The Complexity of Checking Redundancy of CNF Propositional Formulae

Paolo Liberatore[1]

Abstract. A knowledge base is redundant if it contains parts that can be inferred from the rest of it. We study the problem of checking whether a CNF formula (a set of clauses) is redundant, that is, it contains clauses that can be derived from the other ones. Any CNF formula can be made irredundant by deleting some of its clauses: what results is an irredundant equivalent subset (I.E.S.) We study the complexity of some related problems: verification, checking existence of a I.E.S. with a given size, checking necessary and possible presence of clauses in I.E.S.'s, and uniqueness.

1 INTRODUCTION

The problem of redundancy of knowledge bases is relevant to applications in which efficiency of entailment is important. Indeed, the size of a knowledge base is one of the factors that determine the speed of the inference process. While some theorem provers introduce a limited number of redundant formulae for speeding up solving, excessive redundancy can cause problems of storage, which in turns slows down reasoning. In particular, updates can increase the size of knowledge bases exponentially [3], and redundancy makes the problem of storing the knowledge base worst. Algorithms for checking redundancy of knowledge bases have been developed for the case of production rules [5, 14]. In this paper, we instead study redundancy of propositional knowledge base in CNF form, that is, checking whether a clause in a set is implied by the others.

A more general question that has been already investigated in the propositional case is whether a knowledge base is equivalent to a shorter one. This problem is called *minimization of propositional formulae*, and it has been one of the first to be analyzed from the point of view of computational complexity: its study begun in the paper that introduced the polynomial hierarchy [12]. A complexity characterization of this problem has been first given for Horn knowledge bases [11, 1, 9]; afterwards, the problem has been tackled again in the general case [10, 16]. While the Horn case is now quite understood (the problem is NP-complete, using several different notions of minimality), some problems regarding non-Horn formulae are still open.

Among the open problems is redundancy elimination, which is a weak form of formula minimization: if a set of clauses is redundant, it is not minimal, as some clauses can be removed from it while preserving equivalence. On the other hand, redundancy elimination only allows for removal of clauses, so it is not guaranteed to produce a minimal knowledge base. For example, $\{x \vee y, x \vee \neg y\}$ is irredundant, but is equivalent to a shorter set: $\{x\}$. A related problem, not analyzed in this paper, is that of removing redundancy from a single clause, that is, removing literals from clauses rather than removing clauses from sets. The computational analysis of this problem, and of related ones, has been done by Gottlob and Fermüller [8].

The problem of redundancy elimination is relevant for at least two reasons. First, it seems somehow easier to remove redundant clauses, rather than reshaping the whole knowledge base. Indeed, removing redundant clauses can be done by checking whether each clause can be inferred by the other ones, while finding a minimal equivalent formula involves a process of *guessing and checking* a whole knowledge base for equivalence. Even for short knowledge bases, the number of candidate equivalent knowledge bases is very high.

A second reason for preferring redundancy elimination to minimization is that the syntactic form in which a knowledge base is expressed can be important. For example, some semantics for knowledge base revision depend on the syntax of knowledge bases. If a knowledge base is replaced with an equivalent one, even a single update can lead to a completely different result [6, 13].

Problem	Complexity
Checking irredundancy	NP complete
A set is an I.E.S.	D^p complete
Existence of an I.E.S. of size $\leq k$	Σ_2^p complete
A clause is in all I.E.S.'s	NP complete
A clause is in an I.E.S.	Σ_2^p complete
Uniqueness of I.E.S.'s	$\Delta_2^p[\log n]$ complete

Table 1. The complexity results proved in this paper

Several problems are related to that of redundancy. The aim of checking redundancy is to end up with a subset of clauses that is both equivalent to the original one and irredundant. We call it an *irredundant equivalent subset* of the original set, or I.E.S. Note that an I.E.S. is a subset of the original set, and can therefore only contain clauses of the original set. This makes it different to a minimal equivalent *set*, which can instead be composed of arbitrary clauses.

The problems that are analyzed in this paper are: checking whether a set is an I.E.S.; checking the existence of an I.E.S. of size bounded by an integer k; deciding whether a clause is in some, or all, the I.E.S.'s; and checking uniqueness. Table 1 summarizes the results proved in this paper.

