Discrete Applied Mathematics 160 (2012) 365–382

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Boolean functions with a simple certificate for CNF complexity

Ondřej Čepek [a,b,*], Petr Kučera [a,1], Petr Savický [c]

[a] Department of Theoretical Computer Science and Mathematical Logic, Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, 118 00 Praha 1, Czech Republic

[b] Institute of Finance and Administration, Estonska, 500 101 00 Praha 10, Czech Republic

[c] Institute of Computer Science, Academy of Sciences of the Czech Republic, Pod Vodarenskou vezi, 271/2 182 07 Prague 8, Czech Republic

ARTICLE INFO

Article history:
Received 15 January 2010
Received in revised form 13 May 2011
Accepted 31 May 2011
Available online 13 July 2011

Keywords:
Boolean functions
CNF representations

ABSTRACT

In this paper we study relationships between CNF representations of a given Boolean function f and essential sets of implicates of f. It is known that every CNF representation and every essential set must intersect. Therefore the maximum number of pairwise disjoint essential sets of f provides a lower bound on the size of any CNF representation of f. We are interested in functions, for which this lower bound is tight, and call such functions coverable. We prove that for every coverable function there exists a polynomially verifiable certificate (witness) for its minimum CNF size. On the other hand, we show that not all functions are coverable, and construct examples of non-coverable functions. Moreover, we prove that computing the lower bound, i.e. the maximum number of pairwise disjoint essential sets, is *NP*-hard under various restrictions on the function and on its input representation.

1. Introduction

The Boolean minimization (BM) problem can be stated as follows: given a CNF ϕ find a CNF ϕ' representing the same function and such that ϕ' consists of a minimum possible number of clauses. A decision version of the problem is obtained by including a bound in the instance and the question is, whether there is a representation ϕ' of at most the given size. BM has many practical applications. For instance, in artificial intelligence this problem is equivalent to finding the most compact representation of a given knowledge base [11,12]. Such transformation of a knowledge base accomplishes knowledge compression, since the actual knowledge does not change, while the size of the representation can be significantly reduced.

BM is in general a hard problem. Obviously, it contains the satisfiability problem (SAT) as its special case. An unsatisfiable CNF is identically zero, which means that its shortest representation consists only of a constant. In fact, BM was shown to be probably harder than SAT: while SAT is *NP*-complete (i.e. Σ_1^p-complete) [6], the decision version of BM is Σ_2^p-complete [20]. BM remains *NP*-hard even for some classes of Boolean functions for which SAT is solvable in polynomial time. The best known example of such a class are Horn functions (see [2,11,15] for various BM intractability results for the class of Horn functions). The difficulty of BM of course raises a natural question whether for a given input CNF, a nontrivial lower bound can be obtained for the number of clauses in the shortest equivalent CNF. This question was recently addressed in [3] where the concept of essential sets of function f was introduced.

* Corresponding author at: Department of Theoretical Computer Science and Mathematical Logic, Faculty of Mathematics and Physics, Charles University, Malostranské náměstí 25, 118 00 Praha 1, Czech Republic. Tel.: +420 221 914 246; fax: +420 221 914 323.

E-mail addresses: ondrej.cepek@mff.cuni.cz, cepek@rutcor.rutgers.edu (O. Čepek), kucerap@ktiml.mff.cuni.cz (P. Kučera), savicky@cs.cas.cz (P. Savický).

[1] Tel.: +420 221 914 138; fax: +420 221 914 323.

0166-218X/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.dam.2011.05.013

Similarly as in [3], the main object of interest throughout this paper will be the set $\mathcal{I}(f)$ defined as the resolution closure of the set $\mathcal{I}^p(f)$ of all prime implicates of f. A subset $\mathcal{E} \subseteq \mathcal{I}(f)$ is an essential set of f, if $\mathcal{I}(f) \setminus \mathcal{E}$ is closed under resolution. It was shown in [3] that given a Boolean function f, every CNF representation of f must intersect every nonempty essential set. Therefore, the maximum number of pairwise disjoint essential sets constitutes a lower bound on the size of any CNF which represents f.

In this paper we are primarily interested in functions for which the above described lower bound is tight. We shall call such functions coverable. It should be noted that nontrivial subclasses of Boolean functions which consist of coverable functions are already known. These include acyclic and quasi-acyclic Horn functions [3] as well as the class of CQ-Horn functions [5].

After introducing the necessary notation and presenting the basic results from [3] related to essential sets in Section 2, we show in Section 3 that for every coverable function f there exists a polynomially verifiable certificate (witness) for the size of its minimum CNF representation, i.e. a certificate sufficient for a polynomial time verification that no CNF representation of f has fewer clauses than the given minimum one. In Section 4 we study tractable classes of CNFs, and prove that if a tractable class is coverable (i.e. all CNFs in the class represent coverable functions) then the decision version of BM for this class is both in *NP* and *coNP* and derive several consequences of this fact.

Given a CNF which represents a function f, it may be difficult to compute the lower bound (i.e. the maximum number of pairwise disjoint essential sets) simply because the set $\mathcal{I}(f)$ is too large. Therefore we define in Section 5 projections of essential sets on the set $\mathcal{I}^p(f)$ of prime implicates, and show that the lower bound on the size of f can be characterized using these projections only. This allows us to work with smaller sets of implicates and thus prove or disprove the tightness of the lower bound for particular input CNFs more efficiently. Moreover, it is shown in Section 5 that several properties of essential sets carry over to the studied projections. Using the results of Section 5 we construct in Section 6 an example of a function where the lower bound is not tight, and moreover we show that the gap between the lower bound and the size of the minimal CNF can be made arbitrarily large.

In Section 7 we prove that given a CNF which represents a function f, computing the maximum number of pairwise disjoint essential sets of f is *NP*-hard, even if the input is restricted to cubic pure Horn CNFs. Finally, in Section 8, we show that in the unrestricted case, computing the maximum number of pairwise disjoint essential sets is *NP*-hard, when the function is given by its truth table instead of a CNF. On the other hand, given a truth table representation, a relaxation of the lower bound based on linear programming is shown to be obtainable in polynomial time.

2. Basic notation, definitions, and results

In this section we introduce the necessary notation and summarize the basic known results that will be needed later in the text.

2.1. Boolean functions

A *Boolean function* f on n propositional variables $x_1, \ldots, x_n$ is a mapping $\{0, 1\}^n \to \{0, 1\}$. The propositional variables $x_1, \ldots, x_n$ and their negations $\bar{x}_1, \ldots, \bar{x}_n$ are called *literals* (*positive* and *negative literals*, respectively). An elementary disjunction of literals is called a *clause*, if every propositional variable appears in it at most once. A clause C is called an *implicate* of a function f if for every $x \in \{0, 1\}^n$ we have $f(x) \leq C(x)$. An implicate C is called *prime* if dropping any literal from it produces a clause which is not an implicate.

It is a well-known fact that every Boolean function f can be represented by a conjunction of clauses (see e.g. [9]). Such an expression is called a *conjunctive normal form* (or CNF) of the Boolean function f. In the rest of the paper we shall often identify a CNF ϕ with a set of its clauses and we shall use both notions interchangeably. A CNF ϕ representing a function f is called *prime* if each clause of ϕ is a prime implicate of the function f. A CNF ϕ representing a function f is called *irredundant* if dropping any clause from ϕ produces a CNF that does not represent f.

Two clauses C_1 and C_2 are said to be *resolvable* if they contain exactly one complementary pair of literals. That means that we can write $C_1 = \tilde{C}_1 \vee x$ and $C_2 = \tilde{C}_2 \vee \bar{x}$ for some propositional variable x and clauses $\tilde{C}_1$ and $\tilde{C}_2$ which contain no complementary pair of literals. The clauses C_1 and C_2 are called *parent clauses* and the disjunction $R(C_1, C_2) = \tilde{C}_1 \vee \tilde{C}_2$ is called the *resolvent* of the parent clauses C_1 and C_2. Note that the resolvent is a clause (does not contain a propositional variable and its negation). We say that a clause C can be *derived by a series of resolutions* from a CNF ϕ, if there exists a finite sequence $C_1, C_2, \ldots, C_p$ of clauses such that

(1) $C_p = C$, and
(2) for $i = 1, \ldots, p$, either C_i is a clause in ϕ or there exist $j < i$ and $k < i$ such that $C_i = R(C_j, C_k)$.

It is a well-known fact, see for example [4], that a resolvent of two implicates of f is an implicate of f and every prime implicate of f can be derived by a series of resolutions from any CNF representing f. The so-called *Quine's procedure* [4,17,18] takes a CNF φ as an input and outputs the list of all prime implicates of the function represented by φ. Given a set of clauses $\mathcal{C}$, we shall denote by $\mathcal{R}(\mathcal{C})$ the resolution closure of $\mathcal{C}$, i.e. $\mathcal{R}(\mathcal{C})$ is the set of all clauses, which can be derived by a series of resolutions from clauses in $\mathcal{C}$.

For a Boolean function f let us denote by $\mathcal{I}^p(f)$ the set of its prime implicates, and let $\mathcal{I}(f)$ denote the resolution closure of the set of its prime implicates $\mathcal{I}(f) = \mathcal{R}(\mathcal{I}^p(f))$. A clause C is called *negative* if it contains no positive literals. It is called *pure Horn* if it contains exactly one positive literal. To simplify notation, we shall sometimes write a pure Horn clause $C = \bigvee_{x\in S} \bar{x} \vee y$ simply as $C = S \vee y$. Each propositional variable $x \in S$ is called a *subgoal of* C and the propositional variable y is called the *head of* C.

A CNF is called *Horn*, if it contains only negative and pure Horn clauses. A CNF is called *pure Horn*, if it contains only pure Horn clauses. Finally, a Boolean function is called *Horn*, if it has at least one representation by a Horn CNF, and similarly a Boolean function is called *pure Horn*, if it has at least one representation by a pure Horn CNF.

It is known (see [10]) that each prime implicate of a Horn function is either negative or pure Horn, and each prime implicate of a pure Horn function is pure Horn. Thus, in particular, any prime CNF representing a Horn function is Horn, and any prime CNF representing a pure Horn function is pure Horn.

Definition 2.1. A class of CNFs $\mathcal{X}$ will be called *tractable*, if it satisfies the following properties.

- Recognition: Given an arbitrary CNF φ it is possible to decide in polynomial time with respect to the size of φ whether $\varphi \in \mathcal{X}$.
- Satisfiability: Given an arbitrary CNF $\varphi \in \mathcal{X}$ it is possible to decide in polynomial time with respect to the size of φ whether φ is satisfiable.
- Partial assignment: Given an arbitrary CNF $\varphi \in \mathcal{X}$, if ψ is produced from φ by fixing some variables to 0 or 1 and substituting these values into φ, then $\psi \in \mathcal{X}$.
- Prime representations: Given an arbitrary CNF $\varphi \in \mathcal{X}$, if φ represents a function f then all prime CNF representations of f belong to $\mathcal{X}$.

It follows that given a CNF φ from a tractable class, we can decide in polynomial time whether a given clause C is an implicate of φ by substituting the appropriate values (which make C zero) into φ and testing the satisfiability of the resulting formula. This property of tractable classes has two important consequences.

Lemma 2.2. *Let φ be a CNF from a tractable class. Then it is possible to find in polynomial time a prime and irredundant CNF ψ which is equivalent to φ.*

Proof. For every clause C in φ we can delete its literals one by one and test whether the remaining clause is still an implicate of φ. If yes, the literal is deleted permanently, if no, the literal is returned back into the clause. When no literal can be deleted, we have arrived to a prime subclause of C which can replace C in φ. Note that for different orders of literal deletions we may arrive to different prime subclauses of C. After getting a prime CNF we can test for each clause whether it is an implicate of the CNF defined by the remaining clauses. If yes, the clause is redundant and can be deleted, if no, the clause is kept in place. In this way an irredundant CNF is produced. Note again that for different orders of clause deletions we may arrive to different prime and irredundant CNFs (all representing the same function as φ). Since the described procedure amounts to a linear number of tests whether a given clause is an implicate of a given CNF from a tractable class, it follows that the procedure runs in polynomial time with respect to the length of the input CNF. □

Lemma 2.3. *Let φ and ψ be two CNFs from a tractable class. Then it is possible to test in polynomial time whether φ and ψ represent the same Boolean function (are logically equivalent) or not.*

Proof. It suffices to test for each clause C in φ whether it is an implicate of ψ and for each clause C in ψ whether it is an implicate of φ. The two CNFs are logically equivalent if and only if none of these tests fails. □

An example of a tractable class is the class of Horn CNFs, which we use most frequently in the subsequent text.

2.2. Forward chaining procedure

In verifying that a given clause is an implicate of a given pure Horn function, a very useful and simple procedure is the following. Let η be a pure Horn CNF of a pure Horn function h. We shall define a *forward chaining* procedure [13] which associates to any subset Q of the propositional variables of h a set M in the following way. The procedure takes as input the subset Q of propositional variables, initializes the set $M = Q$, and at each step it looks for a pure Horn clause $S \vee y$ in η such that $S \subseteq M$, and $y \notin M$. If such a clause is found, the propositional variable y is included into M, and the search is repeated as many times as possible. The set M output by this procedure will be denoted by $\mathrm{FC}_\eta(Q)$, where η is the input CNF and Q the starting set of variables. It can be shown [11,19] that a clause $C = Q \vee y$ is an implicate of h if and only if $y \in \mathrm{FC}_\eta(Q)$. If η' and η'' are two distinct CNF representations of a given pure Horn function h and if Q is an arbitrary subset of the propositional variables, then $\mathrm{FC}_{\eta'}(Q) = \mathrm{FC}_{\eta''}(Q)$ because η' and η'' have the same set of implicates. Therefore, the set of propositional variables reachable from Q by forward chaining depends only on the underlying function h rather than on a particular CNF representation η. For this reason, we shall also use the expression $\mathrm{FC}_h(Q)$ instead of $\mathrm{FC}_\eta(Q)$ whenever we do not want to refer to a specific CNF.

2.3. Essential sets

In this section we shall define the central notion of this paper, the essential set of clauses, which was introduced in [3].

