$-formula.*
2. *R belongs to the smallest co-clone $\mathrm{M}(S)$ containing S.*
3. *The smallest co-clone $\mathrm{M}(\{R\})$ containing R is a subset of the smallest co-clone $\mathrm{M}(S)$ containing S.*

The equivalence (1) $\Longleftrightarrow$ (2) can be derived from the Galois connection between clones and co-clones in Post's lattice [16], while the equivalence (2) $\Longleftrightarrow$ (3) follows easily from the definitions. An immediate consequence of Fact 1 is that ∃-INVSAT has a polynomial-time reduction to MIN CO-CLONE. Indeed, given R and S, we first compute $\mathrm{M}(\{R\})$ and $\mathrm{M}(S)$ using an algorithm for MIN CO-CLONE, and then inspect Post's lattice to determine in constant time whether or not $\mathrm{M}(\{R\}) \subseteq \mathrm{M}(S)$.

Let us, for a moment, ignore the infinite part of Post's lattice and focus only on its finite part. Every clone in the finite part of Post's lattice has a basis with at most 4 elements in which every function has arity at most 3. As mentioned in Section 2.3, a relation belongs to a co-clone *ICl* if and only if it is closed under every member of a basis for *Cl*. Consequently, if the smallest co-clone containing a relation is in the finite part of Post's lattice, then this smallest co-clone can be found in cubic time (in the size of the given relation). This approach, however, cannot be applied to the infinite part of Post's lattice. Indeed, though each basis is finite (and, in fact, contains at most two functions), the arity of one of the two functions in these bases is unbounded (see Table 1); for instance, the basis for S_0^n contains a function of arity n. Thus, testing a relation for closure under the functions in those bases cannot be done in polynomial time using the naive approach.

In the next section, we show how to efficiently solve the MIN CO-CLONE problem on the infinite part of Post's lattice.

3.1. Infinite part of Post's lattice

The bases for the clones in the infinite part of Post's lattice, as well as those for S_0 and S_1, are presented in Table 1 in which, for $n \geqslant 1$, h_n denotes the $(n+1)$-ary function defined by $h_n(x_1, \ldots, x_{n+1}) = \bigvee_{i=1}^{n+1} x_1 \wedge \cdots \wedge x_{i-1} \wedge x_{i+1} \wedge \cdots \wedge x_{n+1}$, c_0 denotes the 0-ary constant function 0, and c_1 denotes the 0-ary constant function 1.

We shall provide plain bases for the corresponding co-clones. Every relation in one of them will turn out to be the set of models of an *implicative hitting set-bounded* (IHSB) formula, which is a restricted Horn or dual Horn formula. By taking advantage of the duality in Post's lattice, we focus our attention on the right side of the infinite part of Post's lattice. Consequently, we define IHSB− formulas, which are restricted Horn formulas; IHSB+ formulas are defined in a dual manner. The collection of IHSB formulas consists of all IHSB− and all IHSB+ formulas.

Definition 2 *(Implicative Hitting Set-Bounded− clauses and formulas).*

- A clause is said to be IHSB− if it is of one of the following types: (x_i), $(\neg x_{i_1} \vee x_{i_2})$, or $(\neg x_{i_1} \vee \cdots \vee \neg x_{i_k})$ for some $k \geqslant 1$.

- For $n \geqslant 2$, an IHSB$-$ clause is said to be *of width n*, denoted by IHSB$-^n$, if it contains at most n literals.
- A CNF formula is said to be IHSB$-$ (respectively, IHSB$-^n$) if all its clauses are IHSB$-$ (respectively, IHSB$-^n$).

The next proposition establishes a link between the IHSB$-^n$ formulas, $n \geqslant 1$, and the co-clones whose corresponding clones are in the right side of the infinite part of Post's lattice.