[1] Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", via Salaria 113, 00198, Rome, Italy.

2 DEFINITIONS

In this paper, we study irredundancy and equivalence of sets of propositional clauses. Namely, a set of clauses is irredundant if no clause can be removed from it without changing its sets of models.

Definition 1 *A set of clauses* Π *is irredundant if and only if* $\Pi \backslash \{\gamma_i\} \not\models \gamma_i$ *holds for any* $\gamma_i \in \Pi$.

A related definition is that of irredundant equivalent subset: given a set of clauses, we are interested in removing some redundant clauses from it, in such a way equivalence is preserved.

Definition 2 *A set of clauses* Π' *is said to be an* irredundant equivalent subset *(I.E.S.) of another set of clauses* Π *if and only if:*

1. $\Pi' \subseteq \Pi$
2. $\Pi' \models \Pi$
3. Π' *is irredundant*

An alternative definition is that an I.E.S. is an equivalent subset of the original set such that none of its subsets has the same properties. Any set of clauses has at least one I.E.S., but it may also have more than one of them, as shown by the following example.

Example 1 *Let* $\Pi = \{a \vee \neg b, \neg a \vee b, a \vee c, b \vee c\}$. *This set has two I.E.S.'s:*

$$\begin{aligned} \Pi_1 &= \Pi \backslash \{a \vee c\} \\ \Pi_2 &= \Pi \backslash \{b \vee c\} \end{aligned}$$

It is indeed easy to see that the first two clauses of Π *are equivalent to* $a \equiv b$*, which implies that* $a \vee c$ *and* $b \vee c$ *are equivalent. It is also easy to see that neither* $a \vee \neg b$ *nor* $\neg a \vee b$ *can be removed from* Π *while preserving equivalence with it.*

The set of clauses of this example can be used to show that a set of clauses may have exponentially many I.E.S.'s. Consider the set:

$$\Pi_n = \bigcup_{i=1,\ldots,n} \Pi[\{a/a_i, b/b_i, c/c_i\}]$$

In words, Π_n is made of n copies of Π, each built on its own set of three variables. While removing clauses from Π_n, we have n independent choices, one for each copy: for each i we can remove either $a_i \vee c_i$ or $b_i \vee c_i$. This proves that 2^n outcomes are possible, each leading to a different I.E.S.

In this paper, we assume that knowledge bases are sets of clauses. We sometimes use formulae like $a_1 \wedge \cdots \wedge a_m \rightarrow b$, which can be easily translated into the equivalent clauses $\neg a_1 \vee \cdots \vee \neg a_m \vee b$. We also assume that clauses are not tautological. Clearly, tautologies can be easily checked and removed, and do not change the complexity of the problems considered here.

3 COMPLEXITY OF CHECKING IRREDUNDANCY

We prove that checking whether a set of clauses is irredundant is NP-complete.

Theorem 1 *Checking irredundancy of a set of clauses is* NP*-complete.*

Proof. Membership: we have to check whether, for any $\gamma \in \Pi$, it holds $\Pi \backslash \{\gamma\} \not\models \gamma$. This can be done by guessing a model for each set $\Pi \backslash \{\gamma\} \cup \{\neg \gamma\}$, which shows the problem to be in NP.

Hardness is proved by showing that, given a set of non-tautological clauses $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$, the following set of clauses Π is irredundant if and only if Γ is satisfiable (we remind that $c_i \rightarrow \gamma_i$ is actually the clause $\neg c_i \vee \gamma_i$.)

$$\Pi = \{c_i \rightarrow \gamma_i \mid \gamma_i \in \Gamma\} \cup \{\neg c_1 \vee \cdots \vee \neg c_m \vee \neg a\}$$

$\{c_1, \ldots, c_m, a\}$ are new variables not appearing in Γ. First, we prove that no clause of the form $c_i \rightarrow \gamma_i$ is implied by the rest of Π, regardless of the satisfiability of Γ.