Definition 2.4 ([3])**.** Given a Boolean function f, a subset $\mathcal{E} \subseteq \mathcal{I}(f)$ is called an *essential set of* f (or simply an *essential set* if f is clear from the context) if for every pair of resolvable clauses $C_1, C_2 \in \mathcal{I}(f)$ the following implication holds:

$$R(C_1, C_2) \in \mathcal{E} \Longrightarrow C_1 \in \mathcal{E} \text{ or } C_2 \in \mathcal{E},$$

i.e. the resolvent belongs to $\mathcal{E}$ only if at least one of the parent clauses is from $\mathcal{E}$.

It is easy to see that a set is essential if and only if its complement is closed under resolution. Hence, we have also the following characterization.

Theorem 2.5 ([3])**.** *A subset $\mathcal{E}$ of $\mathcal{I}(f)$ is an essential set of f iff $\mathcal{I}(f) \setminus \mathcal{E} = \mathcal{R}(\mathcal{I}(f) \setminus \mathcal{E})$.*

Note that the empty set is an essential set of any Boolean function. We shall often use the notion of a minimal (with respect to inclusion) essential set and we shall require that such a set is nonempty. For this reason, we exclude empty set when defining minimal essential set. In particular, we have the following definition.

Definition 2.6. We shall say that an essential set $\mathcal{E}$ is *minimal*, if $\mathcal{E} \neq \emptyset$ and the only essential set which is properly included in $\mathcal{E}$ is an empty set.

Definition 2.7. For a Boolean function f, let $\mathrm{ess}(f)$ be the maximum number of pairwise disjoint nonempty essential sets of implicates and let $\mathrm{cnf}(f)$ be the minimum number of clauses needed to represent f by a CNF.[2]

An important connection between $\mathrm{ess}(f)$ and $\mathrm{cnf}(f)$ was shown in [3].

Theorem 2.8 ([3])**.** *For every Boolean function f, we have* $\mathrm{cnf}(f) \geq \mathrm{ess}(f)$.

In Section 6.1, we demonstrate an example of a Horn Boolean function, for which we have $\mathrm{cnf}(f) > \mathrm{ess}(f)$. On the other hand, many useful functions satisfy $\mathrm{cnf}(f) = \mathrm{ess}(f)$. When this is satisfied, there is a polynomially verifiable certificate for this fact by Theorem 3.8. The main goal of this paper is to investigate the general properties of essential sets and to derive consequences for the properties of the class of functions satisfying $\mathrm{cnf}(f) = \mathrm{ess}(f)$.

Definition 2.9. Let f be a Boolean function. We shall call f *coverable* if $\mathrm{ess}(f) = \mathrm{cnf}(f)$. Let $\mathcal{X}$ be a set (or class) of CNFs. We shall call $\mathcal{X}$ *coverable* if every CNF from $\mathcal{X}$ represents a coverable function.

3. A polynomially verifiable certificate for ess(f)

Given a falsepoint t of f, we define

$$\mathcal{E}(t) = \{C \in \mathcal{I}(f) \mid C(t) = 0\}.$$

The assumption that t is a falsepoint of f implies $\mathcal{E}(t) \neq \emptyset$. Moreover, it is easy to verify that $\mathcal{E}(t)$ is an essential set of implicates (for a proof, see Lemma 6.5 in [3]). The set $\mathcal{E}(t)$ will be called a *falsepoint essential set defined by* t (or an *FE set defined by* t for brevity). It is easy to see that not every essential set is an FE set (consider e.g. the entire set $\mathcal{I}(f)$ which is of course an essential set of f—if it contains two clauses containing a pair of complementary literals then no vector t can falsify both such clauses). However, every minimal essential set of implicates is equal to $\mathcal{E}(t)$ for some t.

Theorem 3.1. *Let f be a Boolean function and let $\mathcal{E}$ be a minimal essential set of $\mathcal{I}(f)$, then there is some falsepoint t of f, such that $\mathcal{E} = \mathcal{E}(t)$.*

Proof. Let g be a function represented by clauses in $\mathcal{I}(f) \setminus \mathcal{E}$. Clearly $g \geq f$, because it is represented by implicates of f. Since $\mathcal{E}$ is essential, we have that $\mathcal{R}(\mathcal{I}(f) \setminus \mathcal{E}) = \mathcal{I}(f) \setminus \mathcal{E} \neq \mathcal{I}(f)$ by Theorem 2.5. By Lemma 4.3 in [3], we have that a subset of $\mathcal{I}(f)$ defines the function f if and only if its resolution closure is $\mathcal{I}(f)$. Hence, we have $g \neq f$. Therefore, there is a vector t, such that $g(t) = 1$, while $f(t) = 0$, and hence every implicate $C \in \mathcal{I}(f)$ for which $C(t) = 0$ belongs to $\mathcal{E}$. In other words, $\mathcal{E}(t) \subseteq \mathcal{E}$. Since $\mathcal{E}(t)$ is an essential set of implicates and $\mathcal{E}$ is a minimal essential set of $\mathcal{I}(f)$, we have $\mathcal{E}(t) = \mathcal{E}$. □

Let us use this fact to provide an equivalent characterization of $\mathrm{ess}(f)$.

Corollary 3.2. *Let f be an arbitrary Boolean function, then* $\mathrm{ess}(f)$ *is equal to the maximum number of disjoint FE sets.*

Proof. Let $k = \mathrm{ess}(f)$. The maximum number of disjoint FE sets of clauses is at most k because every FE set (i.e. $\mathcal{E}(t)$ for an arbitrary falsepoint t) is essential. For the opposite inequality, let $\mathcal{E}_1, \ldots, \mathcal{E}_k$ be a family of pairwise disjoint essential sets

[2] The first number is denoted by $\epsilon(f)$ and the second number by $\sigma(f)$ in [3].

of f. For every $i = 1, \ldots, k$ let $\mathcal{E}'_i$ be a minimal essential set, which is a subset of $\mathcal{E}_i$. Using Theorem 3.1 there exists a vector t_i, such that $\mathcal{E}'_i = \mathcal{E}(t_i)$. Hence $\mathcal{E}'_1, \ldots, \mathcal{E}'_k$ constitute k disjoint FE sets and so the maximum number of disjoint FE sets of clauses is at least k. □

Let us prove some further properties of FE sets which are used later.

Definition 3.3. Let s, t, r be Boolean vectors of length n. We say that r separates s and t, if for every $i = 1, \ldots, n$, we have $r_i = s_i$ or $r_i = t_i$.

Definition 3.4. Let s, t be Boolean vectors of length n. Then we denote

$$C_{st} = \bigvee_{i \in I(s,t)} \bar{x}_i \vee \bigvee_{i \in O(s,t)} x_i,$$

where sets $I(s, t)$ and $O(s, t)$ are defined as follows

$$I(s, t) = \{i \mid (1 \leq i \leq n) \wedge s[i] = t[i] = 1\}$$
$$O(s, t) = \{i \mid (1 \leq i \leq n) \wedge s[i] = t[i] = 0\}.$$

Note that r separates s, t if and only if $C_{st}(r) = 0$.

Lemma 3.5. *Let s and t be two falsepoints of a Boolean function f. Then the following statements are equivalent:*
1. $\mathcal{E}(s) \cap \mathcal{E}(t) \neq \emptyset$.
2. C_{st} *is an implicate of* f.
3. $\mathcal{E}(s) \cap \mathcal{E}(t) \cap \mathcal{I}^p(f) \neq \emptyset$.

Proof.
- (1) $\Longrightarrow$ (2): Let us assume that there exists an implicate $C' \in \mathcal{E}(s) \cap \mathcal{E}(t)$. Since $C'(t) = C'(s) = 0$, we have that variables of all positive literals of C' belong to $O(s, t)$ and variables of all negative literals of C' belong to $I(s, t)$. This in turn means that C' is a subclause of C_{st}. Therefore $C' \leq C_{st}$ and hence C_{st} is an implicate of f.
- (2) $\Longrightarrow$ (3): Let us assume that C_{st} is an implicate of f. Clearly, C_{st} evaluates to zero on both s and t (it is by its definition the "longest" clause with this property). Since C_{st} is an implicate of f, there exists a prime implicate $C' \in \mathcal{I}^p(f)$ such that $C' \leq C_{st}$ (i.e. C' is a subclause of C_{st}). Since C' also evaluates to zero on both s and t, we have $C' \in \mathcal{E}(s) \cap \mathcal{E}(t) \cap \mathcal{I}^p(f)$, which need not be true for C_{st}.
- (3) $\Longrightarrow$ (1): This implication is trivial. □

Note that if the given CNF representation of f is from a tractable class (e.g. if it is a Horn CNF), then for every pair of vectors s and t we can test in polynomial time, whether $\mathcal{E}(t) \cap \mathcal{E}(s) = \emptyset$ or not. This observation easily follows from Lemma 3.5 and the fact that testing whether a given clause is an implicate of a function given by a CNF from a tractable class can be done in polynomial time.

Corollary 3.6. *Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be two minimal essential sets of implicates of a Boolean function f, then $\mathcal{E}_1$ and $\mathcal{E}_2$ have a nonempty intersection if and only if there is a prime implicate of f which belongs to both $\mathcal{E}_1$ and $\mathcal{E}_2$.*

Proof. This directly follows from Theorem 3.1 and Lemma 3.5. □

The following formulation explicitly shows a certificate for the disjointness of two FE sets.

Lemma 3.7. *Let s and t be two falsepoints of a Boolean function f. Then $\mathcal{E}(s)$ and $\mathcal{E}(t)$ are disjoint if and only if there exists a truepoint r of f, which separates s and t.*

Proof. Since r separates s, t if and only if $C_{st}(r) = 0$, we obtain that there exists a truepoint r, which separates s, t if and only if C_{st} is not an implicate. Then, the lemma follows by taking negations of parts 1 and 2 in Lemma 3.5. □

Let us now formulate the following decision problem.

Problem ESS($\mathcal{F}, k$).
Input: A CNF $\mathcal{F}$ which represents a Boolean function f and a natural number k.
Question: ess(f) $\geq k$?

Now we shall show that this problem belongs to the class *NP*. In Sections 7 and 8, we shall prove that it is also *NP* hard.

Theorem 3.8. *Problem* ESS($\mathcal{F}, k$) *is in NP.*

Proof. Let a pair $\mathcal{F}, k$ be a positive instance of ESS($\mathcal{F}, k$), i.e. let ess(f) $\geq k$ hold, where f is the Boolean function represented by $\mathcal{F}$. Then by Corollary 3.2 there exist k falsepoints $t_1, \ldots, t_k$ of function f which define pairwise disjoint nonempty FE sets $\mathcal{E}(t_1), \ldots, \mathcal{E}(t_k)$. Let $1 \leq i < j \leq k$ be arbitrary. By Lemma 3.7 there exists a truepoint r_{ij} of f which separates t_i and t_j. However, now the vectors t_i, $1 \leq i \leq k$, and r_{ij}, $1 \leq i < j \leq k$ form a certificate for ess(f) $\geq k$. This certificate has a polynomial size with respect to the input CNF $\mathcal{F}$ because it consists of $O(k^2)$ vectors of length n while $\mathcal{F}$ consists of at least ess(f) $\geq k$ clauses by Theorem 2.8 (and we may assume without loss of generality that each of n variables appears at least once in $\mathcal{F}$). Of course, such a certificate is also polynomially verifiable: it suffices to check that every t_i, $1 \leq i \leq k$ is a falsepoint of f (by substituting the appropriate binary values into $\mathcal{F}$), and that every r_{ij}, $1 \leq i < j \leq k$ is a truepoint of f which separates t_i and t_j. □

4. CNF minimization for tractable classes

Let us start this section by formulating CNF minimization as a decision problem.

Problem CNF($\mathcal{F}$, ℓ).
Input: A CNF $\mathcal{F}$ which represents a Boolean function f and a natural number ℓ.
Question: cnf(f) $\leq \ell$?

We shall show that this decision problem is in *NP* when the input CNFs are restricted to some tractable class of CNFs.

Lemma 4.1. *Let $\mathcal{X}$ be a tractable class of CNFs. Then* CNF($\mathcal{F}$, ℓ) *is in NP for $\mathcal{F} \in \mathcal{X}$.*

Proof. Let a pair $\mathcal{F}$, ℓ be a positive instance of CNF($\mathcal{F}$, ℓ), i.e. let cnf(f) $\leq \ell$ hold, where f is the Boolean function represented by $\mathcal{F}$. Then a prime CNF $\mathcal{G}$, which represents f and consists of at most ℓ clauses is a polynomial size certificate for this inequality. Note that we may assume that $\mathcal{G}$ is a prime representation since the existence of a CNF representing f and consisting of at most ℓ clauses clearly implies the existence of a prime CNF with the same property. Moreover, the tractability of $\mathcal{X}$ implies $\mathcal{G} \in \mathcal{X}$. The fact that $\mathcal{G}$ is a polynomially verifiable certificate follows from the fact that both $\mathcal{F}$ and $\mathcal{G}$ belong to the tractable class $\mathcal{X}$, and hence it is possible to test in polynomial time that they both represent the same function f (see Lemma 2.3). □

Theorem 4.2. *Let $\mathcal{X}$ be a class of CNFs which is both tractable and coverable. Then* CNF($\mathcal{F}$, ℓ) *is in NP ∩ coNP for $\mathcal{F} \in \mathcal{X}$.*

Proof. The fact that CNF($\mathcal{F}$, ℓ) is in *NP* for $\mathcal{F} \in \mathcal{X}$, i.e. that there exists a polynomially verifiable certificate for a positive answer, follows directly from Lemma 4.1. Let f be the Boolean function represented by $\mathcal{F}$. A certificate for a negative answer is a certificate for the fact that cnf(f) $\geq \ell + 1$ which is the same as ess(f) $\geq \ell + 1$ since cnf(f) = ess(f) due to the fact that f is coverable. However, such a certificate, which is polynomially verifiable, exists due to Theorem 3.8. □

It should be remarked here that the requirement $\mathcal{X} \in P$, i.e. that there exists a polynomial time recognition algorithm for $\mathcal{X}$ (imposed on tractable classes in Definition 2.1), can be weakened to $\mathcal{X} \in NP \cap coNP$ while both Lemma 4.1 and Theorem 4.2 remain valid.