Proposition 1. *The following statements are true.*

1. *A relation is in the co-clone IS_{10}^n (respectively, in the co-clone IS_{10}) if and only if every prime* CNF *formula representing it is an* IHSB$-^n$ *formula (respectively, an* IHSB$-$ *formula).*
2. *A relation is in the co-clone IS_{11}^n (respectively, in the co-clone IS_{11}) if and only if every prime* CNF *formula representing it is an* IHSB$-^n$ *formula (respectively, an* IHSB$-$ *formula) and contains no clause of the form (x_i).*
3. *A relation is in the co-clone IS_{12}^n (respectively, in the co-clone IS_{12}) if and only if, for every prime* CNF *formula φ representing it, we have that φ is an* IHSB$-^n$ *formula (respectively, an* IHSB$-$ *formula) and for every two variables x_i, x_j, if φ contains the clause $(\neg x_i \vee x_j)$, then it entails the clause $(x_i \vee \neg x_j)$.*
4. *A relation is in the co-clone IS_1^n (respectively, in the co-clone IS_1) if and only if, for every prime* CNF *formula φ representing it, we have that φ is an* IHSB$-^n$ *formula (respectively, an* IHSB$-$ *formula), φ contains no clause of the form (x_i), and for every two variables x_i, x_j, if φ contains the clause $(\neg x_i \vee x_j)$, then it entails the clause $(x_i \vee \neg x_j)$.*

Proof. We give the proof of the first statement only. The other statements can be proved using similar arguments and the statements preceding them.

Böhler *et al.* [5] showed that the relations represented by IHSB$-^n$ clauses form a basis of IS_{10}^n. Hence, if a relation R is represented by some IHSB$-^n$ formula, then R must be in the co-clone IS_{10}^n.

For the other direction, let R be a relation in IS_{10}^n. Since the containment $E_2 \subset S_{10}^n$ holds in Post's lattice, we have that R is a Horn relation, i.e., it is the set of models of some Horn formula. Let φ be a prime CNF formula representing R. Since every prime CNF formula representing a Horn relation must be a Horn formula [21, Proposition 3], each clause of φ must contain either zero or exactly one positive literal. It remains to show that all these clauses are IHSB$-^n$ clauses.

Clauses containing no positive literal. Towards a contradiction, assume that φ contains such a clause which is "too wide," that is to say, a clause C of the form $(\neg x_{i_1} \vee \cdots \vee \neg x_{i_m})$ with $m > n$. Since C is prime, for every j there is a vector m_j in R which falsifies the clause $(\neg x_{i_1} \vee \cdots \vee \widehat{\neg x_{i_j}} \vee \cdots \vee \neg x_{i_m})$, and since all such vectors m_j satisfy C (because φ represents R), we have that there are $n+1$ vectors $m_1, \ldots, m_{n+1} \in R$ whose projections $m_j \upharpoonright \{x_{i_1}, \ldots, x_{i_m}\}$ onto $\{x_{i_1}, \ldots, x_{i_m}\}$ are:

$$\begin{array}{rccccccccccc} & x_{i_1} & x_{i_2} & x_{i_3} & \ldots & x_{i_{n-1}} & x_{i_n} & x_{i_{n+1}} & x_{i_{n+2}} & \ldots & x_{i_m} \\ m_1 \upharpoonright \{x_{i_1}, \ldots, x_{i_m}\} = & 0 & 1 & 1 & \ldots & 1 & 1 & 1 & 1 & \ldots & 1, \\ m_2 \upharpoonright \{x_{i_1}, \ldots, x_{i_m}\} = & 1 & 0 & 1 & \ldots & 1 & 1 & 1 & 1 & \ldots & 1, \\ & & & & \ldots & & & & & & \\ m_n \upharpoonright \{x_{i_1}, \ldots, x_{i_m}\} = & 1 & 1 & 1 & \ldots & 1 & 0 & 1 & 1 & \ldots & 1, \\ m_{n+1} \upharpoonright \{x_{i_1}, \ldots, x_{i_m}\} = & 1 & 1 & 1 & \ldots & 1 & 1 & 0 & 1 & \ldots & 1. \end{array} \tag{1}$$

If we apply the function h_n coordinate-wise to these $n+1$ vectors, we obtain a vector d whose projection on $\{x_{i_1}, \ldots, x_{i_m}\}$ is the vector $1 \ldots 1$; clearly, d is not in R because it falsifies C. It follows that R is not closed under h_n, which contradicts the hypothesis that R is in IS_{10}^n, since h_n is in S_{10}^n (see Table 1).