Consider the following model.

$$M = \{c_i\} \cup \{x_i \mid \neg x_i \in \gamma_i\}$$

M is not a model of $c_i \rightarrow \gamma_i$, as we assumed that no clause in Γ is tautological. On the converse, it is a model of $\Pi \backslash \{c_i \rightarrow \gamma_i\}$. This is an easy consequence of the fact that, for any $j \neq i$, we have that $c_j \notin M$, thus $M \models c_j \rightarrow \gamma_j$, and that $a \notin M$, thus $M \models \neg c_1 \vee \cdots \vee \neg c_m \vee \neg a$.

We now show that the unsatisfiability of Γ implies the redundancy of Π, and then we prove that the satisfiability of Γ implies the irredundancy of Π. Let us define:

$$\Pi_C = \{c_i \rightarrow \gamma_i \mid \gamma_i \in \Gamma\}$$

Γ **is unsatisfiable.** We prove that Π is redundant. Since Γ has no models, no model of Π_C contains all c_i's. As a result, $\Pi_C \models \neg c_1 \vee \cdots \vee \neg c_m$, which implies that $\Pi_C \models \neg c_1 \vee \cdots \vee \neg c_m \vee \neg a$, which in turns implies that Π is redundant.

Γ **is satisfiable.** We already proved that no clause $c_i \rightarrow \gamma_i$ can be derived from the other ones. What remains to prove is that $\Pi_C \not\models \neg c_1 \vee \cdots \vee \neg c_m \vee \neg a$. Since Γ is satisfiable, it has a model M. It is easy to see that $M \cup \{c_1, \ldots, c_m, a\}$ is a model of Π_C. Since this is not a model of $\neg c_1 \vee \cdots \vee \neg c_m \vee \neg a$, the claim follows. □

4 COMPLEXITY OF I.E.S.'S

In this section, we study some problems related to I.E.S.'s. Namely, we prove the complexity results shown from the second to the last row of Table 1.

Theorem 2 *Given two sets of clauses* Π *and* Π'*, checking whether* Π' *is an I.E.S. of* Π *is* D^p*-complete.*

Proof. Membership amounts to showing that $\Pi' \subseteq \Pi$ (a polynomial task), that $\Pi' \models \Pi$ (which is in coNP) and that Π' is irredundant (which we proved to be in NP). Therefore, the problem is in D^p.

Hardness is proved by reduction from the sat-unsat problem: given a pair of sets of clauses $\langle \Gamma, \Sigma \rangle$, check whether the first one is satisfiable while the second one is not. This problem is D^p-complete even if Γ and Σ do not share variables [2], which we assume. Reduction is as follows:

$$\begin{array}{rcl} \Pi & = & \{c_i \rightarrow \gamma_i \mid \gamma_i \in \Gamma\} \cup \{\neg c_1 \vee \cdots \vee \neg c_m \vee \neg a\} \cup \\ & & \{d_i \rightarrow \gamma_i \mid \gamma_i \in \Sigma\} \cup \{\neg d_1 \vee \cdots \vee \neg d_r \vee \neg e\} \\ \Pi' & = & \{c_i \rightarrow \gamma_i \mid \gamma_i \in \Gamma\} \cup \{\neg c_1 \vee \cdots \vee \neg c_m \vee \neg a\} \cup \\ & & \{d_i \rightarrow \gamma_i \mid \gamma_i \in \Sigma\} \end{array}$$

First, we show that Π' is irredundant if and only if Γ is satisfiable. From the proof of Theorem 1, $\{c_i \rightarrow \gamma_i \mid \gamma_i \in \Gamma\} \cup \{\neg c_1 \vee \cdots \vee \neg c_m \vee \neg a\}$ is irredundant if and only if Γ is satisfiable, and $\{d_i \rightarrow \gamma_i \mid \gamma_i \in \Sigma\}$ is always irredundant. Since these two subsets of Π' do not share variables, Π' is irredundant if and only if both parts are.

What remains to prove is only that $\Pi' \models \Pi$ if and only if Σ is unsatisfiable. This is an easy consequence of the fact that $\{d_i \rightarrow \gamma_i \mid \gamma_i \in \Sigma\}$ implies $\neg d_1 \vee \cdots \vee \neg d_r \vee \neg e$ if and only if Σ is unsatisfiable. □

The above theorem shows how hard it is to decide whether a specific subset is an I.E.S. However, it does not tell how hard it is to find an I.E.S. We now consider a problem related to finding the *size* of minimal I.E.S.'s, namely, the problem of deciding whether a set has an I.E.S. of size bounded by an integer k.