It can be also pointed out that even a stricter "equality version" of CNF($\mathcal{F}$, ℓ), where the input stays the same but the question is changed to cnf(f) = ℓ?, is still in *NP ∩ coNP* for $\mathcal{F}$ in a tractable and coverable class. A certificate for a positive answer is a combination of certificates for cnf(f) $\leq \ell$ and ess(f) $\geq \ell$, while a certificate for a negative answer is one of the certificates for cnf(f) $\leq \ell - 1$ or ess(f) $\geq \ell + 1$.

Theorem 4.2 indicates that if for a tractable class $\mathcal{X}$ one can show that $\mathcal{X}$ is coverable, then there is a good chance CNF($\mathcal{F}$, ℓ) is solvable in polynomial time for $\mathcal{F} \in \mathcal{X}$, as most decision problems known to be in *NP ∩ coNP* are in fact in *P*. This is indeed the case for all three classes known to be simultaneously tractable and coverable which were mentioned in the Introduction (acyclic Horn, quasi-acyclic Horn, and CQ-Horn CNFs). Let us now state a simple corollary.

Corollary 4.3. *Let $\mathcal{X}$ be a tractable class of CNFs for which the minimization problem* CNF($\mathcal{F}$, ℓ) *is NP-hard. Then $\mathcal{X}$ is not coverable unless NP = coNP.*

Proof. Let us proceed by contradiction and assume that $\mathcal{X}$ is coverable. Then by Theorem 4.2 we have that CNF($\mathcal{F}$, ℓ) is in *NP ∩ coNP* for $\mathcal{F} \in \mathcal{X}$, and by an assumption CNF($\mathcal{F}$, ℓ) is *NP*-hard for $\mathcal{F} \in \mathcal{X}$. However, the fact that an *NP*-complete problem falls into *coNP* implies *NP = coNP* (see e.g. [8]). □

There are many classes with *NP*-hard minimization which may play the role of class $\mathcal{X}$ in Corollary 4.3. A good example is the class of Horn CNFs [2]. Therefore, unless *NP = coNP*, there must exist a Horn CNF representing function f for which ess(f) < cnf(f). We shall construct such a CNF in Section 6.1 after we introduce further notation and derive results needed to prove the properties of such a CNF. In particular, we shall first concentrate on how to compute ess(f) using only clauses from $\mathcal{I}^p(f)$ instead of looking at the entire $\mathcal{I}(f)$ which may be much larger.

5. Prime essential sets

According to Corollary 3.6, in order to test the disjointness of minimal essential sets, it is sufficient to look at prime clauses. This suggests to consider the following notion.

Definition 5.1. Let f be an arbitrary Boolean function and let $\mathcal{E} \subseteq \mathcal{I}^p(f)$ be a set of prime implicates. We say that $\mathcal{E}$ is a *prime essential set of f* (or simply a *prime essential set* if f is clear from the context), if $\mathcal{E} = \mathcal{E}' \cap \mathcal{I}^p(f)$ for a set of clauses $\mathcal{E}'$ such that $\mathcal{E}'$ is an essential set of f. We shall say that a prime essential set $\mathcal{E}$ of f is *minimal*, if $\mathcal{E} \neq \emptyset$ and the only prime essential set of f which is properly included in $\mathcal{E}$ is the empty set.

Note that every nonempty essential set $\mathcal{E}'$ contains at least one prime implicate (otherwise the complement $\mathcal{I}(f) \setminus \mathcal{E}'$ contains all prime implicates implying $\mathcal{R}(\mathcal{I}(f) \setminus \mathcal{E}') = \mathcal{I}(f)$ and thus contradicting Theorem 2.5), so prime essential set $\mathcal{E} = \mathcal{E}' \cap \mathcal{I}^p(f)$ is nonempty whenever $\mathcal{E}'$ is nonempty.

In order to characterize prime essential sets in a way similar to the characterization of essential sets in Theorem 2.5, we introduce the following notation for the resolution closure restricted to prime clauses. Moreover, to make the presentation in subsequent sections simpler, we extend this notation also to FE sets.

Definition 5.2. Let f be an arbitrary Boolean function. For every set $\mathcal{C}$ of prime implicates of f, let $\mathcal{R}^p(\mathcal{C}) = \mathcal{R}(\mathcal{C}) \cap \mathcal{I}^p(f)$. For every FE set $\mathcal{E}(t)$ let $\mathcal{E}^p(t) = \mathcal{E}(t) \cap \mathcal{I}^p(f)$.

Theorem 5.3. *A subset $\mathcal{E}$ of $\mathcal{I}^p(f)$ is a prime essential set of f if and only if $\mathcal{I}^p(f) \setminus \mathcal{E} = \mathcal{R}^p(\mathcal{I}^p(f) \setminus \mathcal{E})$.*

Proof. First, assume that $\mathcal{E}$ is a prime essential set. Then, there is an essential set $\mathcal{E}'$ such that $\mathcal{E} = \mathcal{E}' \cap \mathcal{I}^p(f)$ and $\mathcal{E}'$ satisfies $\mathcal{I}(f) \setminus \mathcal{E}' = \mathcal{R}(\mathcal{I}(f) \setminus \mathcal{E}')$ by Theorem 2.5. Since $\mathcal{I}^p(f) \setminus \mathcal{E} \subseteq \mathcal{I}(f) \setminus \mathcal{E}'$, we have $\mathcal{R}^p(\mathcal{I}^p(f) \setminus \mathcal{E}) \subseteq \mathcal{R}^p(\mathcal{I}(f) \setminus \mathcal{E}') = (\mathcal{I}(f) \setminus \mathcal{E}') \cap \mathcal{I}^p(f) = \mathcal{I}^p(f) \setminus \mathcal{E}$. On the other hand, we have $\mathcal{I}^p(f) \setminus \mathcal{E} \subseteq \mathcal{R}^p(\mathcal{I}^p(f) \setminus \mathcal{E})$. Altogether, $\mathcal{R}^p(\mathcal{I}^p(f) \setminus \mathcal{E}) = \mathcal{I}^p(f) \setminus \mathcal{E}$.

For the opposite direction, assume $\mathcal{I}^p(f) \setminus \mathcal{E} = \mathcal{R}^p(\mathcal{I}^p(f) \setminus \mathcal{E})$ and define $\mathcal{E}' = \mathcal{I}(f) \setminus \mathcal{R}(\mathcal{I}^p(f) \setminus \mathcal{E})$. We have $\mathcal{I}(f) \setminus \mathcal{E}' = \mathcal{R}(\mathcal{I}^p(f) \setminus \mathcal{E})$ and, hence, $\mathcal{R}(\mathcal{I}(f) \setminus \mathcal{E}') = \mathcal{R}(\mathcal{R}(\mathcal{I}^p(f) \setminus \mathcal{E})) = \mathcal{R}(\mathcal{I}^p(f) \setminus \mathcal{E}) = \mathcal{I}(f) \setminus \mathcal{E}'$. Consequently, by Theorem 2.5, $\mathcal{E}'$ is an essential set. Since $(\mathcal{I}(f) \setminus \mathcal{E}') \cap \mathcal{I}^p(f) = \mathcal{R}^p(\mathcal{I}^p(f) \setminus \mathcal{E}) = \mathcal{I}^p(f) \setminus \mathcal{E}$, we also have $\mathcal{E}' \cap \mathcal{I}^p(f) = \mathcal{E}$ and $\mathcal{E}$ is a prime essential set. □

Theorem 5.4. *Let f be an arbitrary Boolean function. Then* $\mathrm{ess}(f)$ *is equal to the maximum number of pairwise disjoint prime essential sets of f.*

Proof. Let k be the maximum number of pairwise disjoint prime essential sets, and let $\mathcal{E}_1, \ldots, \mathcal{E}_{\mathrm{ess}(f)}$ be disjoint essential sets. Since $\mathcal{E}_i \cap \mathcal{I}^p$ are disjoint prime essential sets, we have $k \geq \mathrm{ess}(f)$.

Let $\mathcal{E}_i$ for $i = 1, \ldots, k$ be disjoint prime essential sets. Consider minimal essential sets $\mathcal{E}_i'$ such that $\mathcal{E}_i \supseteq \mathcal{E}_i' \cap \mathcal{I}^p$. If $\mathcal{E}_i'$ and $\mathcal{E}_j'$ for $i \neq j$ are not disjoint, then by Corollary 3.6 their intersection contains a prime implicate. This is a contradiction with the assumption that $\mathcal{E}_i$ and $\mathcal{E}_j$ are disjoint. Hence, $\mathcal{E}_1', \ldots, \mathcal{E}_k'$ are disjoint and we have $k \leq \mathrm{ess}(f)$. □

The following lemma gives a connection between minimal essential sets and minimal prime essential sets.

Lemma 5.5. *Let f be an arbitrary Boolean function, and let $\mathcal{E}_p$ be a minimal prime essential set of implicates of f, then there exist a minimal essential set $\mathcal{E}$ of implicates of f, such that $\mathcal{E}_p = \mathcal{E} \cap \mathcal{I}^p(f)$.*

Proof. By definition, there is an essential set $\mathcal{E}$ such that $\mathcal{E}_p = \mathcal{E} \cap \mathcal{I}^p$. Consider any minimal essential subset $\mathcal{E}'$ of $\mathcal{E}$. The intersection $\mathcal{E}' \cap \mathcal{I}^p$ is a nonempty subset of $\mathcal{E}_p$, which is a prime essential set. Since $\mathcal{E}_p$ is minimal, we have $\mathcal{E}' \cap \mathcal{I}^p = \mathcal{E}_p$. □

Note that the reverse direction of Lemma 5.5 does not hold in general, i.e. given a minimal essential set $\mathcal{E}$ of implicates of a Boolean function f, we cannot conclude that $\mathcal{E} \cap \mathcal{I}^p(f)$ is a minimal prime essential set. Consider the function f defined by the following set of clauses

$$\mathcal{F} = \{(a \vee b \vee \overline{c}), (a \vee b \vee \overline{d}), (c \vee \overline{d}), (c \vee \overline{e}), (b \vee \overline{c} \vee d), (a \vee b \vee \overline{e}), (\overline{c} \vee d \vee e), (b \vee d \vee \overline{e})\}.$$

The following set of additional clauses can be derived from $\mathcal{F}$ by resolution

$$\mathcal{G} = \{(a \vee b \vee \overline{c} \vee e), (a \vee b \vee \overline{c} \vee d), (a \vee b \vee d \vee \overline{e}),$$
$$(a \vee b \vee \overline{d} \vee e), (a \vee b \vee c \vee \overline{d}), (a \vee b \vee c \vee \overline{e}), (b \vee c \vee \overline{e})\}.$$

Notice that every clause in $\mathcal{G}$ contains some subclause from $\mathcal{F}$ which means that $\mathcal{I}^p(f) = \mathcal{F}$, and $\mathcal{I}(f) = \mathcal{F} \cup \mathcal{G}$. To verify this fact it suffices to check that for every pair of resolvable clauses from $\mathcal{F} \cup \mathcal{G}$ the resolvent already belongs to $\mathcal{F} \cup \mathcal{G}$. Moreover, if we denote $t_1 = (00111)$ and $t_2 = (00110)$, it can be checked that the sets of clauses

$$\mathcal{E}(t_1) = \{(a \vee b \vee \overline{c}), (a \vee b \vee \overline{d}), (a \vee b \vee \overline{e})\}$$
$$\mathcal{E}(t_2) = \{(a \vee b \vee \overline{c}), (a \vee b \vee \overline{d}), (a \vee b \vee \overline{c} \vee e), (a \vee b \vee \overline{d} \vee e)\}$$

are minimal essential sets of f. However, the sets

$$\mathcal{E}^p(t_1) = \mathcal{E}(t_1) \cap \mathcal{I}^p(f) = \mathcal{E}(t_1)$$
$$\mathcal{E}^p(t_2) = \mathcal{E}(t_2) \cap \mathcal{I}^p(f) = (a \vee b \vee \overline{c})(a \vee b \vee \overline{d}),$$

satisfy $\mathcal{E}^p(t_2) \subsetneq \mathcal{E}^p(t_1)$, and therefore $\mathcal{E}^p(t_1)$ is not a minimal prime essential set.

Corollary 5.6. *Let f be an arbitrary Boolean function and let $\mathcal{E}_p$ be a minimal prime essential set of implicates of f, then there is a falsepoint t of f, such that $\mathcal{E}_p = \mathcal{E}^p(t)$.*

Proof. According to Lemma 5.5 there is a minimal essential set $\mathcal{E}$ such that $\mathcal{E}_p = \mathcal{E} \cap \mathcal{I}^p(f)$. According to Theorem 3.1 there is a falsepoint t of f for which $\mathcal{E} = \mathcal{E}(t)$ and thus $\mathcal{E}_p = \mathcal{E}(t) \cap \mathcal{I}^p(f) = \mathcal{E}^p(t)$. □

The following theorem appears in [3] as two parts, one implication as Corollary 6.14 and the other as Theorem 6.15.

Theorem 5.7 ([3])**.** *Let f be an arbitrary Boolean function and let $\mathcal{E} \subseteq \mathcal{I}(f)$ be an arbitrary set of clauses. Then $\mathcal{E}$ is a minimal essential set iff $\mathcal{E}$ is a minimal (with respect to inclusion) subset of $\mathcal{I}(f)$ such that $\mathcal{E} \cap \mathcal{C} \neq \emptyset$ for every $\mathcal{C} \subseteq \mathcal{I}(f)$ which represents f.*

We can now state a similar result for prime essential sets.

Theorem 5.8. *Let f be an arbitrary Boolean function and let $\mathcal{E}_p \subseteq \mathcal{I}^p(f)$ be an arbitrary set of prime implicates of f. Then $\mathcal{E}_p$ is a minimal prime essential set iff $\mathcal{E}_p$ is a minimal (with respect to inclusion) subset of $\mathcal{I}(f)$ such that $\mathcal{E}_p \cap \mathcal{C} \neq \emptyset$ for every $\mathcal{C} \subseteq \mathcal{I}^p(f)$ which represents f.*

Proof. First, let us prove both directions of the equivalence without proving the minimality of the corresponding set in the conclusion.

