ELSEVIER
Theoretical Computer Science 182 (1997) 183–202
Theoretical Computer Science

On compact representations of propositional circumscription[1]

Marco Cadoli[a], Francesco M. Donini[a], Marco Schaerf[a,*], Riccardo Silvestri[b]

[a] *Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Salaria 113, I-00198, Roma, Italy*

[b] *Dipartimento di Scienze dell'Informazione, Università di Roma "La Sapienza", Via Salaria 113, I-00198, Roma, Italy*

Received July 1995; revised May 1996
Communicated by G. Ausiello

Abstract

Circumscription is a popular common-sense reasoning technique, used in the fields of Artificial Intelligence, Databases and Logic Programming. In this paper we investigate the size of representations (formulae, data structures) equivalent to the circumscription of a propositional formula T, taking into account three different definitions of equivalence. We find necessary and sufficient conditions for the existence of polynomial-size representations (formulae, data structures) equivalent to the circumscription of T in the three cases. All such conditions imply the collapse of the polynomial hierarchy. In particular, we prove that – unless the polynomial hierarchy collapses at the second level – the size of the shortest propositional formula T' logically equivalent to the circumscription of T grows faster than any polynomial as the size of T increases. The significance of this result in the related field of closed-world reasoning is then analyzed.

1. Introduction

Reasoning with selected (or intended) models of a logical formula is a common reasoning technique used in databases, logic programming, knowledge representation and artificial intelligence (AI). One of the most popular criteria for selecting intended models is *minimality* w.r.t. the set of true atoms. The idea behind minimality is to assume that a fact is false whenever possible. Such a criterion allows one to represent only true statements of a theory, saving the explicit representation of all false ones. For propositional theories, the explicit (and finite) representation is always possible; but how large is its size, compared with the size of the implicit representation?

* Corresponding author. e-mail: schaerf@dis.uniroma1.it.

[1] Extended and revised version of [4].

PII S0304-3975(96)00182-X

In this paper we address the following problem:

> Is it the case that for each propositional formula T, there is a "compact" representation of the minimal models of T? By compact we mean polynomially-sized w.r.t. the size of T, for some fixed polynomial.

We consider three different formal notions of "representation of the minimal models", and give a negative answer to this problem for all of them (provided the polynomial hierarchy does not collapse).

1.1. Motivation

A well-established formalization of minimality is *circumscription*, which has been introduced in the AI literature [19, 20] for capturing some important aspects of commonsense reasoning, and was shown to be strictly related to closed-world reasoning in databases. From a formal point of view, circumscription is a fragment of second-order logic, as circumscription of a first-order formula yields a second-order universal formula. The propositional version has also been defined: Circumscription of a propositional formula yields a universally quantified boolean formula.

Several studies about computational properties of circumscription appeared in the literature. Several aspects, such as time complexity of inference, model checking and model finding have been studied. Noticeably, those studies proved that reasoning with circumscriptive formulae is harder than reasoning with formulae of classical logic. As an example, inference in propositional circumscription is Π_2^p-complete [7], while the same problem is coNP-complete in classical propositional logic.

Another interesting computational aspect that has been addressed is *collapsibility*. The question can be stated as follows: Given a first-order formula T, is its circumscription (denoted by $CIRC(T)$) – which is a second-order formula – equivalent to some finite first-order formula? The answer in general is no [14, 15], but there are syntactically restricted classes of formulae in which this is true [14, 17, 26].

In principle, two distinct notions of equivalence can be analyzed: *logical* equivalence, i.e., the two formulae have exactly the same models, and *query* equivalence, i.e., the two formulae have exactly the same theorems. Clearly, logical equivalence implies query equivalence, but the converse does not necessarily hold.

As for the propositional case, collapsibility to a logically equivalent formula is not a problem at all: Given a propositional formula T we can easily write a propositional formula T' that is equivalent to $CIRC(T)$. A trivial way to do that is to make a disjunction of all the minimal models of T, as they are exactly the models of $CIRC(T)$. It is easy to see that such a process may generate an exponential-size representation of $CIRC(T)$, as T can have exponentially many minimal models. A smarter method would be to compute the *extended generalized closed world assumption* $EGCWA(T)$ of T, which is equivalent to $CIRC(T)$ [10, 32]. Syntactically, $EGCWA(T)$ is T plus a set of clauses, which constrain the models exactly to the minimal ones. Nevertheless the size of $EGCWA(T)$ may be exponential, as discussed in Section 4.

In this paper we prove that, unless the polynomial hierarchy collapses at a sufficiently low level, as the size of T increases, the size of the explicit representation of $CIRC(T)$ grows faster than any polynomial. This result has several consequences. Suppose you have a knowledge base T and you want to pose it several different queries under circumscription. An (apparently) reasonable approach is to rewrite (off-line) the knowledge base into a propositional one T', equivalent to $CIRC(T)$, and then query (on line) T'. This approach seems to move the complexity from on-line to off-line. Our result shows that, in general, this approach is not feasible and it does not make on-line reasoning any quicker, due to the super-polynomial increase in the size of the knowledge base.

While this is a negative result on circumscription, it has a positive side. In fact, our result also implies that circumscription is able to represent information in a very compact fashion: Imagine you have a certain amount of propositional knowledge to be represented; you may go for classical semantics or for circumscriptive semantics. Let A and B be the formulae you obtain, respectively (obviously $A \equiv CIRC(B)$ must hold). There are cases where the size of A is significantly bigger than the size of B.