Theorem 3 *Given a set of clauses Π and an integer k, deciding whether Π has an I.E.S. of size at most k is Σ_2^p-complete.*

Proof. Membership: the problem amounts to deciding whether there exists a subset of Π that is equivalent to it and of size at most k. Since the problem can be expressed as a $\exists\forall$QBF, it is in Σ_2^p.

Hardness is proved via a quite complicated reduction from $\exists\forall$QBF. Let $\exists X \forall Y . \neg\Gamma$ be a formula, where $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$ is a set of clauses. This problem is Σ_2^p-hard, as it is the complement of the problem of deciding whether a $\forall\exists$QBF, in which the matrix is a CNF formula, is valid [15]. We build a set Π as the union of the following sets of clauses:

$$\begin{array}{rcl} \Pi_1 & = & \bigcup_{i=1,\ldots,n}^{j=1,\ldots,r} \{x_i^j, z_i^j\} \\ \Pi_2 & = & \bigcup_{i=1,\ldots,n} \{x_i^1 \wedge \cdots \wedge x_i^r \rightarrow x_i, z_i^1 \wedge \ldots \wedge z_i^r \rightarrow z_i\} \\ \Pi_3 & = & \bigcup_{i=1,\ldots,n} \{x_i \rightarrow w_i, z_i \rightarrow w_i\} \\ \Pi_4 & = & \bigcup_{j=1,\ldots,m} \{w_1 \wedge \cdots \wedge w_n \rightarrow \gamma_j^N\} \\ \Pi_5 & = & \{v_1, \ldots, v_t, \neg v_1 \vee \cdots \vee \neg v_t\} \end{array}$$

Here, γ_j^N is obtained from γ_j by replacing every positive occurrence of x_i with $\neg z_i$. The values of the constant k, r, and t are chosen as follows: if n is the number of variables and m the number of clauses of Γ, we set $r = m+1$, $k = (r+2) \cdot n + m$, and $t = k + 1$.

We prove that $\exists X \forall Y . \neg\Gamma$ is valid if and only if $\Pi = \Pi_1 \cup \Pi_2 \cup \Pi_3 \cup \Pi_4 \cup \Pi_5$ has an equivalent subset of size at most k.

The set Π is unsatisfiable, and this is an easy consequence of the fact that Π_5 is itself unsatisfiable. Therefore, we are looking for a subset of Π of size (at most) k that is unsatisfiable. Note that, removing even a single clause from Π_5, it becomes satisfiable. Since it does not share any variable with the other subsets, it follows that no proper subset of Π_5 can contribute to the generation of unsatisfiability. Since $t > k$, we are sure that an unsatisfiable set of clauses of size bounded by k either does not contain clauses from Π_5, or they can be removed from it while maintaining inconsistency. In short: while looking for an unsatisfiable subset of Π, clauses of Π_5 can be disregarded (these clauses are used to guarantee that Π is unsatisfiable).

We have therefore proved that Π has an I.E.S. of size bounded by k if and only if $\Pi_1 \cup \Pi_2 \cup \Pi_3 \cup \Pi_4$ has an inconsistent subset of size bounded by k. Let us therefore consider $\Pi' \subseteq \Pi_1 \cup \Pi_2 \cup \Pi_3$ and $\Pi'' \subseteq \Pi_4$, and see what happens when $\Pi' \cup \Pi''$ is an unsatisfiable set of at most k clauses.