Let $\mathcal{E}_p$ be a minimal prime essential set and let $\mathcal{C} \subseteq \mathcal{I}^p(f)$ be an arbitrary prime representation of f. By Lemma 5.5 there exists a minimal essential set $\mathcal{E}$ such that $\mathcal{E}_p = \mathcal{E} \cap \mathcal{I}^p(f)$. Theorem 5.7 now implies that $\mathcal{E} \cap \mathcal{C} \neq \emptyset$. This fact together with the assumption $\mathcal{C} \subseteq \mathcal{I}^p(f)$ gives us $\mathcal{E}_p \cap \mathcal{C} \neq \emptyset$. Since $\mathcal{C}$ was an arbitrary prime representation of f we get that $\mathcal{E}_p \cap \mathcal{C} \neq \emptyset$ for every $\mathcal{C} \subseteq \mathcal{I}^p(f)$ which represents f.

Now let us assume that $\mathcal{E}_p \cap \mathcal{C} \neq \emptyset$ for every $\mathcal{C} \subseteq \mathcal{I}^p(f)$ which represents f and that $\mathcal{E}_p$ is a minimal subset of $\mathcal{I}(f)$ with this property. It follows that $\mathcal{E}_p \subseteq \mathcal{I}^p(f)$. Let us show that $\mathcal{E}_p$ is a prime essential set. Let $\mathcal{E}' \subseteq \mathcal{I}(f) \setminus \mathcal{I}^p(f)$ be a minimal (with respect to inclusion) set of nonprime implicates such that $(\mathcal{E}' \cup \mathcal{E}_p) \cap \mathcal{C} \neq \emptyset$ for every $\mathcal{C} \subseteq \mathcal{I}(f)$ which represents f. Since $\mathcal{E}_p$ already intersects every prime representation of f, adding nonprime implicates is sufficient. By construction, $\mathcal{E} = \mathcal{E}' \cup \mathcal{E}_p$ is a minimal subset of $\mathcal{I}(f)$ such that $\mathcal{E} \cap \mathcal{C} \neq \emptyset$ for every $\mathcal{C} \subseteq \mathcal{I}(f)$ which represents f and thus by Theorem 5.7 we obtain that $\mathcal{E}$ is a minimal essential set of f. Moreover, $\mathcal{E}_p = \mathcal{E} \cap \mathcal{I}^p(f)$ holds and hence $\mathcal{E}_p$ is a prime essential set.

It remains to show that in both directions, we get, in fact, minimal sets. Let $\mathcal{E}_p$ be a minimal prime essential set. By the first paragraph of the proof we know that this set intersects any prime representation of f. If there is a proper subset of $\mathcal{E}_p$, which also intersects every prime representation, then by the second paragraph, this subset is a prime essential set. This is not possible, since $\mathcal{E}_p$ was a minimal prime essential set.

Now, let $\mathcal{E}_p$ be an inclusion minimal set, which intersects every prime representation. By the second paragraph of the proof, we know that it is a prime essential set. If there is a proper subset of $\mathcal{E}_p$, which is also a prime essential set, then, by the first paragraph of the proof, it also intersects every prime representation. This is not possible, since $\mathcal{E}_p$ is an inclusion minimal set with this property. □

Now let us recall Theorem 6.6 from [3]. The following theorem is a minor strengthening of that theorem, which uses the fact that a set intersects every nonempty essential set if and only if it intersects every minimal essential set.

Theorem 5.9 ([3])**.** *Let f be an arbitrary Boolean function. A set $\mathcal{C} \subseteq \mathcal{I}(f)$ is a representation of f iff $\mathcal{C}$ intersects every minimal essential set of f.*

This statement gives a direct corollary for prime essential sets.

Corollary 5.10. *Let f be an arbitrary Boolean function. A set $\mathcal{C} \subseteq \mathcal{I}^p(f)$ is a representation of f iff $\mathcal{C}$ intersects every minimal prime essential set of f.*

In the following section we shall use Corollary 5.10 in the following way: for a given function f we first list all minimal prime essential sets of f, and then use this list to compute how many clauses are needed to intersect every set in the list, i.e. to compute $\mathrm{cnf}(f)$.

6. Examples of functions with $\mathrm{cnf}(f) > \mathrm{ess}(f)$

In the end of Section 4 we have noticed that unless $NP = coNP$, there must exist a Horn CNF representing function f for which $\mathrm{ess}(f) < \mathrm{cnf}(f)$. We shall start this section by constructing such a CNF.

6.1. Cubic pure Horn example on 4 variables

Let us consider pure Horn clauses $C_1 = (x_1 \vee \bar{x}_2 \vee \bar{x}_3)$, $C_2 = (\bar{x}_1 \vee x_2 \vee \bar{x}_3)$, $C_3 = (\bar{x}_1 \vee \bar{x}_2 \vee x_3)$, $Q_1 = (\bar{y} \vee x_1)$, $Q_2 = (\bar{y} \vee x_2)$, $Q_3 = (\bar{y} \vee x_3)$ and function f defined by CNF $\mathcal{F} = C_1 \wedge C_2 \wedge C_3 \wedge Q_1 \wedge Q_2 \wedge Q_3$. Notice that each pair among the three cubic clauses C_1, C_2, C_3 has two complementary pairs of literals and hence no such pair of clauses is resolvable. Moreover, no pair among the three quadratic clauses Q_1, Q_2, Q_3 has a complementary pair of literals and thus again no such pair of clauses is resolvable. In fact, there are only six resolvable pairs in the set $\mathcal{S} = \{C_1, C_2, C_3, Q_1, Q_2, Q_3\}$ (all of them "mixed pairs" of one cubic and one quadratic clause), namely (C_1, Q_2), (C_1, Q_3), (C_2, Q_1), (C_2, Q_3), (C_3, Q_1), (C_3, Q_2). It is easy to check that each of the six resolvents is absorbed by some other clause in $\mathcal{S}$ (e.g. the resolvent $x_1 \vee \bar{y} \vee \bar{x}_3$ of the pair (C_1, Q_2) is absorbed by Q_1). Thus, using Quine's resolution procedure to obtain the set of all prime implicates (canonical CNF) of a function defined by CNF $\mathcal{F}$ (this procedure is described, e.g. in [4]), it follows that $\mathcal{I}^p(f) = \mathcal{S}$.

Consider the vectors $t_1 = (0, 1, 1, 0)$, $t_2 = (1, 0, 1, 0)$, $t_3 = (1, 1, 0, 0)$, $t_4 = (0, 0, 1, 1)$, $t_5 = (1, 0, 0, 1)$, $t_6 = (0, 1, 0, 1)$ as truth value assignments of the variables x_1, x_2, x_3, y. These vectors define the following prime essential sets of clauses.

$$\mathcal{E}(t_1) = \{C_1\}$$
$$\mathcal{E}(t_2) = \{C_2\}$$
$$\mathcal{E}(t_3) = \{C_3\}$$
$$\mathcal{E}(t_4) = \{Q_1, Q_2\}$$
$$\mathcal{E}(t_5) = \{Q_2, Q_3\}$$
$$\mathcal{E}(t_6) = \{Q_1, Q_3\}.$$

It is obvious that $\mathcal{E}(t_1), \mathcal{E}(t_2), \mathcal{E}(t_3)$ are minimal prime essential sets as they contain one clause each. To see that also $\mathcal{E}(t_4), \mathcal{E}(t_5), \mathcal{E}(t_6)$ are minimal prime essential sets it suffices to check that the sets $\mathcal{I}^p(f)\setminus\{Q_1\}$, $\mathcal{I}^p(f)\setminus\{Q_2\}$, and $\mathcal{I}^p(f)\setminus\{Q_3\}$, are not closed under $\mathcal{R}^p$, which by Theorem 5.3 implies that none of the sets $\{Q_1\}$, $\{Q_2\}$, and $\{Q_3\}$ is a prime essential set. Moreover, this observation immediately implies that every nonempty prime essential set must contain either one of the cubic clauses C_1, C_2, C_3 or two of the quadratic clauses Q_1, Q_2, Q_3. In other words every nonempty prime essential set must contain one of $\mathcal{E}(t_1), \ldots, \mathcal{E}(t_6)$, which in turn implies that $\mathcal{E}(t_1), \ldots, \mathcal{E}(t_6)$ is a complete list of minimal prime essential sets of f.

Now, using Corollary 5.10 we obtain that $\mathrm{cnf}(f) = 5$. Indeed, all three cubic clauses must be present in $\mathcal{C}$ to intersect $\mathcal{E}(t_1), \mathcal{E}(t_2), \mathcal{E}(t_3)$ and a single quadratic clause is not sufficient to intersect all of $\mathcal{E}(t_4), \mathcal{E}(t_5), \mathcal{E}(t_6)$. Thus we need a minimum of two quadratic clauses which yields the only three minimum cardinality prime representations of f as follows:

$$\varphi_1 = C_1 \wedge C_2 \wedge C_3 \wedge Q_1 \wedge Q_2$$
$$\varphi_2 = C_1 \wedge C_2 \wedge C_3 \wedge Q_2 \wedge Q_3$$
$$\varphi_3 = C_1 \wedge C_2 \wedge C_3 \wedge Q_1 \wedge Q_3.$$

It can now also be easily checked that there are at most 4 pairwise disjoint minimal prime essential sets of implicates of f ($\mathcal{E}(t_1), \mathcal{E}(t_2), \mathcal{E}(t_3)$ together with one of $\mathcal{E}(t_4), \mathcal{E}(t_5), \mathcal{E}(t_6)$) which implies that there are at most 4 pairwise disjoint nonempty prime essential sets of implicates of f and using Theorem 5.4 we get that $\mathrm{ess}(f) = 4$.

The just constructed example has a gap $\mathrm{cnf}(f) - \mathrm{ess}(f) = 5 - 4 = 1$. In the following section we shall show that the gap $\mathrm{cnf}(f) - \mathrm{ess}(f)$ can be made arbitrarily large.

6.2. More general example

Let x_A be a set of n_1 variables and y_B a set of n_2 variables, where A, B are disjoint sets of indices and $n_1 = 2k - 1$ for some integer k. Let us define a function f_{n_1,n_2} of $n = n_1 + n_2$ variables by

$$f_{n_1,n_2}(x_A, y_B) = \left(\left(\bigvee_{i\in B} y_i \vee \sum_{i\in A} x_i \geq k\right) \Rightarrow \bigwedge_{i\in A} x_i\right)$$

or, equivalently, by the following CNF

$$f_{n_1,n_2}(x_A, y_B) = \left(\bigwedge_{\substack{i\in A\\ j\in B}} (\overline{y_j} \vee x_i)\right)\left(\bigwedge_{\substack{A'\subseteq A,\ |A'|=k\\ j\in A\setminus A'}} \left(\bigvee_{i\in A'} \overline{x_i} \vee x_j\right)\right).$$

Lemma 6.1. *The clauses of the above representation of f_{n_1,n_2} form exactly the list of its prime implicates. In other words, the list of prime implicates of f_{n_1,n_2} is formed by the following two types of clauses.*

1. *For every $i \in A$ and $j \in B$, the clause $\overline{y_j} \vee x_i$.*
2. *For every $A' \subseteq A$ satisfying $|A'| = k$ and every $j \in A \setminus A'$, the clause $\bigvee_{i\in A'} \overline{x_i} \vee x_j$.*

Proof. Resolution may be applied either to two clauses of type 2 or to a clause of type 1 and a clause of type 2. One may verify by case inspection that in the former case, the result is either a clause of type 2 or a superset of some of these clauses. In the latter case, the result is a superset of a clause of type 1. Consequently, the set of clauses from the lemma is stable under Quine's procedure [4] and hence, is a list of all prime implicates of the function defined by the conjunction of its elements. □

Theorem 6.2. *The following two types of sets of clauses represent exactly all minimal prime essential sets for f_{n_1,n_2}.*

1. *For every $j \in B$ and every $A' \subseteq A$, such that $|A'| = k$, let $P(j, A')$ be the set of prime implicates*

$$\{\overline{y_j} \vee x_i \mid i \in A'\}.$$

2. *For every $A' \subseteq A$ satisfying $|A'| \in [k, n_1 - 1]$ let $Q(A')$ be the set of prime implicates*

$$\left\{\bigvee_{i\in A''} \overline{x_i} \vee x_j \mid A'' \subseteq A',\ |A''| = k,\ j \in A \setminus A'\right\}.$$

Proof. Let us describe the falsepoints t of f_{n_1,n_2}, for which $\mathcal{E}^p(t)$ is minimal. It follows from the definition of the function f_{n_1,n_2} that a vector $t \in \{0,1\}^{n_1+n_2}$ is a falsepoint iff t satisfies at least one of the following conditions:

(1) There are $i \in A$ and $j \in B$ such that $t[x_i] = 0$ and $t[y_j] = 1$, or
(2) $\sum_{i\in A} t[x_i] \in [k, n_1 - 1]$.

If a falsepoint t satisfies both (1) and (2), then $\mathcal{E}^p(t)$ is not minimal, since if t' differs from t by setting $t[y_j] = 0$ for every $j \in B$, then t' is again a falsepoint and $\mathcal{E}^p(t') \subsetneq \mathcal{E}^p(t)$. If t is a falsepoint satisfying (1) and not (2), then according to Lemma 6.1

$$\mathcal{E}^p(t) \subseteq \{\overline{y_j} \vee x_i \mid i \in A \text{ and } j \in B\}.$$

If t' is a falsepoint satisfying (2) and not (1), then according to Lemma 6.1

$$\mathcal{E}^p(t') \subseteq \left\{ \bigvee_{i\in A'} \overline{x_i} \vee x_j \mid A' \subseteq A, |A'| = k \text{ and } j \in A \setminus A' \right\}.$$

It follows that for any such pair of falsepoints t and t' we have $\mathcal{E}^p(t) \cap \mathcal{E}^p(t') = \emptyset$, hence, we may consider the candidates for falsepoints t with minimal $\mathcal{E}^p(t)$ in these two groups separately.