1.2. Related work

The problem of collapsibility of the circumscription of first-order formulae has received considerable attention in the literature [6, 8, 17, 26]. The issue of the size of the resulting formulae is addressed in [14], where it is noted that computing the first-order sentence equivalent to the circumscription of a first-order existential formula T is possible, but its size is exponential w.r.t. T. The question whether this is inherent to existential first-order formulae is left as an open problem.

A pragmatic approach to the problem is taken in [23], where an algorithm for computing all minimal models of a deductive database is proposed. The underlying idea of the method is to store the set of models, once they are computed. The algorithm for computing the minimal models is based on a translation of the database into an integer programming problem. Our results suggest that, even for such a sophisticated technique, there must be cases in which the space needed to store the models is super-polynomial.

Related work appears also in AI: A popular idea in this field is that of preprocessing a logical formula T to obtain a data structure in which fast algorithms for answering $T \models Q$, Q being another propositional formula, can be used. In general, one wants a *vivid* form of knowledge where reasoning is polynomially tractable [16]. An example of this kind is reported in [22]. In the paper the authors analyze the possibility of speeding up query answering in propositional logic (i.e., checking whether $T \models Q$ holds) through a previous off-line transformation of the theory T. Abstractly, they want to transform a coNP-complete problem into a polynomial one (obviously not in polynomial time).

In the same spirit, our problem can be seen as an attempt at transforming a reasoning problem into a simpler one via off-line reasoning. In fact, if we are able to construct a polynomial-size T' equivalent to $CIRC(T)$ then inference under circumscription (which is Π_2^p-complete [7]) is transformed into inference in propositional logic (which is coNP-complete). Similarly, model checking in circumscription (i.e., given a propositional theory T and an interpretation M decide whether $M \models CIRC(T)$, a coNP-complete problem [1]), is transformed into model checking in propositional logic (which is solvable in polynomial time). Therefore it is unlikely that the transformation from T to T' can be accomplished in polynomial time. In fact in this paper we do not impose any restriction on the time needed for the construction of T', which could even be a non-recursive process.

This technique has also been used in related fields. As an example, in [2] the issue of the size of a formula which is the result of a revision of a propositional knowledge base is analyzed. In particular, such a size can be proven to be either polynomial or super-polynomial, depending on factors such as the semantics adopted for belief revision, or the notion of equivalence which is taken into account. In [3, 5], the issue of "compiling" polynomially intractable non-monotonic inference problems into polynomial-time solvable problems is addressed.

The idea of "compiling" a propositional formula into another formula (called *Horn least upper bound*) which is not a faithful representation of the original one is proposed in [13, 28]. By using non-uniform complexity classes, the authors are able to exhibit the proof that the Horn least upper bound may have super-polynomial size w.r.t. the original formula.

1.3. Main results

We focus on the size of representations of the circumscription $CIRC(T)$ of a propositional formula T. Two distinct notions of representation are considered: *model-based* (i.e., preserving the set of models), and *query-based* (i.e., preserving the set of theorems). As far as the former is concerned, it is useful to distinguish between *propositional formulae* having the same set of models, and generic *data structures* (e.g., boolean circuits) that allow to do model checking. As for the latter, such a distinction is not necessary, as we will see in Section 3.3.

Three notions of equivalence, which are now informally described, are therefore considered.

Logical equivalence: Propositional formula T' is *logically equivalent* to $CIRC(T)$ iff they have exactly the same models, i.e., for each truth assignment M, $M \models CIRC(T)$ iff $M \models T'$;

Model equivalence: Data structure D is *model equivalent* to $CIRC(T)$ iff there exists a polynomial-time algorithm ASK such that it is possible to do model checking for $CIRC(T)$ by using D, i.e., for each truth assignment M, $M \models CIRC(T)$ iff $ASK(D, M)$ returns "yes";

Query equivalence: Propositional formula T' is *query equivalent* to $CIRC(T)$ iff they have exactly the same theorems on the common language, i.e., for each formula Q in which only symbols of T occur, $\{Q \mid CIRC(T) \models Q\} = \{Q \mid T' \models Q\}$.
The three notions are partially ordered with respect to their strength: A formula T' satisfying logical equivalence is also a data structure satisfying model equivalence

and a formula satisfying query equivalence. Analogously, a data structure D satisfying model equivalence satisfies query equivalence as well. The other direction does not necessarily hold. Intuitively, model equivalence allows to do model checking by using circuits, while query equivalence gives the possibility of introducing new propositional atoms.

Logical equivalence is the strongest notion, but the other ones might have a practical interest in fields such as automated theorem proving or deductive databases.

We find necessary and sufficient conditions for the existence of polynomial-size representations equivalent to $CIRC(T)$ in the three different cases. All such conditions imply the collapse of the polynomial hierarchy. In particular, we prove that – unless the polynomial hierarchy collapses at the second level – the size of the shortest propositional formula T' logically equivalent to $CIRC(T)$ grows faster than any polynomial as the size of T increases. The major tools we use are

1. the notion of *non-uniform* computation [11, 12];
2. results proving that – in propositional circumscription – logical entailment is Π_2^p-complete [7] and model checking is coNP-complete [1];
3. results relating inclusion of uniform complexity classes into non-uniform complexity classes to the collapse of the polynomial hierarchy [12, 31].