Clauses containing one positive literal. Again towards a contradiction, assume that φ contains a clause of the form $C = (x_{i_1} \vee \neg x_{i_2} \vee \cdots \vee \neg x_{i_m})$ with $m > 2$. Reasoning as above, we obtain three vectors m_1, m_2, m_3 in R whose projections on $\{x_{i_1}, \ldots, x_{i_m}\}$ are as below:

$$\begin{array}{rcccccc} & x_{i_1} & x_{i_2} & x_{i_3} & x_{i_4} & \ldots & x_{i_m} \\ m_1 \upharpoonright \{x_{i_1}, \ldots, x_{i_m}\} = & 1 & 1 & 1 & 1 & \ldots & 1, \\ m_2 \upharpoonright \{x_{i_1}, \ldots, x_{i_m}\} = & 0 & 0 & 1 & 1 & \ldots & 1, \\ m_3 \upharpoonright \{x_{i_1}, \ldots, x_{i_m}\} = & 0 & 1 & 0 & 1 & \ldots & 1. \end{array}$$

If we apply the function $(x, y, z) \mapsto x \wedge (y \vee z)$ coordinate-wise to these three vectors, we obtain a vector d whose projection on $\{x_{i_1}, \dots, x_{i_m}\}$ is the vector $011\dots1$; clearly, d is not in R because it falsifies C, yielding a contradiction again since the function $(x, y, z) \mapsto x \wedge (y \vee z)$ is in S^n_{10}.

Since we have shown that every clause in φ consists of at most n negative literals or has at most 2 literals, we conclude that φ is an IHSB$-^n$ formula. □

As an immediate consequence of Proposition 1, we obtain the following plain bases for co-clones corresponding to clones in the right side of the infinite part of Post's lattice.

Corollary 1. *The following statements are true.*

1. *The set* $\{(x), (\neg x \vee y), (\neg x_1 \vee \dots \vee \neg x_k)\colon k \leqslant n\}$ *of all* IHSB$-^n$ *clauses is a plain basis for the co-clone* IS^n_{10}.
2. *The set* $\{(\neg x \vee y), (\neg x_1 \vee \dots \vee \neg x_k)\colon k \leqslant n\}$ *is a plain basis for the co-clone* IS^n_{11}.
3. *The set* $\{(x), Eq, (\neg x_1 \vee \dots \vee \neg x_k)\colon k \leqslant n\}$ *is a plain basis for the co-clone* IS^n_{11}. *Here, Eq is the equality relation* $\{00, 11\}$ *(i.e., the relation represented by the formula* $(x \leftrightarrow y)$*).*
4. *The set* $\{Eq, (\neg x_1 \vee \dots \vee \neg x_k)\colon k \leqslant n\}$ *is a plain basis for the co-clone* IS^n_1.

By duality, results analogous to Proposition 1 and Corollary 1 can be obtained for co-clones corresponding to clones in the left side of the infinite part of Post's lattice; we leave it to the reader to formulate these results.

We can now give a quadratic algorithm for the MIN CO-CLONE problem on the infinite part of Post's lattice.

Proposition 2. *Given a relation* R *in* IS_{10}, *the smallest co-clone in* $\{IS^n_1, IS^n_{10}, IS^n_{12}, IS^n_{11} \mid n \geqslant 1\}$ *containing* R *can be found in time* $O(k^2m^2)$, *where* k *is the arity of* R *and* m *is the number of elements of* R. *A dual result holds for a relation* R *in the co-clone* IS_{00}.

Proof. Zanuttini and Hébrard [21] showed that, given a Boolean relation R, a prime CNF formula φ representing R can be computed in time $O(k^2m^2)$, and that φ contains $O(km)$ clauses. By scanning φ once, one can find the maximum size n of its clauses in time $O(k^2m)$, and also decide whether φ contains unary positive clauses. Finally, for every clause of the form $(\neg x \vee y)$ in φ one can decide whether φ entails $(x \vee \neg y)$ in time $O(mn)$ by testing whether every vector in R satisfies the clause $(x \vee \neg y)$; since φ contains $O(km)$ clauses, this requires $O(k^2m^2)$ operations. Once this information is collected, one can find the smallest co-clone in $\{IS^n_1, IS^n_{10}, IS^n_{12}, IS^n_{11} \mid n \geqslant 1\}$ containing R immediately by referring to Proposition 1. □

Proposition 2 and the remarks preceding Section 3.1 yield the following result.