Falsepoints satisfying (1) *and not* (2)

Let t be an arbitrary falsepoint satisfying (1) and not (2) such that $\mathcal{E}^p(t)$ is a minimal prime essential set. It follows that

$$\sum_{i\in A} t[x_i] < k.$$

If $\sum_{i\in A} t[x_i] < k-1$, then we can produce a falsepoint t' from t by setting $t'[x_i] = 1$ for some $i \in A$ for which $t[x_i] = 0$. Falsepoint t' still satisfies (1) and not (2) and $\mathcal{E}^p(t') \subsetneq \mathcal{E}^p(t)$, because there are fewer unsatisfied clauses of form $\overline{y_j} \vee x_i$ on t'. Since we assume that $\mathcal{E}^p(t)$ is a minimal prime essential set, we get

$$\sum_{i\in A} t[x_i] = k - 1.$$

Similarly, if there are at least two indices $j_1, j_2 \in B$ such that $t[y_{j_1}] = t[y_{j_2}] = 1$, then we can produce falsepoint t' from t by setting $t'[y_{j_2}] = 0$. Falsepoint t' again satisfies (1) and not (2) and $\mathcal{E}^p(t') \subsetneq \mathcal{E}^p(t)$ because there are fewer unsatisfied clauses of form $\overline{y_j} \vee x_i$ on t'. Since we assume that $\mathcal{E}^p(t)$ is a minimal prime essential set this is not possible, and thus there is exactly one $j \in B$ such that $t[y_j] = 1$. Together we have that a falsepoint t which satisfies (1) and not (2) defines a minimal prime essential sets only if the following two conditions are satisfied:

(a) There is exactly one $j \in B$ such that $t[y_j] = 1$ and
(b) $\sum_{i\in A} t[x_i] = k - 1$.

Now we shall show that these two conditions are also sufficient for t to define a minimal prime essential set $\mathcal{E}^p(t)$.

Let t and t' be two different falsepoints satisfying (a) and (b), and let us assume that $j_1 \in B$ is the only index such that $t[y_{j_1}] = 1$ and that $j_2 \in B$ is the only index such that $t'[y_{j_2}] = 1$. We shall show that $\mathcal{E}^p(t)$ is incomparable with $\mathcal{E}^p(t')$. If $j_1 \neq j_2$ then clearly $\mathcal{E}^p(t) \cap \mathcal{E}^p(t') = \emptyset$ so let us suppose that $j_1 = j_2$. Since $t \neq t'$ but $\sum_{i\in A} t[x_i] = \sum_{i\in A} t'[x_i] = k - 1$, we have that there are $i_1, i_2 \in A$ for which $t[x_{i_1}] = 0$, $t[x_{i_2}] = 1$, $t'[x_{i_1}] = 1$, and $t'[x_{i_2}] = 0$. Clearly $\overline{y_{j_1}} \vee x_{i_1} \in \mathcal{E}^p(t) \setminus \mathcal{E}^p(t')$ and $\overline{y_{j_1}} \vee x_{i_2} \in \mathcal{E}^p(t') \setminus \mathcal{E}^p(t)$, and therefore $\mathcal{E}^p(t)$ and $\mathcal{E}^p(t')$ are incomparable. It follows that every falsepoint t satisfying both (a) and (b) defines a minimal prime essential set $\mathcal{E}^p(t)$.

Let t be a falsepoint satisfying (a) and (b), let $j \in B$ be the only index for which $t[y_j] = 1$, and let us denote $A' = \{i \in A \mid t[x_i] = 0\}$. Since $n_1 = 2k - 1$ we have that $|A'| = k$ and thus

$$\mathcal{E}^p(t) = \{\overline{y_j} \vee x_i \mid i \in A'\} = P(j, A').$$

On the other hand, if $j \in B$ and $A' \subseteq A$ with $|A'| = k$ one can easily construct a falsepoint t which satisfies (a) and (b) and for which $\mathcal{E}^p(t) = P(j, A')$ as follows:

$$\begin{aligned} &t[y_{j'}] = 0 \quad j' \in B \setminus \{j\} \\ &t[y_j] = 1 \\ &t[x_i] = 0 \quad i \in A' \\ &t[x_i] = 1 \quad i \in A \setminus A'. \end{aligned}$$

Therefore, the sets $P(j, A')$ are in one-to-one correspondence with falsepoints satisfying (a) and (b). Note also that since $k > n_1/2$ we have that $P(j, A') \cap P(j, A'') \neq \emptyset$ for every $j \in B$ and $A', A'' \subseteq A$ where $|A'| = |A''| = k$.

Falsepoints satisfying (2) *and not* (1)

If t is a falsepoint satisfying (2) and not (1) then $t[y_j] = 0$ for every $j \in B$. Let us show that for any two such falsepoints t_1 and t_2 the sets $\mathcal{E}^p(t_1)$ and $\mathcal{E}^p(t_2)$ are incomparable. Let $A'_i = \{j \in A \mid t_i[x_j] = 1\}$, $i = 1, 2$, we have $|A'_i| \in [k, n_1 - 1]$ for $i = 1, 2$. We shall distinguish two cases, whether A'_1 and A'_2 are comparable, or not.

If A'_1 and A'_2 are incomparable, choose some $a_1 \in A'_1 \setminus A'_2$ and $a_2 \in A'_2 \setminus A'_1$. Further, let A''_1 be an arbitrary subset of A'_1 of size k and similarly, A''_2 a subset of A'_2 of size k. Then the clause

$$\bigvee_{i \in A''_1} \overline{x_i} \vee x_{a_2} \tag{1}$$

evaluates to 0 on t_1 and to 1 on t_2. Similarly, the clause

$$\bigvee_{i \in A''_2} \overline{x_i} \vee x_{a_1} \tag{2}$$

evaluates to 1 on t_1 and to 0 on t_2. Consequently, $\mathcal{E}^p(t_1)$ and $\mathcal{E}^p(t_2)$ are incomparable.

If A'_1 and A'_2 are comparable, assume w.l.o.g. $A'_1 \subsetneq A'_2$ and let $a_2 \in A'_2 \setminus A'_1$. Then, there is a clause constructed similarly to (1), which evaluates to 0 on t_1 and to 1 on t_2. On the other hand, if A''_2 is a k element subset of A'_2, which is not a subset of A'_1, and a_1 is an arbitrary index not contained in A'_2, then the clause of the form (2) evaluates to 1 on t_1 and to 0 on t_2. Consequently, $\mathcal{E}^p(t_1)$ and $\mathcal{E}^p(t_2)$ are incomparable.

It follows that given a falsepoint t satisfying (2) and not (1), $\mathcal{E}^p(t)$ is a minimal prime essential set, if we denote $A' = \{i \in A \mid t[x_i] = 1\}$, then $|A'| \in [k, n_1 - 1]$ and

$$\mathcal{E}^p(t) = \left\{ \bigvee_{i \in A''} \overline{x_i} \vee x_j \mid A'' \subseteq A',\ |A''| = k,\ j \in A \setminus A' \right\}$$

and therefore $\mathcal{E}^p(t) = Q(A')$. On the other hand, given $A' \subseteq A$, $|A'| \in [k, n_1 - 1]$ we can easily define a falsepoint t satisfying (2) and not (1) for which $\mathcal{E}^p(t) = Q(A')$ by setting $t[x_i] = 1$ iff $i \in A'$ and $t[y_j] = 0$ for every $j \in B$. Thus the sets $Q(A')$ are in one-to-one correspondence with falsepoints satisfying (2) and not (1).

Conclusion

By considering the two cases above, we obtained that the list of prime essential sets presented in the theorem is the list of all minimal prime essential sets of f_{n_1,n_2}. □

Lemma 6.3. *Let n, k be integers such that $n \geq 3$ and $1 \leq k \leq n-1$. Let A be a set of size n. Let $G_{n,k}$ be the undirected graph of subsets of A of size k, where two sets A' and A'' form an edge if and only if their symmetric difference has size 2. Then $G_{n,k}$ contains a Hamiltonian cycle.*

Proof. If $n \geq 3$ and $k = 1$ or $k = n-1$, then $G_{n,k}$ is a complete graph, so it contains Hamiltonian cycle. In particular, this proves the lemma for $n = 3$. Let us continue by induction on n.

Assume $n > 3$. It is sufficient to prove the statement for k satisfying $2 \leq k \leq n-2$. The vertices of $G_{n,k}$ are subsets of $\{1, \ldots, n\}$. Let G_1 be the subgraph of $G_{n,k}$ induced by the vertices containing 1, and let G_2 be the subgraph of $G_{n,k}$ induced by the remaining vertices. Graph G_1 is isomorphic to $G_{n-1,k-1}$ and G_2 is isomorphic to $G_{n-1,k}$. Hence, by induction hypothesis, both G_1 and G_2 contain a Hamiltonian cycle. Fix a Hamiltonian cycle in G_2 and choose an edge (A', A'') contained in it. Let u be an element in the intersection of A' and A''. Such an element exists, since the intersection has size $k-1$ and $k \geq 2$. Let $B' = A' \setminus \{u\} \cup \{1\}$ and $B'' = A'' \setminus \{u\} \cup \{1\}$. Sets B' and B'' are vertices of G_1 connected by an edge. Since each edge in G_1 may be mapped to any other edge in G_1 by an isomorphism of G_1, there is a Hamiltonian cycle in G_1 containing the edge (B', B''). By removing the edges (A', A'') and (B', B'') from the two Hamiltonian cycles and by connecting them using edges (A', B') and (A'', B''), we obtain a Hamiltonian cycle in $G_{n,k}$. □

Theorem 6.4. *For the function f_{n_1,n_2} defined above, we have*

$$\mathrm{cnf}(f_{n_1,n_2}) = \binom{n_1}{k} + kn_2 \tag{3}$$

$$\binom{n_1}{k} + n_2 \leq \mathrm{ess}(f_{n_1,n_2}) < 2^{n_1} + n_2. \tag{4}$$

Proof. In order to prove (3), we may restrict ourselves to prime CNFs representing f_{n_1,n_2}. Given an arbitrary prime CNF φ representing f_{n_1,n_2} and an arbitrary $A' \subseteq A$ of size $|A'| = k$, φ has to contain at least one clause of the form $\bigvee_{i \in A'} \overline{x_i} \vee x_j$ where $j \in A \setminus A'$, since otherwise $\mathrm{FC}_\varphi(A') = A'$ contradicting the fact that $\bigvee_{i \in A'} \overline{x_i} \vee x_j$ is an implicate of f_{n_1,n_2} for every $j \in A \setminus A'$. Hence, φ has to contain at least $\binom{n_1}{k}$ clauses of this form. To show that this number of clauses of this type is also sufficient, use Lemma 6.3 to prove the existence of a cycle consisting of all subsets of A of size k and such that sets, which are neighbors in the cycle, have symmetric difference of size 2. Then, consider the cycle as an ordered cycle with any of the two possible orderings. Finally, for each set A' in the cycle, consider the clause $\bigvee_{i \in A'} \overline{x_i} \vee x_j$, where j is the uniquely determined index not contained in A', but contained in the set, which follows A' in the cycle. It is easy to verify that the obtained set of clauses of size $\binom{n_1}{k}$ generates by forward chaining every prime clause listed in item 2 in Lemma 6.1 (by starting with the set of subgoals and following the cycle to the desired head).

Similarly, for each $i \in B$, φ has to contain a superset of the set of clauses $P(i, A')$ for some set A' of size k, since otherwise $\mathrm{FC}_\varphi(\{y_i\})$ contains only those x_j for which clauses $\overline{y_i} \vee x_j$ are explicitly present in φ, contradicting the fact that $\overline{y_i} \vee x_j$ is an

implicate of f_{n_1,n_2} for every $j \in A$. Thus φ contains at least kn_2 clauses of this form. To show that this number of clauses of this type is also sufficient, take for each $i \in B$ exactly one set $P(i, A')$ for some arbitrary set A' of size k. Now $\mathrm{FC}_\varphi(\{y_i\})$ contains all x_j for $j \in A'$ and using the cycle of clauses from the previous paragraph forward chaining derives all remaining x_j for $j \in A \setminus A'$.

In order to prove (4), we use Theorem 6.2 to find disjoint essential sets. Let us consider the two types of minimal essential sets listed in Theorem 6.2 separately. For each index $i \in B$, we can choose at most one of the sets $P(i, A')$, since for every pair of sets A', A'', sets $P(i, A')$ and $P(i, A'')$ have nonempty intersection. Hence, we have at most n_2 disjoint minimal essential sets of this type. If we choose $Q(A')$ for all $A' \subseteq A$ of size $|A'| = k$, we obtain $\binom{n_1}{k}$ further disjoint essential sets. On the other hand, the number of different sets A' is at most 2^{n_1}. Altogether, we can find at least $\binom{n_1}{k} + n_2$ and at most $2^{n_1} + n_2$ pairwise disjoint minimal essential sets. □

Corollary 6.5. *For fixed n_1, k and $n_2 \to \infty$, we have* $\mathrm{cnf}(f_{n_1,n_2})/\mathrm{ess}(f_{n_1,n_2}) \to k$.

7. Hardness of computing ess(f) for pure Horn 3CNFs

In this section we shall show that the following problem is *NP*-complete:

Problem: *ESS-Horn-3CNF*.
Input: A pure Horn 3CNF φ representing a pure Horn function f and an integer $k \geq 0$.
Question: Is $\mathrm{ess}(f) \geq k$?