1.4. Structure of the paper

The structure of the paper is the following: In the next section we recall some definitions about propositional circumscription and non-uniform computation; then, in Section 3 we prove our main results. In Section 4 we discuss these results, and analyze their significance in the related field of closed-world reasoning. In the last section we draw some conclusions and address open problems.

2. Preliminaries

The *alphabet* of a propositional formula is the set of all propositional atoms occurring in it. An *interpretation* of a formula is a truth assignment to the atoms of its alphabet. A *model* M of a formula T is an interpretation that satisfies T (written $M \models T$). Interpretations and models of propositional formulae will be denoted as sets of atoms (those which are mapped into 1). Given a propositional formula T, we denote with $\mathcal{M}(T)$ the set of its models. Following Lifschitz [17], we define:

Definition 1. Let $M \in \mathcal{M}(T)$. M is called a minimal model of T if there is no model N of T such that $N \subset M$ (i.e., $N \not\equiv M$ and $N \subseteq M$.)

Definition 2. Let T be a propositional formula and $X = \{x_1, \dots, x_n\}$ its alphabet. The circumscription $CIRC(T)$ is the following quantified boolean formula

$$T \wedge (\forall Y . T[Y] \rightarrow \neg (Y < X)), \tag{1}$$

where $Y=\{y_1,\dots,y_n\}$ is an ordered set of atoms disjoint from X, $T[Y]$ is T with all the occurrences of atoms of X substituted by the corresponding ones in Y. The meaning of $Y < X$ is defined in terms of the relation $\leqslant$. In particular, $Y < X$ is $(Y \leqslant X) \wedge \neg(X \leqslant Y)$, and $Y \leqslant X$ stands for the conjunction of the formulae

$$y_i \to x_i \quad (1 \leqslant i \leqslant n).$$

Proposition 1 (Lifschitz [17, Proposition 1]). *A model M of T is minimal iff it is a model of $CIRC(T)$, i.e., iff $M \models CIRC(T)$.*

More sophisticated definitions of minimal models and circumscription have been defined (e.g., not all atoms are minimized [10, Definition 3.1]). As such definitions are extensions of basic circumscription, the results we present in this paper hold for them.

Throughout this paper, the symbol $|x|$ denotes the size of x and also the cardinality of x, when x is a set. Furthermore, $\langle\cdot,\cdot\rangle$ denotes a pairing function over binary finite strings with the standard nice computability and invertibility properties, and such that $\forall x, y \,.\, |\langle x, y\rangle| = 2|x| + 2|y|$.

As already pointed out, our proof uses the notion of non-uniform computations. We now briefly recall the definitions needed in the sequel (cf. [12]).

Definition 3. An *advice function* is a function $A:\mathbb{N} \to \{0,1\}^*$, where $\mathbb{N}$ is the set of non-negative integers. The advice function A is *polynomial* if $|A(n)| \leqslant p(n)$ for some polynomial p and all non-negative integers n.

Definition 4. Let $\mathscr{C}$ be any class of languages. $\mathscr{C}$/poly is the class of all languages of the form $\{x \mid \langle A(|x|), x\rangle \in L\}$ where $L \in \mathscr{C}$ and A is a polynomial advice function.

Any class $\mathscr{C}$/poly is also known as non-uniform $\mathscr{C}$. Non-uniformity is due to the presence of the advice. Notice that the advice depends only on the size of the input, not on the input itself. Throughout the paper, we will be interested in several non-uniform classes. More precisely, we use the classes P/poly, NC^1/poly, NP/poly and coNP/poly. Following Johnson [11], the class NC^1 is defined as

Definition 5. The class NC^1 consists of all languages recognizable by log-space uniform families of Boolean circuits having polynomial size and depth $O(\log n)$.

It is important to point out that both P/poly and NC^1/poly are closed under complementation, while NP/poly and coNP/poly are classes of complementary languages.

Relations between non-uniform and uniform complexity classes, were studied in the literature by several researchers (e.g. [12, 31]). The relevant results, for our work, can be summarized as follows:

NP $\subseteq$ NC^1/poly	$\Rightarrow$	Σ_2^p = PH	[12]
NP $\subseteq$ P/poly	$\Rightarrow$	Σ_2^p = PH	[12]
NP $\subseteq$ coNP/poly	$\Rightarrow$	Σ_3^p = PH	[31]

That is, if NP is included in any of the three classes NC^1/poly, P/poly or coNP/poly then the polynomial hierarchy collapses at a low level (either the second or the third one).

Clearly, NP $\subseteq$ NC^1/poly $\Rightarrow$ NP $\subseteq$ P/poly $\Rightarrow$ NP $\subseteq$ coNP/poly, but the inverse implications are not known to hold. In particular, condition "NP $\subseteq$ NC^1/poly" seems to be much stronger than "NP $\subseteq$ P/poly" in that NC^1 is a class that is believed much smaller than P; indeed NC^1 $\subseteq$ LOGSPACE. Also, while it is not currently known whether P is a proper subclass of PSPACE, this is known to be true for NC^1.

3. Compact representations

In this section we address the three forms of equivalence previously defined, each one in a separate subsection. We prove that the existence of an equivalent compact representation of the circumscription of a propositional formula corresponds to the inclusion of NP in a non-uniform class, different for any notion of equivalence considered.