Corollary 2. *The* MIN CO-CLONE *problem and the* ∃-INVSAT *problem can be solved in cubic time.*

By Proposition 2, the algorithms for MIN CO-CLONE and ∃-INVSAT take quadratic time on the infinite part of Post's lattice; however, they take cubic time on the finite part of Post's lattice, since there we have to test that a Boolean relation is closed under all functions in the bases of the corresponding clone, and the maximum arity of these functions can be 3. In the next section, we shall give a quadratic algorithm for MIN CO-CLONE and ∃-INVSAT by first obtaining plain bases for every Boolean co-clone and then reasoning as in the proof of Proposition 2.

3.2. Plain bases for co-clones and quadratic algorithms for MIN CO-CLONE *and* ∃-INVSAT

Table 2 gives a plain basis for every Boolean co-clone in Post's lattice. In this table, whenever possible, we denote relations by clauses that represent them; for example, the clause $(\neg x \vee y)$ denotes the binary relation $\{00, 01, 11\}$. The positive clause $(x_1 \vee \dots \vee x_k)$ of width k is denoted by P_k; similarly, the negative clause $(\neg x_1 \vee \dots \vee \neg x_k)$ of width k is denoted by N_k. We use a similar kind of notation for relations that are represented by linear equations; we write Eq to denote the binary equality relation $\{00, 11\}$. Finally, $Compl_{k,\ell}$ denotes the $(k+\ell)$-ary relation represented by the conjunction of clauses $(x_1 \vee \dots \vee x_k \vee \neg y_1 \vee \dots \vee \neg y_\ell) \wedge (\neg x_1 \vee \dots \vee \neg x_k \vee y_1 \vee \dots \vee y_\ell)$, i.e., the complementive