We shall prove the hardness of this problem by a transformation from the problem of finding a maximum independent set in a graph G. This reduction is inspired by a similar construction in [2] where a reduction from the Set Cover problem to Boolean minimization (BM) is presented. For this purpose, let us associate a pure Horn function f_G with every undirected graph $G = (V, E)$, where $V = \{x_1, x_2, \ldots, x_n\}$, $n = |V|$, $E = \{e_{i,j} \mid e_{i,j} = \{x_i, x_j\}\}$, and $m = |E|$. With every vertex $x_i \in V$ of G we associate a Boolean variable x_i and similarly with every edge $e_{i,j} \in E$ we associate a Boolean variable $e_{i,j}$ (note that since G is an undirected graph, $e_{i,j} = e_{j,i}$). f_G is then a function on $n + m + 1$ variables, n variables associated with vertices, m variables associated with edges and an additional variable z. f_G is defined by the following pure Horn CNF expression

$$\mathcal{F}_G = \bigwedge_{e_{i,j} \in E} \left((\overline{e_{i,j}} \vee x_i) \wedge (\overline{e_{i,j}} \vee x_j) \wedge (\overline{x_i} \vee \overline{x_j} \vee e_{i,j}) \right) \wedge \bigwedge_{i=1}^{n} (\overline{z} \vee x_i).$$

Let us at first examine, how the prime implicates of f_G (i.e. the set $\mathcal{J}^p(f_G)$) may look like.

Lemma 7.1. *Let G be an arbitrary undirected graph and let f_G be its associated pure Horn function defined by CNF $\mathcal{F}_G$. A clause C is a prime implicate of f_G if and only if one of the following is true:*

(a) $C = (\overline{e_{i,j}} \vee x_i)$ *for some edge* $e_{i,j} \in E$,
(b) $C = (\overline{x_i} \vee \overline{x_j} \vee e_{i,j})$ *for some edge* $e_{i,j} \in E$,
(c) $C = (\overline{e_{i,j}} \vee \overline{x_k} \vee e_{i,k})$ *for some edges* $e_{i,j}, e_{i,k} \in E$ *where* x_i, x_j, *and* x_k *are three pairwise different vertices of* G,
(d) $C = (\overline{e_{i,j}} \vee \overline{e_{k,l}} \vee e_{i,k})$ *for some edges* $e_{i,j}, e_{i,k}, e_{k,l} \in E$ *where* x_l *may be the same vertex as* x_j,
(e) $C = (\overline{z} \vee e_{i,j})$ *for some edge* $e_{i,j} \in E$, *or*
(f) $C = (\overline{z} \vee x_i)$ *for some vertex* $x_i \in V$.

Proof. Let us first verify that each clause described in the proposition of the lemma is an implicate of f_G. The cases (a), (b), and (f) are trivial as these are the clauses appearing directly in $\mathcal{F}_G$. A clause $C = (\overline{e_{i,j}} \vee \overline{x_k} \vee e_{i,k})$ from case (c) is a resolvent of $C_1 = (\overline{e_{i,j}} \vee x_i)$ and $C_2 = (\overline{x_i} \vee \overline{x_k} \vee e_{i,k})$ and is therefore an implicate of f_G. A clause $C = (\overline{e_{i,j}} \vee \overline{e_{k,l}} \vee e_{i,k})$ from case (d) is a resolvent of $C_1 = (\overline{e_{i,j}} \vee \overline{x_k} \vee e_{i,k})$, which is an implicate due to (c), and $C_2 = (\overline{e_{k,l}} \vee x_k)$, which is an implicate due to (a). A clause $C = (\overline{z} \vee e_{i,j})$ from case (e) is a resolvent of $C_1 = (\overline{z} \vee x_i)$ and $C_2 = (\overline{z} \vee \overline{x_i} \vee e_{i,j})$, where C_2 is a resolvent of $C_4 = (\overline{z} \vee x_j)$ and $C_5 = (\overline{x_i} \vee \overline{x_j} \vee e_{i,j})$.

In order to verify that the clauses (a)–(f) are prime implicates, let us use the following set of satisfying assignments.

- All ones assignment.
- For every vertex x_a, $a \in \{1, \ldots, n\}$, the assignment, which sets z, x_a and $e_{a,i}$ for all $i \neq a$ to 0 and all other variables to 1.

Consider any of the clauses (a)–(f). For every literal in it, it is possible to find an assignment from the above list, which satisfies the chosen literal, but no other literal in the considered clause. Since the assignment satisfies the whole formula, the literal cannot be omitted without changing the represented function. This implies that the clauses (a)–(f) are prime implicates of f_G.

For the other direction let us start by examining the forward chaining closure of a set of variables S with respect to f_G which shall be denoted by $\mathrm{FC}_G(S)$. Given an arbitrary set S of variables of f_G, let us denote by $V_S = \{x_i \mid x_i \in S \text{ or } x_i \in e_{i,k} \text{ for some } e_{i,k} \in S\}$, i.e. V_S consists of those vertices which are either present in S directly, or they are incident to some edge, which belongs to S. By E_S let us denote the set of edges of G, whose both vertices belong to V_S. Now we claim that

$$\mathrm{FC}_G(S) = \begin{cases} V \cup E \cup \{z\} & \text{if } z \in S \\ V_S \cup E_S & \text{otherwise.} \end{cases}$$

Let us at first assume that $z \in S$. Then according to the fact that clauses in (e) and (f) are implicates of f_G, we can derive everything from z and therefore clearly $\mathrm{FC}_G(\{z\}) = \mathrm{FC}_G(S) = V \cup E \cup \{z\}$. Now let us assume that $z \notin S$. By using clauses from (a) and (b) we can observe that $V_S \cup E_S \subseteq \mathrm{FC}_G(S)$. By definition of $V_S \cup E_S$ we can see that $S \subseteq V_S \cup E_S$. Let C be a clause in $\mathcal{F}_G$ and let us assume that all its subgoals are contained in $V_S \cup E_S$. We shall show that in this case also its head belongs to $V_S \cup E_S$, and thus it follows that $\mathrm{FC}_G(S) = V_S \cup E_S$. Let us at first assume that C is of type (a), i.e. $C = (\overline{e_{i,j}} \vee x_i)$. In this case $e_{i,j} \in E_S$ and therefore by definition of E_S, we have that $x_i \in V_S$. Now, let us assume that C is of type (b), i.e. $C = (\overline{x_i} \vee \overline{x_j} \vee e_{i,j})$ for some edge $e_{i,j} \in E$. In this case $x_i, x_j \in V_S$ and hence also $e_{i,j} \in E_S$.

Now let us assume that $C = (S \vee y)$ is an implicate of f_G, which is nontrivial, i.e. $y \notin S$. If $z \in S$, then z itself is sufficient for deriving anything and therefore if C should be prime, then $S = \{z\}$, $y \in V \cup E$ and C has the form of (e) or (f). If $z \notin S$, then since C is an implicate of f_G, we get that $y \in \mathrm{FC}_G(S) \subseteq V_S \cup E_S$. If $y = x_i \in V_S$, then since $y \notin S$ it must be the case that $e_{i,j} \in S$ for some edge $e_{i,j} \in E$ incident to x_i. If C should be prime, then we must have $S = \{e_{i,j}\}$ and C has the form of (a). If on the other hand $y = e_{i,k} \in E_S$, then we have three possibilities.

1. $x_i, x_k \in S$. In this case, if C is prime, then $S = \{x_i, x_k\}$ and C has the form of (b).
2. $e_{i,j}, x_k \in S$ for some $e_{i,j} \in E$ or $e_{j,k}, x_i \in S$ for some $e_{j,k} \in E$. In the former case, if C is prime, we have that $S = \{e_{i,j}, x_k\}$ and C has the form of (c). The latter case is symmetric.
3. $e_{i,j}, e_{k,l} \in S$ for some $e_{i,j}, e_{k,l} \in E$ and then if C is prime, we must have that $S = \{e_{i,j}, e_{k,l}\}$ and C has the form of (d).

By this we have shown that every prime implicate of f_G must have the form of one of the cases (a)–(f) in the proposition of the lemma. □

Let us denote the size of the largest independent set of the undirected graph G by $\alpha(G)$, then we claim that the following holds.

Theorem 7.2. *Let $G = (V, E)$ be an undirected graph, then $\alpha(G) = \mathrm{ess}(f_G) - 3m$, where $m = |E|$.*

Proof. Let us at first assume that we have an independent set I of G of size $\alpha(G)$. We shall define three sets of $(m+n+1)$-bit vectors, which define pairwise disjoint essential sets.

1. Given an edge $e_{i,j}$ and a vertex $x_i \in e_{i,j}$ we define vector $t^i_{i,j}$
   - $t^i_{i,j}[x_j] = 1$,
   - $t^i_{i,j}[x_k] = 0$ for $x_k \in V \setminus \{x_j\}$,
   - $t^i_{i,j}[e_{i,j}] = 1$,
   - $t^i_{i,j}[e_{k,l}] = 0$ for $e_{k,l} \in E \setminus \{e_{i,j}\}$, and
   - $t^i_{i,j}[z] = 0$.
2. Given an edge $e_{i,j} \in E$, we define vector $t_{i,j}$
   - $t_{i,j}[x_i] = t_{i,j}[x_j] = 1$,
   - $t_{i,j}[x_k] = 0$ for $x_k \in V \setminus \{x_i, x_j\}$,
   - $t_{i,j}[e_{k,l}] = 0$ for $e_{k,l} \in E$ (including $e_{i,j}$), and
   - $t_{i,j}[z] = 0$.
3. Given $x_a \in I$ we define vector t_a as follows.
   - $t_a[x_a] = 0$,
   - $t_a[x_j] = 1$ for $x_j \in V \setminus \{x_a\}$,
   - $t_a[e_{a,k}] = 0$ for $e_{a,k} \in E$,
   - $t_a[e_{j,k}] = 1$ for $e_{j,k} \in E$ where $x_a \notin e_{j,k}$, and
   - $t_a[z] = 1$.

Let us prove that the prime essential sets defined using these falsepoints are

$$\mathcal{E}^p(t^i_{i,j}) = \{(\overline{e_{i,j}} \vee x_i)\}, \tag{5}$$

$$\mathcal{E}^p(t_{i,j}) = \{(\overline{x_i} \vee \overline{x_j} \vee e_{i,j})\}, \quad \text{and} \tag{6}$$

$$\mathcal{E}^p(t_a) = \{(\overline{z} \vee x_a)\} \cup \{(\overline{z} \vee e_{a,j}) \mid e_{a,j} \in E\} \tag{7}$$

and, in particular, these essential sets are disjoint.

- Let us at first consider vector $t^i_{i,j}$ for an arbitrary edge $e_{i,j} \in E$. Clearly $(\overline{e_{i,j}} \vee x_i) \in \mathcal{E}^p(t^i_{i,j})$. On the other hand, let C be an implicate of f_G for which $C(t^i_{i,j}) = 0$, then the subgoals of C may contain only x_j and $e_{i,j}$ (because these are the only bits set to 1 in $t^i_{i,j}$). According to Lemma 7.1, this condition is satisfied by the prime implicates $(\overline{e_{i,j}} \vee x_j)$ and $(\overline{e_{i,j}} \vee x_i)$. The former is not falsified by $t^i_{i,j}$, therefore the latter is the only prime implicate which belongs to $\mathcal{E}^p(t^i_{i,j})$.
- Now let us consider vector $t_{i,j}$ for an arbitrary $e_{i,j} \in E$. Clearly $(\overline{x_i} \vee \overline{x_j} \vee e_{i,j}) \in \mathcal{E}^p(t_{i,j})$. On the other hand, let C be an implicate of f_G, for which $C(t_{i,j}) = 0$, then the subgoals of C may contain only x_i and x_j (because these are the only bits set to 1 by $t_{i,j}$), According to Lemma 7.1, the only prime implicate satisfying this condition is $(\overline{x_i} \vee \overline{x_j} \vee e_{i,j})$ and it is also the only implicate belonging to $\mathcal{E}^p(t_{i,j})$.

- Now let us consider an arbitrary $x_a \in I$ and the corresponding vector t_a. The fact that $\{(\overline{z} \vee x_a)\} \cup \{(\overline{z} \vee e_{a,j}) \mid e_{a,j} \in E\} \subseteq \mathcal{E}^p(t_a)$ follows from the definition of t_a and it should also be clear that every clause with subgoal z, which belongs to $\mathcal{E}^p(t_a)$, is contained in the left-hand side of the above set inclusion. It therefore remains to show that every implicate C, which does not contain z as a subgoal, evaluates to 1 on t_a. Since C does not contain z as a subgoal, it must have the form (a)–(c), or (d) from the proposition of Lemma 7.1. If $C = (\overline{e_{i,j}} \vee x_i)$ for some $e_{i,j} \in E$ (case (a)), then $C(t_a) = 1$ because if $t_a[x_i] = 0$, then $i = a$ and $t_a[e_{i,j}] = 0$. If $C = (\overline{x_i} \vee \overline{x_j} \vee e_{i,j})$ for some edge $e_{i,j} \in E$ (case (b)), then $C(t_a) = 1$, because if x_i and x_j are both set to 1 by t_a, then also $e_{i,j}$ is set to 1. If $C = (\overline{e_{i,j}} \vee \overline{x_k} \vee e_{i,k})$ for some edges $e_{i,j}, e_{i,k}$ (case (c)), then either $a \in \{i, j, k\}$ in which case one of $e_{i,j}$ or x_k is set to 0 by t_a, or $a \notin \{i, j, k\}$ and $t_a[e_{i,k}] = 1$, therefore also in this case $C(t_a) = 1$. If $C = (\overline{e_{i,j}} \vee \overline{e_{k,l}} \vee e_{i,k})$ for some edges $e_{i,j}, e_{k,l}, e_{i,k} \in E$ (case (d)), then either $a \in \{i, j, k, l\}$, in which case $t_a[e_{i,j}] = 0$ or $t_a[e_{k,l}] = 0$, or $a \notin \{i, j, k, l\}$, in which case $t_a[e_{i,k}] = 1$. According to Lemma 7.1, there are no other prime clauses, which could belong to $\mathcal{E}^p(t_a)$.