First, neither Π' nor Π'' is unsatisfiable alone, as both $\Pi_1 \cup \Pi_2 \cup \Pi_3$ and Π_4 are satisfiable (the first is satisfied by the model that evaluates to true all variables, the second by the model that evaluates to false all variables.) Second, if Π' does not imply all w_i's, then $\Pi' \cup \Pi''$ is satisfiable. There exists exactly two minimal subsets of $\Pi_1 \cup \Pi_2 \cup \Pi_3$ that imply w_i:

$$\begin{array}{rcl} \Sigma_i & = & \bigcup_{j=1,\ldots,r} \{x_i^j\} \cup \{x_1^1 \wedge \cdots \wedge x_i^r \rightarrow x_i, x_i \rightarrow w_i\} \\ \Sigma'_i & = & \bigcup_{j=1,\ldots,r} \{z_i^j\} \cup \{z_1^1 \wedge \cdots \wedge z_i^r \rightarrow z_i, z_i \rightarrow w_i\} \end{array}$$

These two sets have the same size. The number k has been chosen so that $k = n \cdot (r+2) + m = n \cdot |\Sigma_i| + m$. Since all w_i's have to be implied, $\Sigma_i \subset \Pi'$ or $\Sigma'_i \subset \Pi'$ for each i. Since $m < |\Sigma_i|$, we have that $k < n \cdot (|\Sigma_i| + 1)$, that is, Π' cannot contain more than n sets Σ_i or Σ'_i. More precisely, $r + 1 = m + 2$ other clauses are necessary to imply another x_i or z_i, which are shared with Π''. Therefore, Π' must contain exactly one group among Σ_i and Σ'_i for any i, which amounts to $n \cdot (r+2)$ clauses. The remaining m clauses can be taken from Π''. Since Π_4 has size m, we can simply take $\Pi'' = \Pi_4$.

We have proved that Π' implies either x_i or z_i, for any i, but not both. Candidate unsatisfiable subset are therefore in correspondence with truth assignments on the variables x_i. Moreover, all variables w_i are true, which makes Π_4 equivalent to $\bigcup_{j=1,\ldots,m} \{\gamma_j^N\}$. If Π' contains x_i, then $\neg x_i$ can be removed from any clause γ_j^N containing it, while $\neg z_i$ remains. The opposite happens if z_i is in Π'.

Either way, if a variable of $\{x_i, z_i\}$ is in Π', the other one is not mentioned in Π', so we can assign it to false in order to satisfy as many clauses as possible (we are trying to prove unsatisfiability, so we have to test the most unfavorable possibility). What remains of Π_4 is the set Γ in which all variables x_i has been removed, by assigning them either to true (if $\Sigma_i \subset \Pi'$) or to false (if $\Sigma'_i \subset \Pi'$). Therefore, the choice of including Σ_i or Σ'_i makes Π_4 equivalent to Γ after setting x_i to some truth value. Therefore, Π has an unsatisfiable subset of size k if and only if $\exists X \forall Y . \neg\Gamma$ is true.

Note that the choice of an unsatisfiable set Π is not necessary. Indeed, by adding a new variable u to all clauses, Π and all its subsets are made satisfiable. Since Π is now equivalent to u, one of its subsets can be equivalent to it only if, assigning false to u, leads to unsatisfiability, which has been proved to be equivalent to the QBF problem. □

We now turn the problem of deciding whether a clause is in an I.E.S. A clauses that is in an I.E.S. can be considered "weakly irredundant", in the sense that there is a way of turning the set into an irredundant one that still contains the clause. A stronger version of this concept is that of clauses that are contained in all I.E.S.'s. Quite surprisingly, the complexity of the latter problem is lower than that of the former.

Theorem 4 *Given a clause γ and a set of clauses Π, deciding whether γ belongs to all I.E.S.'s of Π is* NP*-complete.*

Proof. At a first glance, this problem looks like the typical problem in Π_2^p: it amounts of checking whether γ belongs to all subsets of Π that are equivalent to it. Expressing it as a QBF, it would be something like: for all subsets $\Pi' \subseteq \Pi$, $\gamma \in \Pi'$ or Π' is not equivalent to Π. The latter can be expressed as: there is a model of Π' that is not a model of Π.

Nevertheless, the problem can be expressed in a simpler way: γ is in all I.E.S.'s of Π if and only if $\Pi \backslash \{\gamma\} \not\models \gamma$. This is a simple inference test, and is therefore in NP. We prove that this is also the same as the problem we are studying.