Many of the proofs share a common property, which we state and prove as a general lemma below. Intuitively, the lemma tells us that there is a class of propositional formulae $T_{n,m}$, such that the validity of any quantified boolean formula F with n universally quantified variables followed by m existentially quantified variables is equivalent to entailment under circumscription between $T_{n,m}$ and a particular query Q_F. The formal statements follow.

Given two sets of propositional atoms $X = \{x_1,\dots,x_n\}$, $Y = \{y_1,\dots,y_m\}$, and a 3CNF formula E containing literals on the alphabet $X \cup Y$, we call a $\forall\exists$-QBF the following quantified boolean formula F:

$$F = \forall x_1,\dots,x_n \;\exists y_1,\dots,y_m \,.\, E \qquad (2)$$

We call E the matrix of F.

Lemma 2. *Let X and Y be two alphabets of n and m atoms, respectively, Z an alphabet one-to-one with X (i.e., Z has n atoms), and W a fourth alphabet with $16(n+m)^3+1$ atoms. There exists a 5CNF formula $T_{n,m}$ over $X \cup Y \cup W \cup Z$ (depending only on n and m, of polynomial size w.r.t. $n+m$) such that given any $\forall\exists$-QBF F of the form* (2), *there exists a clause Q_F, containing all atoms in $Y \cup W$, such that F is valid iff $CIRC(T_{n,m}) \models Q_F$.*

Proof. The proof is inspired by a reduction given in [7], showing that inference in a 3CNF-theory under circumscription is Π_2^p-hard. The key difference is that we now need to encode *every* possible $\forall\exists$-QBF of the form (2) in our theory $T_{n,m}$.

Let C be a set of new atoms, one for each three-literals clause over $X \cup Y$, i.e., $C = \{c_i \mid \gamma_i$ is a three-literals clause of $X \cup Y\}$. Moreover, let D be a set of new atoms in one-to-one correspondence with atoms of C, and let Z be as above. Finally, let W be the set of atoms $C \cup D \cup \{u\}$, where u is a distinguished atom. Notice that $|W|$ is $2(2(n+m))^3 + 1 = 16(n+m)^3 + 1$.

We want to impose non-equivalence between atoms in X and their corresponding atoms in Z, and the same for C and D. We call $\Sigma_{n,m}$ the 2CNF formula made up of the following clauses:

1. for each atom x_i of X, there are two clauses $x_i \vee z_i$, and $\neg x_i \vee \neg z_i$ in $\Sigma_{n,m}$ ($2n$ clauses),
2. for each atom c_i of C, there are two clauses $c_i \vee d_i$, and $\neg c_i \vee \neg d_i$ in $\Sigma_{n,m}$ ($2(2(n+m))^3$ clauses),

Now we want to encode every possible 3CNF formula over $X \cup Y$, using the atoms in C as "enabling gates". We call $\Gamma_{n,m}$ the 4CNF formula containing, for each three-literals clause γ_i over $X \cup Y$, a clause $\gamma_i \vee \neg c_i$ ($(2(n+m))^3$ clauses).

We define T (omitting subscripts n,m from now on, for readability) as:

$$\Sigma \wedge ((u \wedge y_1 \wedge \cdots \wedge y_n) \vee (\neg u \wedge \Gamma)) \tag{3}$$

Notice that the size of T is $O((n+m)^3)$, and T can be rewritten as an equivalent 5CNF formula. Moreover, T does not depend on a specific $\forall\exists$-QBF F, but only on X and Y.

Given the $\forall\exists$-QBF F with matrix E, we denote

$$C_E = \{c_i \in C \mid \gamma_i \text{ is a clause of } E\}$$

and similarly for D_E. Moreover, $\overline{C_E} = C - C_E$, and $\overline{D_E} = D - D_E$. Given $M \in \mathcal{M}(\Sigma)$, we denote $\mathcal{E}(M) = M \cup Y \cup \{u\}$.

We first note the following properties of T.

Lemma 3. *Let T be as in* (3). *Then*:

1. *T is satisfiable.*
2. *If $M \in \mathcal{M}(\Sigma)$ then $\mathcal{E}(M) \in \mathcal{M}(T)$.*
3. *Every model $M \in \mathcal{M}(T)$ defines a unique matrix $E = \{\gamma_i \mid c_i \in M\}$.*
4. *For every model $M \in \mathcal{M}(T)$ such that $M \cap C = C_E$, M satisfies Γ iff M satisfies E.*

Proof. 1. Let $M = D \cup X$. Obviously, $M \models \Sigma$ and $M \models \neg u \wedge \Gamma$. Therefore $M \models T$.

2. M already satisfies Σ; moreover, $Y \cup \{u\}$ satisfies the first disjunct in $(u \wedge y_1 \wedge \cdots \wedge y_n) \vee (\neg u \wedge \Gamma)$.

3. Obvious.

4. M satisfies Γ iff it satisfies all the clauses $\gamma_i \vee \neg c_i$ of Γ. Since $M \cap C = C_E$ and each clause of Γ contains a different $\neg c_i$, each clause is satisfied iff either it contains

a $\neg c_i$ such that $c_i \in \overline{C_E}$ or the remaining part of the clause (γ_i) is satisfied. But the conjunction of these remaining clauses is exactly E. □

We define the query Q_F as:

$$Q_F = \neg u \vee \neg y_1 \vee \cdots \vee \neg y_m \vee \left(\bigvee_{c_i \in C_E} \neg c_i \right) \vee \left(\bigvee_{c_i \in \overline{C_E}} c_i \right) \vee \left(\bigvee_{d_i \in D_E} d_i \right) \vee \left(\bigvee_{d_i \in \overline{D_E}} \neg d_i \right)$$

Observe that Q_F contains all atoms of $Y \cup W$. We prove the lemma by showing that F is valid iff $CIRC(T) \models Q_F$.