Table 2
Plain bases for all co-clones

Co-clone	Plain basis	Property
IBF	$\{Eq\}$	only equalities
IR_0	$\{Eq, (\neg x)\}$	neg^1
IR_1	$\{Eq, (x)\}$	pos^1
IR_2	$\{Eq, (\neg x), (x)\}$	unary
IM	$\{(\neg x \vee y)\}$	implicative
IM_0	$\{(\neg x), (\neg x \vee y)\}$	implicative or pos^1
IM_1	$\{(x), (\neg x \vee y)\}$	implicative or neg^1
IM_2	$\{(x), (\neg x), (\neg x \vee y)\}$	implicative or unary
IS_0^n	$\{Eq\} \cup \{P_k \mid k \leqslant n\}$	pos^n
IS_0	$\{Eq\} \cup \{P_k \mid k \in \mathbb{N}\}$	pos.
IS_1^n	$\{Eq\} \cup \{N_k \mid k \leqslant n\}$	neg^n
IS_1	$\{Eq\} \cup \{N_k \mid k \in \mathbb{N}\}$	neg.
IS_{02}^n	$\{Eq, (\neg x)\} \cup \{P_k \mid k \leqslant n\}$	neg^1 or pos^n
IS_{02}	$\{Eq, (\neg x)\} \cup \{P_k \mid k \in \mathbb{N}\}$	neg^1 or positive
IS_{12}^n	$\{Eq, (x)\} \cup \{N_k \mid k \leqslant n\}$	pos^1 or neg^n
IS_{12}	$\{Eq, (x)\} \cup \{N_k \mid k \in \mathbb{N}\}$	pos^1 or negative
IS_{01}^n	$\{(\neg x \vee y)\} \cup \{P_k \mid k \leqslant n\}$	implicative or pos^n
IS_{01}	$\{(\neg x \vee y)\} \cup \{P_k \mid k \in \mathbb{N}\}$	implicative or positive
IS_{11}^n	$\{(\neg x \vee y)\} \cup \{N_k \mid k \leqslant n\}$	implicative or neg^n
IS_{11}	$\{(\neg x \vee y)\} \cup \{N_k \mid k \in \mathbb{N}\}$	implicative or negative
IS_{00}^n	$\{(\neg x), (\neg x \vee y)\} \cup \{P_k \mid k \leqslant n\}$	$\text{IHSB}+^n$
IS_{00}	$\{(\neg x), (\neg x \vee y)\} \cup \{P_k \mid k \in \mathbb{N}\}$	IHSB+
IS_{10}^n	$\{(x), (\neg x \vee y)\} \cup \{N_k \mid k \leqslant n\}$	$\text{IHSB}-^n$
IS_{10}	$\{(x), (\neg x \vee y)\} \cup \{N_k \mid k \in \mathbb{N}\}$	IHSB−
ID	$\{(x \oplus y = c) \mid c \in \{0,1\}\}$	affine of width exactly 2
ID_1	$\{(x = c) \mid c \in \{0,1\}\} \cup \{(x \oplus y = c) \mid c \in \{0,1\}\}$	affine of width 2
ID_2	$\{(x), (\neg x), (x \vee y), (\neg x \vee y), (\neg x \vee \neg y)\}$	bijunctive
IL	$\{(x_1 \oplus \cdots \oplus x_k = 0) \mid k \text{ even}\}$	even homogeneous linear equation
IL_0	$\{(x_1 \oplus \cdots \oplus x_k = 0) \mid k \in \mathbb{N}\}$	homogeneous linear equation
IL_1	$\{(x_1 \oplus \cdots \oplus x_k = c) \mid k \in \mathbb{N}, c = k \bmod 2\}$	1-valid linear equation
IL_2	$\{(x_1 \oplus \cdots \oplus x_k = c) \mid k \in \mathbb{N}, c \in \{0,1\}\}$	linear equation
IL_3	$\{(x_1 \oplus \cdots \oplus x_k = c) \mid k \text{ even}, c \in \{0,1\}\}$	even linear equation
IV	$\{(x_1 \vee \cdots \vee x_k \vee \neg y) \mid k \geqslant 1\}$	definite dual Horn and not neg^1
IV_0	$\{(x_1 \vee \cdots \vee x_k \vee \neg y) \mid k \in \mathbb{N}\}$	definite dual Horn
IV_1	$\{P_k \mid k \in \mathbb{N}\} \cup \{(x_1 \vee \cdots \vee x_k \vee \neg y) \mid k \geqslant 1\}$	dual Horn and not neg^1
IV_2	$\{P_k \mid k \in \mathbb{N}\} \cup \{(x_1 \vee \cdots \vee x_k \vee \neg y) \mid k \in \mathbb{N}\}$	dual Horn
IE	$\{(\neg x_1 \vee \cdots \vee \neg x_k \vee y) \mid k \geqslant 1\}$	definite Horn and not pos^1
IE_0	$\{N_k \mid k \in \mathbb{N}\} \cup \{(\neg x_1 \vee \cdots \vee \neg x_k \vee y) \mid k \geqslant 1\}$	Horn and not pos^1
IE_1	$\{(\neg x_1 \vee \cdots \vee \neg x_k \vee y) \mid k \in \mathbb{N}\}$	definite Horn
IE_2	$\{N_k \mid k \in \mathbb{N}\} \cup \{(\neg x_1 \vee \cdots \vee \neg x_k \vee y) \mid k \in \mathbb{N}\}$	Horn
IN	$\{Compl_{k,\ell} \mid k, \ell \geqslant 1\}$	complementive, 0-valid and 1-valid
IN_2	$\{Compl_{k,\ell} \mid k, \ell \in \mathbb{N}\}$	complementive
II	$\{(x_1 \vee \cdots \vee x_k \vee \neg y_1 \vee \cdots \vee \neg y_\ell) \mid k, \ell \geqslant 1\}$	0-valid and 1-valid
II_0	$\{(x_1 \vee \cdots \vee x_k \vee \neg y_1 \vee \cdots \vee \neg y_\ell) \mid k \in \mathbb{N}, \ell \geqslant 1\}$	0-valid
II_1	$\{(x_1 \vee \cdots \vee x_k \vee \neg y_1 \vee \cdots \vee \neg y_\ell) \mid k \geqslant 1, \ell \in \mathbb{N}\}$	1-valid
II_2	$\{(x_1 \vee \cdots \vee x_k \vee \neg y_1 \vee \cdots \vee \neg y_\ell) \mid k, \ell \in \mathbb{N}\}$	any clause

In this table: (i) neg^n means negative and containing at most n literals, and dually for pos^n; (ii) *definite Horn* means Horn with exactly one positive literal, and dually for *definite dual Horn*.

relation $\{0,1\}^{k+\ell} \setminus \{0 \ldots 01 \ldots 1, 1 \ldots 10 \ldots 0\}$. The last column gives the usual name given to the property satisfied by each clause, equation or relation in the basis.