If $\Pi \backslash \{\gamma\} \not\models \gamma$, then γ belongs to all I.E.S.'s: this is an easy consequence of the fact that no subset of $\Pi \backslash \{\gamma\}$ can imply γ. What remains to prove is that $\Pi \backslash \{\gamma\} \models \gamma$ implies that there is an I.E.S. of Π that does not contain γ. We can build this I.E.S. as follows: we start from $\Pi \backslash \{\gamma\}$ and iteratively remove clauses that can be derived from it, until we obtain a set from which no clause can be removed. This is clearly an I.E.S., and it does not contain γ.

Having proved the problem to be in NP, what remains to be proved is its hardness. We can use the reduction of Theorem 1: the set Π is irredundant if and only if the original set of clauses Γ is satisfiable. Moreover, if Γ is unsatisfiable, Π has the single I.E.S. $\Pi \backslash \{\neg c_1 \vee \cdots \vee \neg c_m \vee \neg a\}$. Therefore, the clause $\neg c_1 \vee \cdots \vee \neg c_m \vee \neg a$ is in all I.E.S.'s of Π if and only if Γ is satisfiable. □

While deciding whether a clause is in all I.E.S.'s is in NP, the similar problem of deciding whether a clause is in *at least one* I.E.S. is complete for the class Σ_2^p, and is therefore harder. This result is somehow surprising, as these two problems have very similar definitions, and checking the existence of an I.E.S. containing a clause may look even simpler than checking all of them.

Theorem 5 *Deciding whether a clause γ is in at least one I.E.S. of a set of clauses Π is Σ_2^p-complete.*

Proof. Membership is trivial: the problem can be expressed as the existence of a set $\Pi' \subseteq \Pi$ containing γ that is equivalent to Π.

Hardness is proved by reduction from $\exists\forall$QBF. We assume that the matrix of the QBF formula is the negation of a CNF: this problem is Σ_2^p-hard, as it is the complement of deciding whether a $\forall\exists$QBF formula, in which the matrix is in CNF, is valid [15]. We prove that $\exists X \forall Y . \neg\Gamma$ is valid (where $\Gamma = \{\gamma_1, \ldots, \gamma_m\}$) if and only if w is in at least one I.E.S. of the following set Π:

$$\Pi = \bigcup_{i=1,\ldots,n} \{x_i, \neg x_i\} \cup \{w\} \cup \bigcup_{i=1,\ldots,m} \{w \rightarrow \gamma_i\}$$

This set is clearly unsatisfiable. Its I.E.S.'s are its unsatisfiable minimal subsets. Let us now show how a subset Π' of this kind is composed. If both x_i and $\neg x_i$ are in Π', they are enough to generate contradiction, so no other clause can be in Π', otherwise the other clauses would be redundant. We have therefore found a first group of minimal unsatisfiable subsets of Π: those composed exactly by a pair $\{x_i, \neg x_i\}$.

Let us now try to build an unsatisfiable $\Pi' \subseteq \Pi$ that contains w. Besides w, such set Π' can include $\bigcup_{i=1,\ldots,m} \{w \rightarrow \gamma_i\}$, as well as a literal between x_i and $\neg x_i$ for any i (but not both, otherwise the other clauses would be redundant). It is now evident that such set can be unsatisfiable only if, for the given choice of the x_i's, the set Γ is unsatisfiable. Thus, there exists an unsatisfiable subset of Π containing w if and only if Γ is unsatisfiable. What remains to prove is that any I.E.S. obtained by removing redundant clauses from Π' contains w. This is an easy consequence of the fact that $\Pi' \backslash \{w\}$ is always satisfiable. □

Any set of clauses has at least one I.E.S. Checking the existence of an I.E.S. is thus trivial. On the other hand, a set may have more than one I.E.S. Deciding whether this holds for a specific set of clauses is important, as it shows that there is a choice among the possible minimal representations of the same piece of information. For example, a trivial algorithm for producing an I.E.S. is that of iteratively removing the first clause that is implied by the other ones. This algorithm clearly outputs an I.E.S. However, other ones may exist, and be better either because are shorter (have less clauses), or because their structure make them more effective to use (for example, they are Horn or in a similar special form that makes reasoning with them easier.) The problem of uniqueness is $\Delta_2^p[\log n]$ complete.