If part. Suppose that Q_F is true in all minimal models of T. Consider any model $M \in \mathscr{M}(\Sigma)$ such that $M \cap C = C_E$, that is, M contains exactly the atoms of C corresponding to clauses of E. Note that the model $\mathscr{E}(M) = M \cup Y \cup \{u\}$ cannot be a minimal model, because it does not satisfy Q_F. Hence for any such M there exists a different extension $\mathscr{E}_1(M)$, with $\mathscr{E}_1(M) \subset \mathscr{E}(M)$, that is a minimal model of T satisfying Q_F. Since $\mathscr{E}_1(M)$ satisfies Q_F, it must satisfy $\neg u$. As a consequence, $\mathscr{E}_1(M) \models \Gamma$ (cf. (3)). But any model satisfying Γ and whose intersection with C equals C_E satisfies also E (cf. Lemma 3, point 4). Hence for any $M \in \mathscr{M}(\Sigma)$ such that $M \cap C = C_E$, there is an extension $\mathscr{E}_1(M)$ satisfying E. Since each $M \in \mathscr{M}(\Sigma)$ contains a truth assignment to variables in X, it follows that for each assignment to variables in X there is an assignment to variables in Y (namely, $\mathscr{E}_1(M) \cap Y$) such that E is satisfied. Therefore, F is valid.

Only if part. Assume that there exists a minimal model M of T such that $M \models \neg Q_F$. Observe that $M \models u \wedge y_1 \wedge \cdots \wedge y_n$, $M \cap C = C_E$, $M \cap D = \overline{D_E}$. Let $M_\Sigma = M \cap (X \cup Z \cup C \cup D)$. Obviously, M is an extension of M_Σ. The minimality of M implies that $M \not\models \Gamma$, otherwise $M - \{u\}$ would be a model of T. Since $M \cap C = C_E$ and $M \not\models \Gamma$, by Lemma 3 (point 4) we also have that $M \not\models E$. Again from the minimality of M, it follows that no other extension M' of M_Σ satisfies T. As a consequence, there exists an assignment to the variables in X (i.e., $M_\Sigma \cap X$) for which there is no assignment to the variables in Y that makes E true. Therefore F is not valid. □

3.1. Model equivalence

In this section we prove that, unless the polynomial hierarchy collapses at the second level, there is no polynomial in $|T|$ bounding the size of the shortest data structure representing exactly the minimal models of T. We recall the notion of model equivalence, fixing it to polynomial-size data structures:

Let p be a fixed polynomial; for any propositional formula T we want to find a data structure D_T with the following characteristics:

1. $|D_T| < p(|T|)$;

2. there exists a relation $ASK(\cdot, \cdot)$, such that given any interpretation M of T, $ASK(D_T, M)$ is true iff $M \not\models CIRC(T)$ (i.e., ASK computes the complement of model checking);

3. deciding the relation $ASK(\cdot,\cdot)$ is a problem in P, where the inputs are its arguments.

Intuitively, this means that we are trying to "compile" $CIRC(T)$ in such a way that the NP-complete problem of deciding $M \not\models CIRC(T)$ becomes a problem in P. Note that a way of doing that would be to rewrite $CIRC(T)$ into an equivalent propositional formula T' of size bounded by $p(|T|)$, where ASK corresponds to the complement of classical model checking, i.e., $ASK(T',M) = true$ iff $M \not\models T'$ (which can be checked in time polynomial w.r.t. the size of M and T'). However, we are now looking not just for a formula, but for any data structure (i.e., any circuit).

We are able to show that it is very unlikely that such a polynomial p and data structure D_T may exist. As a consequence, T' does not exist either. In order to prove this, we resort to the notion of non-uniform computation. In what follows, a relation R such that deciding R is a problem in P will be called a P-*relation*.

Theorem 4. *Let p be any polynomial and let $ASK(\cdot,\cdot)$ be a* P*-relation. If for each CNF formula T there is a data structure D_T such that $|D_T| < p(|T|)$ and for any interpretation M, $M \not\models CIRC(T)$ if and only if $ASK(D_T,M)$ is true, then* NP $\subseteq$ P/*poly.*

Proof. Consider Lemma 2, with $X = \emptyset$ (i.e., $n = 0$, and $Z = \emptyset$, too). In this case the lemma says that there exists a 5CNF formula $T_{0,m}$ over $Y \cup W$ such that given any $\forall\exists$-QBF F using the atoms in Y for existentially quantified variables, there exists a clause Q_F, containing all atoms in $Y \cup W$, such that F is valid iff $CIRC(T_{0,m}) \models Q_F$. Observe that $CIRC(T_{0,m}) \not\models Q_F$ iff there exists a minimal model M_F satisfying $\neg Q_F$. Since Q_F contains all atoms in $T_{0,m}$, $\neg Q_F$ uniquely identifies the model M_F of $CIRC(T_{0,m})$ falsifying Q_F. Hence, F is valid iff $M_F \not\models CIRC(T_{0,m})$.