The next proposition asserts that Table 2 is correct.

Proposition 3. *Each line in Table* 2 *gives a plain basis for the corresponding co-clone.*

Proof. The correctness of the list for co-clones in the infinite part of Post's lattice follows from Proposition 1 and Corollary 1. For the remaining co-clones, we proceed from the largest co-clone to the smallest one. Since the proofs for co-clones of the form ICl, ICl_0, ICl_1 follow from the proofs for co-clone ICl_2 in a straightforward manner, we only consider the latter in many cases.

- $[II, II_c]$ Obviously, every relation can be represented by a CNF formula.
- $[IN, IN_2]$ Obviously, every complementive relation can be represented by a CNF containing the clause $(\neg x_1 \vee \cdots \vee \neg x_k \vee y_1 \vee \cdots \vee y_\ell)$ as soon as it contains $(x_1 \vee \cdots \vee x_k \vee \neg y_1 \vee \cdots \vee \neg y_\ell)$; grouping these clauses two by two in the CNF formula yields a conjunction of $Compl_{k,\ell}$ relations; conversely the set of models of such a formula is complementive.
- $[IE, IE_c, IV, IV_c, IL, IL_c, ID, ID_c]$ For the co-clones IE_2, IV_2, IL_2, ID_2, the correctness of the plain bases follows from results in [10,18], and [21].
- $[IM, IM_c]$ The proof for IM_2 follows from the inclusions $IM_2 \subseteq IS_{00}^2, IS_{10}^2$ in one direction, and from the closure of the clauses in the plain basis under *and* and *or* in the other direction.
- $[IR_c, IBF]$ In one direction, the result for IR_2 follows from the inclusions $IR_2 \subseteq IM_2, ID_1$, because clauses $(\neg x \vee y)$ of the plain basis of IM_2 are not in ID_1, while unary clauses are (equation $(x = 1)$ is equivalent to clause (x), and equation $(x = 0)$ is equivalent to clause $(\neg x)$). In the other direction, the proof follows from the closure of unary clauses under both *or* and $(x, y, z) \mapsto x \wedge (y \oplus z \oplus 1)$. □

Remark 1. When considering bases for mathematical objects, an important question is that of minimality. For instance, Böhler *et al.* gave bases for all Boolean co-clones and showed that their bases are of minimal order, where the *order* of a set of Boolean relations is the maximum arity of the relations in the set. As listed in Table 2, our plain bases are *minimal* in the sense that they are included in every other plain basis for the same co-clone, provided replicated variables in the scope of an atom in CNF(S)-formulas are disallowed (see [7] for more details).

We are now ready to derive the main result of this paper.

Theorem 1. *Given a Boolean relation R, the minimal co-clone $M(\{R\})$ containing R can be found in time $O(k^2m^2)$, where k is the arity of R and m is the number of elements of R. Consequently, the* MIN CO-CLONE *problem and the* ∃-INVSAT *problem can be solved in quadratic time.*

Proof. Using the results in [21] and the list of plain bases in Table 2, we design an algorithm that extends the quadratic algorithm given in the proof of Proposition 2. Specifically, given R, first compute a prime CNF formula φ representing R in time $O(k^2m^2)$ using the algorithm in [21]; the formula φ contains $O(km)$ clauses. By the results in [21], our Proposition 1 (and a similar reasoning for other co-clones), for every co-clone ICl whose plain basis consists entirely of clauses, we know that φ is over this plain basis if and only if R is in ICl. This can be decided in time linear in the size of φ; actually, in time $O(k^2m)$. Thus we are left with co-clones whose plain bases contain relations that are not equivalent to an individual clause.