The above shows that the sets of type $\mathcal{E}(t^i_{i,j})$ and $\mathcal{E}(t_{i,j})$ are pairwise disjoint and that they are disjoint with sets of type $\mathcal{E}(t_a)$. It remains to show that given two different $x_a, x_b \in I$, the sets $\mathcal{E}(t_a)$ and $\mathcal{E}(t_b)$ are disjoint. A clause $(\overline{z} \vee x_a) \notin \mathcal{E}(t_b)$ and similarly $(\overline{z} \vee x_b) \notin \mathcal{E}(t_a)$, therefore if there is a clause in $\mathcal{E}(t_a) \cap \mathcal{E}(t_b)$, then it is the clause $(\overline{z} \vee e_{a,b})$. However, this clause is not a prime implicate by Lemma 7.1, because I is an independent set and hence $\{x_a, x_b\} \notin E$, which means that $e_{a,b}$ does not appear as a variable in $\mathcal{F}_G$. The number of pairwise disjoint essential sets we have found is $3|E| + |I| = 3m + \alpha(G)$. This implies that $\mathrm{ess}(f_G) \geq 3m + \alpha(G)$.

Now let us assume that we have $\mathrm{ess}(f_G)$ pairwise disjoint FE sets of f_G. The above construction implies that $\mathrm{ess}(f_G) \geq 3m$. We shall construct an independent set I of size $k = \mathrm{ess}(f_G) - 3m \geq 0$. Let the falsepoints defining the $\mathrm{ess}(f_G)$ pairwise disjoint sets be denoted by $s_1, \ldots, s_{\mathrm{ess}(f_G)}$ and assume without loss of generality that each $\mathcal{E}^p(s_i)$ is minimal. Prime implicates C, which have the form (a) or (b) of the proposition of Lemma 7.1, form themselves singleton prime essential sets, which are also minimal essential sets. Hence, for every such C, we have some i for which $\mathcal{E}^p(s_i) = \{C\}$, otherwise, we could find a larger collection of pairwise disjoint essential sets by adding $\{C\}$ to it. Let us assume that $s_1, \ldots, s_{3m}$ correspond to these singleton sets. Therefore, $s_{3m+1}, \ldots, s_{3m+k}$ are falsepoints which define the remaining k essential sets. Let us inspect the set $\mathcal{E}^p(s_{3m+w})$ for an arbitrary $w \in \{1, \ldots, k\}$. The set $\mathcal{E}^p(s_{3m+w})$ must have a nonempty intersection with $\mathcal{F}_G$ by Theorem 5.8, but it cannot contain clauses of the form (a) or (b) from the proposition of Lemma 7.1, because $\mathcal{E}^p(s_{3m+w})$ is disjoint with $\mathcal{E}^p(s_i)$ for every $i \in \{1, \ldots, 3m\}$. Hence, it must contain a clause $(\overline{z} \vee x_i)$ for some variable $x_i \in V$. Let us associate with every vector s_{3m+w} one of these variables and let us denote it by x_{i_w}. We set $I = \{x_{i_1}, \ldots, x_{i_k}\}$ and claim that I is an independent set of G. Let $w, y \in \{1, \ldots, k\}$ be two arbitrary, but distinct indices and x_{i_w}, x_{i_y} their associated variables. Let us assume by contradiction that $e_{i_w,i_y} = \{x_{i_w}, x_{i_y}\} \in E$. Since $(\overline{z} \vee x_{i_w}) \in \mathcal{E}^p(s_{3m+w})$ and $(\overline{z} \vee x_{i_y}) \in \mathcal{E}^p(s_{3m+y})$, we have $s_{3m+w}[z] = s_{3m+y}[z] = 1$ and $s_{3m+w}[x_{i_w}] = s_{3m+y}[x_{i_y}] = 0$. Since $\mathcal{E}^p(s_{3m+w}) \cap \mathcal{E}^p(s_{3m+y}) = \emptyset$, we also have $s_{3m+w}[x_{i_y}] = s_{3m+y}[x_{i_w}] = 1$. Moreover, at least one of $s_{3m+w}[e_{i_w,i_y}]$ and $s_{3m+y}[e_{i_w,i_y}]$ must be 1, otherwise $(\overline{z} \vee e_{i_w,i_y}) \in \mathcal{E}^p(s_{3m+w}) \cap \mathcal{E}^p(s_{3m+y})$. But if $s_{3m+y}[e_{i_w,i_y}] = 1$, then $(\overline{e_{i_w,i_y}} \vee x_{i_y})$ evaluates to 0 on s_{3m+y}, which is a contradiction to the disjointness of $\mathcal{E}^p(s_{3m+y})$ and $\{(\overline{e_{i_w,i_y}} \vee x_{i_y})\}$ included among $\mathcal{E}^p(s_1), \ldots, \mathcal{E}^p(s_{3m})$. Similarly, if $s_{3m+w}[e_{i_w,i_y}] = 1$, then $(\overline{e_{i_w,i_y}} \vee x_{i_w})$ evaluates to 0 on s_{3m+w}, which is a contradiction to the disjointness of $\mathcal{E}^p(s_{3m+w})$ and $\{(\overline{e_{i_w,i_y}} \vee x_{i_w})\}$. Therefore x_{i_w}, x_{i_y} cannot form an edge of G. By this we have shown that I is an independent set of G of size $|I| = k \leq \alpha(G)$ and, hence, $\mathrm{ess}(f_G) = 3m + k \leq 3m + \alpha(G)$.

The first and the second half of the proof together imply that $\mathrm{ess}(f_G) = 3m + \alpha(G)$. □

The fact that the problem ESS-Horn-3CNF belongs to *NP* follows directly from Theorem 3.8 and we may therefore conclude the following.

Corollary 7.3. *The problem ESS-Horn-3CNF is NP-complete.*

It is also worth to note that while computing $\mathrm{ess}(f_G)$ is NP-hard (equivalent to computing $\alpha(G)$), computing $\mathrm{cnf}(f_G)$ can be done in polynomial time. As was shown in [2], computing $\mathrm{cnf}(f_G)$ is equivalent to computing the size of a minimum edge cover of G, which is long known to be in P.

8. Computing ess(f) and its relaxation from the truth table of f

In this section we first prove *NP*-completeness of the following problem.

Problem *ESS-TT*(f, k).
Input: A Boolean function f represented by its truth table and an integer $k \geq 0$.
Question: Is $\mathrm{ess}(f) \geq k$?

Minimization of DNF for a Boolean function given by its truth table is proved to be *NP*-hard in [1] using a reduction from 3-*Partite Set Cover*. We use essentially the same reduction, although we use it as a reduction of the problem 3-*PARTITE-TRIANG-INDSET* described below to *ESS-TT*. An instance of the input problem is a 3-uniform hypergraph $\mathcal{H} = (V, S)$, whose set of edges S is a subset of $U_1 \times U_2 \times U_3$ for some pairwise disjoint sets of vertices U_1, U_2, and U_3. We consider this hypergraph as a representation of an ordinary graph, which is a union of a set of 3-partite triangles. Namely, for $\mathcal{H}$ as above, we define $\mathcal{G}(\mathcal{H})$ as a graph on the set of vertices V and whose edges are all two-element subsets of the edges in S. Formally, we define problem 3-*PARTITE-TRIANG-INDSET*($\mathcal{H}$, k) as follows:

Problem 3-*PARTITE-TRIANG-INDSET*$(\mathcal{H}, k)$.
Input: A hypergraph $\mathcal{H} = (U, \mathcal{S})$, where $U = U_1 \cup U_2 \cup U_3$ and $\mathcal{S} \subseteq U_1 \times U_2 \times U_3$ for some pairwise disjoint sets of vertices U_1, U_2, and U_3, and an integer $k \geq 0$.
Question: Is there an independent set of vertices $I \subseteq U$ in $\mathcal{G}(\mathcal{H})$, $|I| \geq k$?

Note that the instance of 3-*PARTITE-TRIANG-INDSET* is the same as in 3-*Partite Set Cover*, the only difference is in the question, which we ask about the input hypergraph. We shall start by proving that the problem we have just defined is *NP*-complete.

Theorem 8.1. $3 - \text{PARTITE} - \text{TRIANG} - \text{INDSET}(\mathcal{H}, k)$ *problem is NP-complete.*

Proof. The problem is clearly in *NP*, since an independent set I of at most k vertices can serve as a polynomially verifiable certificate of a positive answer. We prove that it is *NP*-complete using a reduction from the maximum independent set problem restricted to instances $(G = (V, E), k)$, where G is a graph with no isolated vertices satisfying $|E| \geq |V|$ and $k \geq 2$. In order to see that this problem is *NP*-complete, consider an unrestricted instance of maximum independent set problem. If we eliminate all isolated vertices and decrease the size bound for the independent set accordingly, we get an equivalent instance. If we add a vertex connected to all vertices of the original graph, we do not change the size of the maximum independent set, but we get a graph, which has at least so many edges as vertices. The assumption that $k \geq 2$ does not change the *NP*-complete status of the problem, since the instances with $k \leq 1$ are trivial.

In order to reduce the problem from the previous paragraph into 3-*PARTITE-TRIANG-INDSET* problem, we construct a 3-partite graph G', using a reduction from [16]. Namely, construct $G' = (V', E')$ from G by replacing each edge by a path of length 3 and consider the instance $(G', k + |E|)$ of the maximum independent set problem. Note that V' consists of the original vertices and of $2|E|$ new vertices, which are internal vertices of the paths replacing original edges. Let V_2 be a set of the new nodes containing one of the two nodes from each of these paths chosen arbitrarily. Let V_3 be the set of the remaining new nodes. For simplicity of notation, let us denote $V_1 = V$. Then, we have $V' = V_1 \cup V_2 \cup V_3$ and these three sets form the partitions of the 3-partite graph G'. Since $|V| \leq |E|$, we have $|V_1| \leq |V_2| = |V_3| = |E|$.

Using the argument from [16], G contains an independent set of size k if and only if G' contains an independent set of size $k + |E|$.

Let $G'' = (V'', E'')$ be obtained from G' by adding three nodes u_1, u_2, u_3 and all edges between u_i and all vertices of V_j, where $i \neq j$. Note that G'' is 3-partite with the partitions $V_i \cup \{u_i\}$ for $i = 1, 2, 3$. Consider the instance of maximal independent set $(G'', k + |E|)$. We will prove that it is equivalent to the instance $(G', k + |E|)$. The independent sets of G'' are of two types. If it contains one of the nodes u_i, then it is a subset of $V_i \cup \{u_i\}$. If it does not contain any of the nodes u_i, then it is an independent set in G'. Now note that the independent sets contained in $V_i \cup \{u_i\}$ have size smaller than $k + |E|$, since $|V_1| \leq |V_2| = |V_3| = |E|$ and $k \geq 2$.

Given 3-partite graph G'', it is easy to construct a hypergraph $\mathcal{H}$ such that $G'' = \mathcal{G}(\mathcal{H})$. Namely, let

$$U_i = V_i \cup \{u_i\}, \quad i = 1, 2, 3,$$

and

$$\mathcal{S} = \{\{v_i, v_j, u_k\} \mid \{v_i, v_j\} \in E' \text{ and } \{i, j, k\} = \{1, 2, 3\}\}.$$

The sets U_i are clearly disjoint. Since, $G'' = \mathcal{G}(\mathcal{H})$, the instance $(G'', k + |E|)$ of maximum independent set is equivalent to the instance $(\mathcal{H}, k + |E|)$ of 3-*PARTITE-TRIANG-INDSET*. □

Now we shall describe a reduction of an instance $(\mathcal{H}, k)$ of 3-*PARTITE-TRIANG-INDSET* to an instance of *ESS-TT*, i.e. to a Boolean function f. The construction we use here is the same as the one used in [1] to transform an instance of 3-Partite Set Cover to a minimization of DNF for a Boolean function given by its truth table. Here, we use the fact that an instance of 3-*PARTITE-TRIANG-INDSET* is described by the same hypergraph as an instance of 3-Partite Set Cover, use the same transformation to a Boolean function and consider its negation, since we are dealing with CNFs instead of DNFs. However, finally, we ask a different question about the constructed Boolean function, which is equivalent to a question on the input instance considered as a 3-*PARTITE-TRIANG-INDSET* and not a 3-Partite Set Cover instance. Since the proof of the correctness of the transformation for our purpose is different from the one in [1], we describe the transformation here in full detail.

Let $\mathcal{H} = (U_1 \cup U_2 \cup U_3, \mathcal{S})$ be an arbitrary hypergraph, where $\mathcal{S} \subseteq U_1 \times U_2 \times U_3$ for some pairwise disjoint sets of vertices U_1, U_2, and U_3. We shall describe, how to associate a Boolean function $f_{\mathcal{H}}$ with $\mathcal{H}$ in the same way as it was done in [1].

Let $n = \max\{|U_1|, |U_2|, |U_3|\}$, let q be the smallest integer satisfying $\binom{q}{q/2} \geq n$, and let $t = 3q$. Note that the fact that q is the smallest integer with the required property implies that $q = O(\log n)$. Let $b(j)$ for $j = 1, \ldots, n$ be distinct vectors from $\{0, 1\}^q$ each of which contains exactly $q/2$ ones. Then, let $V \subseteq \{0, 1\}^t$ be such that it contains encodings of the elements of $U_1 \cup U_2 \cup U_3$ defined as follows. The jth element u of U_i, where $j = 1, \ldots, |U_i|$, is encoded by a vector $e(u)$ consisting of three blocks of length q, ith of which is $b(j)$ and the remaining two blocks consist of q zeros.

For each $A \in \mathcal{S}$, let its encoding $e(A)$ be the bitwise disjunction of the encodings of the three elements of A in V. Note that the construction of the encodings guarantees that different sets A correspond to incomparable vectors in $\{0, 1\}^t$, since the sets differ in at least one of their elements and the corresponding blocks of length q are incomparable. Let $W = \{e(A) \mid A \in \mathcal{S}\}$. The following lemma was proved in [1].