Let us assume that there exists a polynomial p with the properties claimed in the statement of the theorem. Then, for each $T_{0,m}$ there exists a data structure $D_{T_{0,m}}$, with $|D_{T_{0,m}}| < p(|T_{0,m}|)$, and a P-relation $ASK(\cdot,\cdot)$ such that given any interpretation M of $T_{0,m}$, $ASK(D_{T_{0,m}},M)$ is true iff $M \not\models CIRC(T_{0,m})$. From the above particular case of Lemma 2, it follows that for any $\forall\exists$-QBF F with $X = \emptyset$ and $|Y| = m$, F is valid iff $ASK(D_{T_{0,m}},M_F)$ is true. From $D_{T_{0,m}}$ one can define a polynomial advice function which depends only on m for deciding validity of $\forall\exists$-QBF-formulae with $X = \emptyset$. Since deciding the validity of such formulae is an NP-complete problem (it is just the well-known 3SAT problem), NP $\subseteq$ P/poly. □

Non-uniform complexity classes for proving lower bounds on the size of formulae in knowledge representation were used first in [13].

The above theorem shows the unfeasibility, under certain conditions, of compiling the original circumscription so that the compiled version is more effective when performing model checking. Notice that no bound is imposed on the time spent in the compilation process.

One may wonder if a stronger unconditioned theorem holds, namely, a theorem saying that there is no bounding polynomial at all, without referring to NP $\subseteq$ P/poly. We are not able to prove such a theorem. However, we can prove that such an unconditioned theorem would imply NP $\not\subseteq$ P/poly, by proving the converse of Theorem 4. Note that, since P$\subseteq$ P/poly, if NP $\not\subseteq$ P/poly then NP $\neq$ P. Hence proving the unconditioned non-existence of a compact data structure would be a result at least as strong as proving NP $\neq$ P.

Theorem 5. *If* NP $\subseteq$ P/poly *then there exists a* P*-relation* $ASK(\cdot,\cdot)$ *and a polynomial* p *such that for each propositional formula* T *there is a data structure* D_T *such that* $|D_T| \leqslant p(|T|)$ *and for any interpretation* M *of* T, $M \not\models CIRC(T)$ *if and only if* $ASK(D_T,M)$ *is true.*

Proof. For any propositional formula T, let n_T be the number of distinct propositional atoms of T. If M is an interpretation of T we denote by M also the encoding of M as a binary string of length n_T. Moreover, let T denote also an encoding of the formula T.

Let $L = \{\langle T,M\rangle \mid M$ is an interpretation of T and $M \not\models CIRC(T)\}$. Clearly, $L \in$ NP. Since NP $\subseteq$ P/poly, it follows that there exist $R \in$ P, an advice function A, and a polynomial q such that $\forall n |A(n)| \leqslant q(n)$ and

$$\forall T,M \quad \langle T,M\rangle \in L \;\Leftrightarrow\; \langle A(|\langle T,M\rangle|), \langle T,M\rangle\rangle \in R.$$

For any T, let $D_T = \langle T, A(2|T| + 2n_T)\rangle$. It is clear that there is a polynomial p such that $\forall T |D_T| \leqslant p(|T|)$. Define $ASK(D_T,M)$ as $\langle A(2|T| + 2n_T), \langle T,M\rangle\rangle \in R$. Note that from D_T it is possible to compute in polynomial time both T and $A(2|T| + 2n_T)$. Hence, $ASK(D_T,M)$ is computable in polynomial time. Moreover, $ASK(D_T,M)$ is true iff $M \not\models CIRC(T)$. In fact, for any interpretation M of T,

$$\begin{aligned} ASK(D_T,M) \text{ is true} &\Leftrightarrow \langle A(2|T| + 2n_T), \langle T,M\rangle\rangle \in R && \text{(def. of } ASK)\\ &\Leftrightarrow \langle A(|\langle T,M\rangle|), \langle T,M\rangle\rangle \in R && \text{(def. of } |\langle T,M\rangle|)\\ &\Leftrightarrow \langle T,M\rangle \in L && \text{(def. of } R \text{ and } A). \quad \Box \end{aligned}$$

3.2. *Logical equivalence*

Theorem 4 shows that, in general, compact representations of $CIRC(T)$ by data structure do not exist, unless the polynomial hierarchy collapses to the second level. Moreover, Theorem 5 says that this result cannot be improved, in the sense of proving the non existence of compact representations unconditionally, unless we are able to settle some very hard conjecture like P $\neq$ NP. Nevertheless, we could still hope to unconditionally prove the non-existence, in general, of compact representations by a very specialized kind of data structures such as propositional formulae. However, the next result shows that this is at least as hard as to solve another old conjecture, namely $NC^1 \neq$ NP.

Theorem 6. *There exists a polynomial* p *such that for each propositional formula* T *there is a formula* T', *over the same alphabet of* T, *which is logically equivalent to* $CIRC(T)$ *and whose size is bounded by* $p(|T|)$, *if and only if* NP $\subseteq NC^1$/poly.

Proof. *Only if part.* We need the following notations. For any integer $k > 0$, let $V_k = \{v_1, \ldots, v_k\}$ be a fixed alphabet of propositional atoms. For any binary string v of length k, let $M_v = \{v_i \mid i\text{th symbol of } v \text{ is } 1\}$.