For plain bases containing the relation Eq, it is easily seen that it is enough to decide whether R entails $(\neg x_i \vee x_j)$ as soon as $(x_i \vee \neg x_j)$ is in φ. In the affirmative, $(x_i \vee \neg x_j)$ can be replaced with $Eq(x_i, x_j)$; otherwise, R is not in the co-clone. Once again, this requires $O(k^2m)$ operations. As shown in [21], the affine co-clones can be handled in a similar manner, by essentially replacing $\vee$ with $\oplus$ in φ and by testing whether each vector in R satisfies the resulting affine formula, a task that takes time $O(k^2m^2)$. The reasoning for complementive co-clones is similar.

This process makes it possible to decide membership of a relation R in each co-clone in quadratic time. The smallest co-clone $M(\{R\})$ containing R can then be computed in constant time using Post's lattice. The MIN CO-CLONE problem can be solved in quadratic time as follows: given a finite set S of Boolean relation, first compute the smallest co-clone containing each member of S and then use Post's lattice to compute the union of these smallest co-clones. Finally, by Fact 1, the ∃-INVSAT problem can be solved in quadratic time using the quadratic algorithm for the MIN CO-CLONE problem. □

We conclude the paper with several remarks.

- Although the MIN CO-CLONE problem was used here as a stepping stone to solve the ∃-INVSAT problem, it is of independent interest. In particular, the quadratic algorithm for MIN CO-CLONE implies a quadratic algorithm for the so-called *meta-problem* (see [6]) associated with the classification of the complexity of a family of decision problems $\Gamma(S)$, where S is a finite set of Boolean relations, provided this classification follows the lines of Post's lattice.
  As an illustration, consider Schaefer's Dichotomy Theorem [18] which, as described in detail in Section 2.2, asserts that, for every finite set of Boolean relations, either SAT(S) is in P or SAT(S) is NP-complete; moreover, the tractable cases of SAT(S) are the cases in which S is Horn, or S is dual Horn, or S is bijunctive, or every S is affine. Thus, the quadratic algorithm for MIN CO-CLONE implies that, given a finite set S of Boolean relations, we can decide in quadratic time whether or not SAT(S) is in P. A similar result holds for the meta-problem associated with the INVSAT problem studied in [13]. Earlier known algorithms for these meta-problems were cubic, as they relied on closure properties.
- In the vein of the previous remark, we note that an important, but not well understood, issue is what makes the classification of the complexity of a family of problems follow Post's lattice. Indeed, assume a family of decision problems $\Gamma(S)$, where S is a finite set of relations, is such that the property $S' \subseteq M(S)$ (where S' is a finite set of relations) does not *a priori* guarantee that $\Gamma(S')$ is polynomial-time reducible to $\Gamma(S)$. Then a complexity classification for this family cannot *a priori* be obtained by using Post's lattice.
  However, assume that whenever every relation in S' can be expressed from the relations in S using only finite Cartesian products and identification of variables, then $\Gamma(S')$ is polynomial-time reducible to $\Gamma(S)$ (which is true of many decision problems about formulas). Then Schnoor and Schnoor [19] show that a complexity classification for the family $\Gamma(S)$ can be obtained by Post's lattice, provided that for every finite set of relations S and for every finite subset B of a plain basis for $M(S)$, $\Gamma(B)$ is polynomial-time reducible to $\Gamma(S^{[\mathrm{ext}]})$, where $S^{[\mathrm{ext}]}$ is a particular relation which they define. Consequently, our notion of a plain bases complements Schnoor and Schnoor's work as a step towards a deeper understanding of complexity classifications. For more details we refer the reader to [19].
- The ∃-INVSAT problem has a dual version, which asks: given a Boolean function f and a finite set of Boolean functions F, does f belong to the clone generated by F. This problem was shown to be solvable in polynomial time by Bergman and Slutzki [2]; in fact, it was shown to be in NL. It is not clear, however, that this result can be used to derive a polynomial-time algorithm for ∃-INVSAT. The main reason is that, in the problem studied by Bergman and Slutzki, the set F of functions is given as an input. Thus, if we wanted to take advantage of their result, then we would have to compute a basis for the clone corresponding to the smallest co-clone containing a given set of Boolean relations, which is exactly the difficult part in the ∃-INVSAT problem.
  Along these lines, note also that, by definition, the ∃-INVSAT problem could also be reformulated as the problem of deciding whether a given set of Boolean relations S is a basis for $\{R\}$ (in the standard sense of a basis, as studied by Böhler *et al.*). It appears, however, that this standard notion of a basis is of no help in solving ∃-INVSAT efficiently, whereas the stronger notion of a plain basis gives rise to a quadratic algorithm for ∃-INVSAT
- Finally, all results presented here are special to the Boolean domain, as they depend heavily on Post's lattice. The ∃-INVSAT problem is a perfectly meaningful, and interesting, structure identification problem over higher domains. As mentioned earlier, Dalmau [8] pointed out that, for every finite domain, ∃-INVSAT is a decidable problem. Its exact complexity, however, is not known on any domain with more than two elements.