Lemma 8.2 (*[1], First Part of Lemma 3.1*). *For each $A \in \mathcal{S}$ and each $u \in U_1 \cup U_2 \cup U_3$, we have*

$$u \in A \Leftrightarrow e(u) \leq e(A).$$

Let $R = \{x \in \{0,1\}^t \mid x \notin V \text{ and for some } w \in W, x \leq w\}$. Let g be a partial function with the domain $\{0,1\}^t$ such that $g(x) = 0$ if $x \in V$, $g(x) = *$ if $x \in R$, and $g(x) = 1$ otherwise. We shall finish the transformation by reduction of the partial function g of the variables $x \in \{0,1\}^t$ to the total function $f_{\mathcal{H}}$ of the variables $(x, y_1, y_2) \in \{0,1\}^{t+2}$

$$f_{\mathcal{H}}(x, y_1, y_2) = \begin{cases} 0, & \text{if } g(x) = 0 \text{ and } y_1 = y_2 = 1 \\ 0, & \text{if } g(x) = * \text{ and } y_1 = y_2 = 1 \\ 0, & \text{if } g(x) = * \text{ and } y_1 = p(x), \text{ and } y_2 = \neg p(x) \\ 1, & \text{otherwise} \end{cases}$$

where $p(x)$ is the parity of x, i.e. the sum of the bits in x mod 2.

As we have already mentioned, the construction of the function $f_{\mathcal{H}}$ is exactly the same as described in [1] (see also [7]) with the only difference caused by using CNFs instead of DNFs—we had to negate the final function $f_{\mathcal{H}}$. Now we shall show that $\text{ess}(f_{\mathcal{H}}) \geq |R| + k$ if and only if the graph $\mathcal{G}(\mathcal{H})$ has an independent set of size k.

Theorem 8.3. *Let $\mathcal{H} = (U = U_1 \cup U_2 \cup U_3, \mathcal{S} \subseteq U_1 \times U_2 \times U_3)$, where U_1, U_2, U_3 are pairwise disjoint sets of vertices, let $f_{\mathcal{H}}$ be its associated Boolean function, let R be the set of inputs defined during its construction, and let k be an arbitrary integer. Then $\text{ess}(f_{\mathcal{H}}) \geq |R| + k$ if and only if the hypergraph $\mathcal{H}$ has an independent set of size k.*

Proof. Let us start by description of the list of all prime implicates of $f_{\mathcal{H}}$. For every $x \in R$, consider the clause, which is 0 on the two points $(x, p(x), \neg p(x))$ and $(x, 1, 1)$. This is a prime implicate, since $(x, 1, 1)$ is the only neighbor of $(x, p(x), \neg p(x))$, where $f_{\mathcal{H}}$ is 0. Moreover, these are the only prime implicates, which are zero on some of the points with $y_1 = 0$ or $y_2 = 0$. Hence, all the remaining prime implicates are zero only on the points $(z, 1, 1)$ for some $z \in \{0,1\}^t$. Since $f_{\mathcal{H}}(z, 1, 1) = 0$ if and only if $z \leq w$ for some $w \in W$ and the elements of W are pairwise incomparable, it is easy to verify that all the remaining prime implicates of $f_{\mathcal{H}}$ may be obtained in such a way that for any $w \in W$, we consider the clause, which is 0 exactly on the vectors $(z, 1, 1)$, where $z \leq w$ for the given w.

Now, assume that I is an independent set in $\mathcal{G}(\mathcal{H})$ of size k and let us construct a set of $|R| + k$ essential sets of $f_{\mathcal{H}}$, which are pairwise disjoint. Consider the prime essential sets $\mathcal{E}^p((x, p(x), \neg p(x)))$ for $x \in R$ and $\mathcal{E}^p(e(v))$ for $v \in I$. Prime essential sets of the former type contain a single clause, which is not contained in any other prime essential set from the presented list. Hence, these essential sets are disjoint with all the others. Consider prime essential sets $\mathcal{E}^p(e(v_1))$ and $\mathcal{E}^p(e(v_2))$ for different points $v_1, v_2 \in I$. If these two essential sets are not disjoint, then there is a vector $w \in W$ such that $v_1 \leq w$ and $v_2 \leq w$. By the definition of W, this implies that there is $A \in S$ such that $v_1, v_2 \in A$ and so, (v_1, v_2) is an edge of $\mathcal{G}(\mathcal{H})$. This is not possible, since I is an independent set. Hence, the presented $|R| + k$ essential sets are indeed disjoint.

For the opposite direction, assume that Z is a set of vectors in $\{0,1\}^{t+2}$ such that $|Z| \geq |R| + k$ and the prime essential sets $\mathcal{E}^p(z)$ for all $z \in Z$ are pairwise disjoint. If some of the points $(x, p(x), \neg p(x))$ is not in Z, then modify Z by including this point to Z and removing the point $(x, 1, 1)$ from Z, if it is there. The size of Z does not decrease and one may verify that the points from the modified Z still define disjoint prime essential sets. Now, note that except of $|R|$ points $(x, p(x), \neg p(x))$, Z contains only points from V. Hence, there are at least k points $v \in V$, such that for every $w \in W$, at most one of them satisfies $v \leq w$. These k vectors from V are encodings of k vertices of $\mathcal{H}$, no two of which belong to the same set $A \in \mathcal{S}$. It follows that no two of these points are connected by an edge in $\mathcal{G}(\mathcal{H})$ and so, $\mathcal{G}(\mathcal{H})$ contains an independent set of size k. □

As a corollary we now obtain the following.

Theorem 8.4. *The problem to determine, whether $\text{ess}(f) \geq k$ for a function f defined by its truth table, i.e. the problem* ESS − TT*(f, k), is NP-complete.*

Proof. The fact that *ESS-TT* belongs to *NP* can be easily observed and it also follows from Theorem 3.8. *NP*-hardness of *ESS-TT* follows from Theorem 8.1, construction of $f_{\mathcal{H}}$ described in this section, and Theorem 8.3. □

8.1. Relaxation of ess(f) *for functions given by their truth table*

Computing $\text{ess}(f)$ for functions given by their truth table is intractable by Theorem 8.4. On the other hand, it appears that a relaxation of $\text{ess}(f)$ may be computed more efficiently under the same conditions, namely that the truth table of f is given as the input.

Definition 8.5. For every Boolean function f, let $lp(f)$ be the maximum of

$$\sum_{x \in \{0,1\}^n : f(x) = 0} w(x),$$

(over all possible choices of weights w) where $w(x)$ is a nonnegative real number assigned to every falsepoint x of f and such that for every prime implicant C of f, the inequality

$$\sum_{x\in\{0,1\}^n:C(x)=0} w(x) \leq 1$$

is satisfied.

Theorem 8.6. *For every Boolean function f, we have*

$$\mathrm{cnf}(f) \geq lp(f) \geq \mathrm{ess}(f).$$

Proof. Let $\mathcal{F}$ be a set of prime implicates, which form a minimal CNF representation and let $w(x)$ be an assignment of the weights, on which the value $lp(f)$ is achieved in Definition 8.5. Since $f = \bigwedge_{C\in\mathcal{F}} C$, we have also

$$\sum_{x\in\{0,1\}^n:f(x)=0} w(x) \leq \sum_{C\in\mathcal{F}} \sum_{x\in\{0,1\}^n:C(x)=0} w(x) \leq |\mathcal{F}|.$$

Since $|\mathcal{F}| = \mathrm{cnf}(f)$ and $\sum_{x\in\{0,1\}^n:f(x)=0} w(x) = lp(f)$, we have $\mathrm{cnf}(f) \geq lp(f)$.

Let $\mathcal{T}$ be a set of $\mathrm{ess}(f)$ falsepoints t, for which the sets $\mathcal{E}(t)$ are pairwise disjoint. Let $w(t) = 1$ for $t \in \mathcal{T}$ and $w(t) = 0$ otherwise. If C is a prime implicate, then it belongs to at most one $\mathcal{E}(t)$, $t \in \mathcal{T}$, and thus there is at most one falsepoint t falsifying C for which $w(x) = 1$. Hence, the weights $w(x)$ satisfy the condition in Definition 8.5. Since $lp(f)$ is a maximum over all possible weights satisfying this condition, we have

$$lp(f) \geq \sum_{x\in\{0,1\}^n:f(x)=0} w(x) = |\mathcal{T}| = \mathrm{ess}(f). \quad \square$$

Let us verify that the linear programming problem corresponding to computing $lp(f)$ has size polynomial in the size of the table of the function f. The variables of the corresponding LP problem are the weights $w(x)$ for all falsepoints of f. If the function has n variables, then the size of the table is 2^n and this is clearly an upper bound on the number of falsepoints. Moreover, the largest set of constraints correspond to prime implicates. Since there are at most $3^n = (2^n)^{\log_2 3}$ clauses on n variables, the number of prime implicates is also bounded by a polynomial in the table size and they can be found in polynomial time. Consequently, the problem can be solved, e.g. by Karmarkar's algorithm [14], in time polynomial in the size of the table of f and the number of bits of the precision of the representation of the numbers used in the computation.

9. Conclusion

In this paper we have studied a lower bound on the minimum CNF size of a given function f represented by a CNF φ. The lower bound which we have considered is given by $\mathrm{ess}(f)$, which denotes the number of pairwise disjoint essential sets of implicates of f. We are mainly interested in functions for which this lower bound matches the minimum CNF size. We have called such functions coverable, and we have shown in Sections 3 and 4 that if a class of Boolean function $\mathcal{C}$ is tractable and coverable, then the problem of minimization of functions in this class belongs to both *NP* and *co-NP*. This fact proves that such a minimization problem is not NP-hard (unless $NP = co\text{-}NP$), and thus indicates that it might in fact belong to P. In Section 5 we study the intersections of essential sets with the set of prime implicates, call these intersections prime essential sets and prove that many properties of essential sets carry over to prime essential sets. This fact allows us to restrict our attention to prime implicates only.

We have also proved several negative results about $\mathrm{ess}(f)$. In Section 6 we have shown that not every function is coverable and moreover for every constant k we can construct a function f, for which $\mathrm{cnf}(f)/\mathrm{ess}(f) \geq k$. In Section 7 we have shown that the problem of checking whether $\mathrm{ess}(f) \geq k$ is NP-complete if its input is a pure Horn 3CNF. In Section 8 we have shown that this problem remains NP-complete even in the case when the input is allowed to be much larger, namely when the input function is represented by a truth table. On the other hand we have shown that a relaxed value of $\mathrm{ess}(f)$ can be computed using linear programming.

Given the fact that minimization seems to be easier for coverable functions than in the general case, one might ask, whether it would be possible to check in polynomial time, if a given function f is coverable, i.e. whether $\mathrm{ess}(f) = \mathrm{cnf}(f)$. Unfortunately, it turns out that this problem is NP-complete even if the input is a pure Horn 3CNF. This result is not contained in the present paper, because we have found the corresponding reduction just recently. On the other hand, all classes for which a polynomial time minimization algorithm is known to us are coverable. This gives an indication that if a tractable class of Boolean functions is found to be coverable, then we can hope for a polynomial minimization algorithm for it. From theoretical point of view, it would be therefore interesting to find a class of Boolean functions which would be tractable and coverable, and yet we would not be able to find a polynomial minimization algorithm for it.

Acknowledgments

Petr Savicky was supported by grant number 1M0545 (MŠMT ČR) and by Institutional Research Plan AV0Z10300504. Petr Kučera and Ondřej Čepek gratefully acknowledge the support by the Czech Science Foundation (grants 201/07/P168 and

P202/10/1188). The authors would like to thank the two anonymous referees for their valuable comments which helped to improve the presentation of this paper.

References

[1] E. Allender, L. Hellerstein, P. McCabe, T. Pitassi, M. Saks, Minimizing DNF formulas and AC^0_d circuits given a truth table, in: Proceedings of the 21st Annual IEEE Conference on Computational Complexity, IEEE Computer Society, 2006, pp. 237–251.
[2] G. Ausiello, A. D'Atri, D. Sacca, Minimal representation of directed hypergraphs, SIAM Journal on Computing 15 (2) (1986) 418–431.
[3] E. Boros, O. Čepek, A. Kogan, P. Kučera, Exclusive and essential sets of implicates of Boolean functions, Discrete Applied Mathematics 158 (2) (2010) 81–96.
[4] H. Kleine Büning, T. Letterman, Propositional Logic: Deduction and Algorithms, Cambridge University Press, New York, NY, 1999.
[5] O. Čepek, P. Kučera, Disjoint essential sets of implicates of a CQ Horn function, in: Proceedings of 12th Czech-Japan Seminar on Data Analysis and Decision Making under Uncertainty, Litomyšl, Czech Republic, 2009, pp. 79–92.
[6] S.A. Cook, The complexity of theorem-proving procedures, in: STOC'71: Proceedings of the Third Annual ACM Symposium on Theory of Computing, New York, NY, 1971, pp. 151–158.
[7] S. Czort, The complexity of minimizing disjunctive normal form formulas, Master's Thesis, University of Aarhus, 1999.
[8] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, San Francisco, CA, 1979.
[9] M.R. Genesereth, N.J. Nilsson, Logical Foundations of Artificial Intelligence, Morgan Kaufmann, Los Altos, CA, 1987.
[10] P.L. Hammer, A. Kogan, Horn functions and their DNFs, Information Processing Letters 44 (1992) 23–29.
[11] P.L. Hammer, A. Kogan, Optimal compression of propositional horn knowledge bases: complexity and approximation, Artificial Intelligence 64 (1993) 131–145.
[12] P.L. Hammer, A. Kogan, Knowledge compression—logic minimization for expert systems, in: Proceedings of IISF/ACM Japan International Symposium, World Scientific, Tokyo, Singapore, 1994, pp. 306–312.
[13] P.L. Hammer, A. Kogan, Quasi-acyclic propositional horn knowledge bases: optimal compression, IEEE Transactions on Knowledge and Data Engineering 7 (5) (1995) 751–762.
[14] N. Karmarkar, A new polynomial time algorithm for linear programming, Combinatorica 4 (4) (1984) 373–395.
[15] D. Maier, Minimal covers in the relational database model, Journal of the ACM 27 (1980) 664–674.
[16] S. Poljak, A note on stable sets and colorings of graphs, Commentationes Mathematicae Universitatis Carolinae 15 (2) (1974) 307–309.
[17] W. Quine, The problem of simplifying the truth functions, American Mathematical Monthly 59 (1952) 521–531.
[18] W. Quine, A way to simplify truth functions, American Mathematical Monthly 62 (1955) 627–631.
[19] S.J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Pearson Education, 2003.
[20] C. Umans, The minimum equivalent DNF problem and shortest implicants, Journal of Computer and System Sciences 63 (4) (2001) 597–611.