We will prove that for any $L \in \mathrm{NP}$ a polynomial q exists such that, for any k, there is a propositional formula ψ_k over the alphabet V_k, of size at most $q(k)$, such that $v \in L$ if and only if $M_v \models \psi_k$, for all v of length k. From a result of Spira [29], showing that every polynomial-size formula can be converted into an equivalent boolean circuit of logarithmic depth, it follows that $L \in \mathrm{NC}^1/\mathrm{poly}$, and thus $\mathrm{NP} \subseteq \mathrm{NC}^1/\mathrm{poly}$.

Let L be any language in NP. By the Cook–Levin theorem, for any k, there is a 3CNF formula φ_k, over an alphabet Y with $V_k \subseteq Y$, whose size is polynomially bounded in k, and such that, for every v of length k,

$$v \in L \Leftrightarrow \exists N \subseteq Y - V_k:\ M_v \cup N \models \varphi_k.$$

Let $T_{0,m}$ be the formula constructed in the proof of Lemma 2 with respect to the alphabets $X = \emptyset$ and Y with $|Y| = m$. Recall that the alphabet of $T_{0,m}$ is, in this case, the set $Y \cup C \cup D \cup \{u\}$, where C and D are two sets of new atoms which are both in one-to-one correspondence with the set of all the three-literals clauses of Y, and u is a distinguished new atom. By Lemma 2 it follows that for any 3CNF formula E over the alphabet Y, E is satisfiable if and only if $M_E \not\models CIRC(T_{0,m})$, where $M_E = Y \cup C_E \cup \overline{D_E} \cup \{u\}$, $C_E = \{c \in C \mid c \text{ corresponds to a clause of } E\}$, and $\overline{D_E} = \{d \in D \mid d \text{ does not correspond to a clause of } E\}$.

For any v of length k, let $E_v = \alpha_1 \wedge \cdots \wedge \alpha_k$, where α_i is equal to $v_i \vee v_i \vee v_i$ if the ith symbol of v is 1, and is equal to $\neg v_i \vee \neg v_i \vee \neg v_i$ otherwise. Observe that M_v is the unique model of E_v. Let $\varphi_v = E_v \wedge \varphi_k$. It holds that $v \in L$ if and only if φ_v is satisfiable. Therefore, $v \in L$ if and only if $M_{\varphi_v} \not\models CIRC(T_{0,m})$. By hypothesis there exists a propositional formula T', over the same alphabet of $T_{0,m}$, which is equivalent to $CIRC(T_{0,m})$ and whose size is bounded by $p(|T_{0,m}|)$. For every $i = 1, \ldots, k$, let a_i and $\overline{a_i}$ be the atoms of C that correspond to the clauses $v_i \vee v_i \vee v_i$ and $\neg v_i \vee \neg v_i \vee \neg v_i$ respectively, and similarly for b_i and $\overline{b_i}$ as atoms of D. Moreover, let $C_1 = \{a_i, \overline{a_i} \mid i = 1, \ldots k\}$ and $D_1 = \{b_i, \overline{b_i} \mid i = 1, \ldots, k\}$. Let ψ_k be the formula over the alphabet V_k obtained from $\neg T'$ by substituting *true* for every occurrence of atoms in $M_{\varphi_v} - (C_1 \cup D_1)$, *false* for every occurrence of the remaining atoms that do not belong to $(C_1 \cup D_1)$, v_i for every occurrence of atoms a_i and $\overline{b_i}$, and $\neg v_i$ for every occurrence of atoms $\overline{a_i}$ and b_i. It is immediate to verify that $M_v \models \psi_k$ if and only if $M_{\varphi_v} \models \neg T'$. Hence, $v \in L$ if and only if $M_v \models \psi_k$.

If part. For any propositional formula T, let n_T be the number of propositional atoms of T. If M is an interpretation of T we denote by M also the encoding of M as a binary string of length n_T. Moreover, T will also denote an encoding of the formula T.

Since $L = \{\langle T, M \rangle \mid M \text{ is an interpretation of } T \text{ and } M \not\models CIRC(T)\}$ belongs to NP and $\mathrm{NP} \subseteq \mathrm{NC}^1/\mathrm{poly}$, there exists a family $\{C_n\}$ of boolean circuits such that, for any n, C_n has n inputs, computes $L \cap \{0,1\}^n$, and has depth at most $k \log n$ for some constant k. Let T be any propositional formula and let $m = 2|T| + 2n_T$. Circuit C_m, on input $\langle T, M \rangle$, outputs 1 if and only if $M \not\models CIRC(T)$. It is easy to see that C_m can be converted to a circuit C with n_T inputs, depth at most that of C_m, and such that C on input M outputs 1 if and only if $M \models CIRC(T)$. Since any circuit of depth d can be converted to a formula of size at most 2^d, there is a formula T', over the same alphabet of T, of size at most $2^{2k}|T|^k$, which is equivalent to $CIRC(T)$. □

3.3. Query equivalence

Theorem 6 states that the size of any formula T' such that $T' \equiv CIRC(T)$ grows faster than any polynomial as the size of T increases. What happens if we give up logical equivalence and go for the weaker "query equivalence"? Using a similar technique, we are able to prove that a polynomial-sized T' query-equivalent to T exists if and only if NP $\subseteq$ coNP/poly.

Theorem 7. *There exists a polynomial p such that for each propositional formula T there is a formula T' over an extended alphabet, whose size is bounded by $p(|T|)$, such that $\{Q \mid T' \models Q\} = \{Q \mid CIRC(T) \models Q\}$ where Q is any formula over the alphabet of T, if and only if* NP $\subseteq$ coNP/poly.