References

[1] S. Abiteboul, R. Hull, V. Vianu, Foundations of Databases, Addison–Wesley, 1995.

[2] C. Bergman, G. Slutzki, Complexity of some problems concerning varieties and quasi-varieties of algebras, SIAM J. Comput. 30 (2) (2000) 359–382.

[3] E. Böhler, N. Creignou, S. Reith, H. Vollmer, Playing with Boolean blocks, Part I: Post's lattice with applications to complexity theory, ACM–SIGACT News 34 (4) (2003) 38–52, Complexity Theory Column 42.

[4] E. Böhler, N. Creignou, S. Reith, H. Vollmer, Playing with Boolean blocks, Part II: Constraint satisfaction problems, ACM–SIGACT News 35 (1) (2004) 22–35, Complexity Theory Column 43.

[5] E. Böhler, S. Reith, H. Schnoor, H. Vollmer, Bases for Boolean co-clones, Inform. Process. Lett. 96 (2005) 59–66.

[6] N. Creignou, S. Khanna, M. Sudan, Complexity Classifications of Boolean Constraint Satisfaction Problems, SIAM Monogr. Discrete Math. Appl., 2001.

[7] N. Creignou, Ph.G. Kolaitis, B. Zanuttini, Preferred representations of Boolean relations, Electronic Colloquium on Computational Complexity (ECCC), Technical report TR05-119, 2005.
[8] V. Dalmau, Computational complexity of problems over generalized formulas, PhD thesis, Department de Llenguatges i Sistemes Informàtica, Universitat Politécnica de Catalunya, 2000.
[9] R. Dechter, Constraint Processing, Morgan Kaufmann, 2003.
[10] R. Dechter, J. Pearl, Structure identification in relational data, Artificial Intelligence 58 (1992) 237–270.
[11] D. Geiger, Closed systems of functions and predicates, Pacific J. Math. 27 (2) (1968) 228–250.
[12] H. Garcia-Molina, J.D. Ullman, J. Widom, Database Systems: The Complete Book, Prentice Hall, 2002.
[13] D. Kavvadias, M. Sideri, The inverse satisfiability problem, SIAM J. Comput. 28 (1) (1998) 152–163.
[14] R. Ladner, On the structure of polynomial time reducibility, J. ACM 22 (1) (1975) 155–171.
[15] N. Pippenger, Theories of Computability, Cambridge University Press, Cambridge, 1997.
[16] R. Pöschel, L.A. Kalužnin, Funktionen- und Relationenalgebren, DVW, Berlin, 1979.
[17] E.L. Post, The two-valued iterative systems of mathematical logic, in: Ann. of Math. Stud., vol. 5, 1941, pp. 1–122.
[18] T. Schaefer, The complexity of satisfiability problems, in: Proc. 10th STOC, San Diego, CA, USA, Association for Computing Machinery, 1978, pp. 216–226.
[19] H. Schnoor, I. Schnoor, Partial polymorphisms and constraint satisfaction problems, in: N. Creignou, Ph.G. Kolaitis, H. Vollmer (Eds.), Complexity of Constraints, Springer-Verlag, Berlin, in press.
[20] A. Szendrei, Clones in Universal Algebra, Séminaire de Mathématiques Supérieures, vol. 99, Les Presses de l'Université de Montréal, Montréal, 1986.
[21] B. Zanuttini, J.-J. Hébrard, A unified framework for structure identification, Inform. Process. Lett. 81 (6) (2002) 335–339.