Theorem 6 *Deciding whether a set of clauses Π has a single I.E.S. is $\Delta_2^p[\log n]$ complete.*

Proof. Expressing this problem as a QBF, it looks like a problem in Σ_2^p. However, we can show that a logarithmic number of satisfiability checks suffice to solve the problem. This is proved by showing that a set Π has a unique I.E.S. if and only if $\Pi_R \models \Pi$, where:

$$\Pi_R = \{\gamma \in \Pi \mid \Pi \backslash \{\gamma\} \not\models \gamma\}$$

Let us first assume that $\Pi_R \models \Pi$, and prove that Π_R is the unique I.E.S. of Π. Since no clause of Π_R is implied by the rest of Π, it is not implied by the rest of Π_R either, which proves that Π_R is a I.E.S. We only have to prove that Π does not have any other I.E.S. Assume, by contradiction, that $\Pi' \neq \Pi_R$ is a I.E.S.: if $\Pi_R \subset \Pi'$, then Π' is not irredundant; otherwise, there exists $\gamma \in \Pi_R \backslash \Pi'$. This condition can be decomposed into $\gamma \in \Pi_R$ and $\gamma \notin \Pi'$. The first formula implies $\Pi \backslash \{\gamma\} \not\models \gamma$ by construction of Π_R. The second formula, together with $\Pi' \models \Pi$, implies that $\Pi' \backslash \{\gamma\} \models \gamma$. This is a contradiction, as $\Pi' \subseteq \Pi$.

The problem can therefore be solved by determining Π_R and then checking whether $\Pi_R \models \Pi$. This can be done with a polynomial number of parallel calls to an oracle in NP, followed by a single other call. By a well-known result by Gottlob [7], the problem is in $\Delta_2^p[\log n]$.

We prove that the problem of uniqueness is $\Delta_2^p[\log n]$-hard by reduction from the problem of odd satisfiability: given a sequence of sets of clauses $(\Pi^1, \ldots, \Pi^r)$, each built on its own alphabet, such that the unsatisfiability of Π^j implies that of Π^{j+1}, decide whether the first Π^k that is unsatisfiable is of odd index, that is, k is odd.

Let Π^j be a set of clauses, and let $\Pi_C^j = \{c_i^j \rightarrow \gamma_i^j \mid \gamma_i^j \in \Pi^j\}$ and $\gamma_g^j = \neg c_1^j \vee \cdots \vee \neg c_m^j$. As proved in Theorem 1, Π_C^j implies the clause γ_g^j if and only if Π^j is unsatisfiable.

Let j be an odd index between 1 and r. Define:

$$\begin{aligned} \Pi_D^j &= \Pi_C^j \cup \{a^j \vee c^j \vee \gamma_g^j, b^j \vee c^j \vee \gamma_g^j\} \\ \Pi_D^{j+1} &= \Pi_C^{j+1} \cup \{c_i^{j+1} \vee a^j \vee \neg b^j \vee \gamma_g^j \mid \gamma_i^{j+1} \in \Pi^{j+1}\} \cup \\ & \quad \{c_i^{j+1} \vee \neg a^j \vee b^j \vee \gamma_g^j \mid \gamma_i^{j+1} \in \Pi^{j+1}\} \end{aligned}$$

We give a sketch of the proof that the first unsatisfiable set of the sequence is of odd index if and only if $\Pi = \Pi_D^1 \cup \cdots \cup \Pi_D^r$ has multiple I.E.S.'s.

Variables are only shared between Π_D^j and Π_D^{j+1}, and only if j is odd. We therefore only have to check whether $\Pi_D^j \cup \Pi_D^{j+1}$ has a unique I.E.S., where j is odd.

Intuitively, if Π_j is unsatisfiable, then γ_g^j is implied, thus making all clauses but those in $\Pi_D^j \cup \Pi_D^{j+1}$ redundant, which leads to a single I.E.S. If both sets are satisfiable, all clauses are irredundant (this is the longest part of the proof.) Finally, if Π_D^j is satisfiable while Π_D^{j+1} is not, then at least a variable c_i^{j+1} is implied, thus making $(a^j \equiv b^j) \vee \gamma_g^j$ implied as well, and the two last clauses of Π_D^j are made equivalent; therefore, one of them can be removed, but not the other one.

The complete proof is omitted due to the lack of space. $\square$

5 CONCLUSIONS

We have presented a computational analysis of some problems related to redundancy and redundancy elimination. Namely, checking whether a set of clauses is irredundant has been proved to be coNP-complete. We have then defined an I.E.S. as an irredundant equivalent subset of a given set, and studied some problems related to I.E.S.'s: checking, size, uniqueness, and membership of clauses to some or all I.E.S.'s. All problems have been given an exact characterization within the polynomial hierarchy, that is, we have found classes these problems are complete for.

There are still open problems, however, and are about the problem of minimality of formulae in general. Indeed, irredundancy is only one way of defining minimal representation of a formula, but other ones exist. In the Horn case, several different definitions of minimality have been used, both by Meier [11] and by Ausiello et al. [1], including irredundancy and number of occurrences of literals. In the general (non-Horn) case, only the number of occurrences of literals (and, in this paper, irredundancy) have been considered. An open problem is whether the other notions of minimality used in the Horn case make sense in the general case as well.

A further open question is whether the results changes when alternative forms of equivalence, like model equivalence or query equivalence [4], are considered. Both these forms of equivalence are based on adding new variables to form a circuit or a formula which is equivalent to the original one only on the original variables. Since both forms of equivalence can be used for reducing the size of a propositional formula, it makes sense to investigate whether a formula is (query or model) equivalent to another one of smaller size.

6 ACKNOWLEDGMENTS

Many thanks to Marco Schaerf for his comments on a previous version of this paper, and to the anonymous reviewers for their very detailed comments.

REFERENCES

[1] G. Ausiello, A. D'Atri, and D. Saccà, 'Minimal representation of directed hypergraphs', *SIAM J. on Computing*, **15**(2), 418–431, (1986).

[2] A. Blass and Y. Gurevich, 'On the unique satisfiability problem', *Information and Control*, **55**(1–3), 80–88, (1982).

[3] M. Cadoli, F. M. Donini, P. Liberatore, and M. Schaerf, 'The size of a revised knowledge base', *Artificial Intelligence*, **115**(1), 25–64, (1999).

[4] M. Cadoli, F. M. Donini, P. Liberatore, and M. Schaerf, 'Space efficiency of propositional knowledge representation formalisms', *Journal of Artificial Intelligence Research*, **13**, 1–31, (2000).

[5] A. Ginsberg, 'Knowledge base reduction: A new approach to checking knowledge bases for inconsistency & redundancy', in *Proc. of AAAI'88*, pp. 585–589, (1988).

[6] M. L. Ginsberg, 'Conterfactuals', *Artificial Intelligence*, **30**, 35–79, (1986).

[7] G. Gottlob, 'NP trees and Carnap's modal logic', *J. of the ACM*, **42**(2), 421–457, (1995).

[8] G. Gottlob and C. G. Fermüller, 'Removing redundancy from a clause', *Artificial Intelligence*, **61**, 263–289, (1993).

[9] P. Hammer and A. Kogan, 'Optimal compression of propositional Horn knowledge bases: Complexity and approximation', *Artificial Intelligence*, **64**(1), 131–145, (1993).

[10] E. Hemaspaandra and G. Wechsung, 'The minimization problem for Boolean formulas', in *Proc. of FOCS'97*, pp. 575–584, (1997).

[11] D. Maier, 'Minimum covers in relational database model', *J. of the ACM*, **27**(4), 664–674, (1980).

[12] A. Meyer and L. Stockmeyer, 'The equivalence problem for regular expressions with squaring requires exponential space', in *Proc. of FOCS'72*, pp. 125–129, (1972).

[13] B. Nebel, 'Belief revision and default reasoning: Syntax-based approaches', in *Proc. of KR'91*, pp. 417–428, (1991).

[14] J. Schmolze and W. Snyder, 'Detecting redundant production rules', in *Proc. of AAAI'97*, pp. 417–423, (1997).

[15] L. J. Stockmeyer and A. R. Meyer, 'Word problems requiring exponential time', in *Proc. of STOC'73*, pp. 1–9, (1973).

[16] C. Umans, 'The minimum equivalent DNF problem and shortest implicants', in *Proc. of FOCS'98*, pp. 556–563, (1998).