Proof. We exploit the fact that NP $\subseteq$ coNP/poly if and only if $\Pi_2^p \subseteq$ coNP/poly (see [31]), and refer to the latter inclusion in what follows.

Only if part. We show that, under the above hypothesis, the Π_2^p-complete problem of deciding validity of $\forall\exists$-QBF formulae belongs to coNP/poly. By Lemma 2, for any n, m, there is a formula $T_{n,m}$ such that for any $\forall\exists$-QBF F of the form (2) it holds

$$F \text{ is valid} \Leftrightarrow CIRC(T_{n,m}) \models Q_F.$$

By hypothesis, in correspondence with $T_{n,m}$, there exists a polynomial-size propositional formula $T'_{n,m}$ such that $\{Q \mid T'_{n,m} \models Q\} = \{Q \mid CIRC(T_{n,m}) \models Q\}$ where Q is any formula over the alphabet of $T_{n,m}$. This implies that, for any $\forall\exists$-QBF formula F with n $\forall$-quantifiers and m $\exists$-quantifiers,

$$F \text{ is valid} \Leftrightarrow \langle T'_{n,m}, F\rangle \in R,$$

where $R = \{\langle T, F\rangle \mid T \models Q_F\}$ is clearly a language in coNP. It is very easy to verify that the above implies that the $\forall\exists$-QBF problem belongs to coNP/poly, and hence $\Pi_2^p \subseteq$ coNP/poly.

If part. For any propositional formula T, let n_T be the number of propositional atoms of T. If M is an interpretation of T we denote by M also the encoding of M as a binary string of length n_T. Moreover, T will also denote an encoding of the formula T.

We need the following notations. For any integer $n > 0$, let $Z_n = \{z_1, \ldots, z_n\}$ be a fixed alphabet of propositional atoms. For any binary string z of length n, let $M_z = \{z_i \mid i\text{th symbol of } z \text{ is } 1\}$.

Observe that $\Pi_2^p \subseteq$ coNP/poly implies that coNP $\subseteq$ NP/poly. Thus, since $L = \{\langle T, M\rangle \mid M \text{ is an interpretation of } T \text{ and } M \models CIRC(T)\}$ belongs to coNP and coNP $\subseteq$ NP/poly, there exists an $R \in$ NP, an advice function A and a polynomial q, such that $|A(n)| \leqslant q(n)$ for all n, and

$$\forall T, M \quad \langle T, M\rangle \in L \Leftrightarrow \langle A(|\langle T, M\rangle|),\ \langle T, M\rangle\rangle \in R.$$

By the Cook–Levin theorem, for any n, there is a 3CNF formula φ_n, over the alphabet $Z_n \cup Y_n$, whose size is bounded by $r(n)$ for some fixed polynomial r, and such that, for every z of length n,

$$z \in R \Leftrightarrow \exists N \subseteq Y_n\colon M_z \cup N \models \varphi_n.$$

Let T be any propositional formula, let X be the alphabet of T, and let $m = 2|A(2|T| + 2n_T)| + 2(2|T| + 2n_T)$. Note that $|\langle A(|\langle T, M\rangle|), \langle T, M\rangle\rangle| = m$ for any interpretation M of T, and $m \leqslant 2q(4|T|) + 8|T|$. Since T and $A(2|T| + 2n_T)$ are fixed, we can easily convert φ_m into a formula T' over the alphabet $X \cup Y_m$, whose size is at most that of φ_m, and such that, for any interpretation M of T,

$$M \models CIRC(T) \Leftrightarrow \exists N \subseteq Y_m\colon M \cup N \models T'.$$

Thus, the size of T' is at most $r(2q(4|T|) + 8|T|)$ and, for any formula Q over the alphabet X of T, $T' \models Q$ if and only if $CIRC(T) \models Q$. □

Note that, in the case of the query equivalence, the existence of compact representations by data structures is equivalent to the existence of compact representations by propositional formulae. This is in contrast to the case of the model and logical equivalences in which the two kinds of compact representations do not seem equivalent. In fact, the former is possible if and only if NP $\subseteq$ P/poly, while the latter is possible if and only if the stronger condition NP $\subseteq$ NC^1/poly holds. This leads us to conjecture that there may exist subclasses of formulae whose circumscriptions admit compact representations by data structures but not by propositional formulae.

4. Analysis of the results

In this section we analyze the generality of our results and their impact on a topic strictly related to circumscription, namely *closed-world reasoning*. Closed-world reasoning is a collection of ideas and definitions developed in the database field for addressing the issue of reasoning using *lack* of information. Motivations for closed-world reasoning are very close in spirit to those behind circumscription. The main difference is that, while the circumscription of a propositional formula T is defined as a second-order formula (cf. formula (1)), making the closure of T amounts to adding to T new propositional formulae according to some criterion (cf. formula (4)). Despite these syntactical differences, the two approaches are strictly related at the semantic level.

We first recall two different proposals of Closed-World Assumption (CWA): Generalized CWA (GCWA) and Extended Generalized CWA (EGCWA). Then we show how the proof of our main theorem can be used to define theories whose closure under EGCWA has super-polynomial size. Finally, we discuss the generality of our technique ("is it always possible to exploit intractability results to show non-compactability?") and take GCWA as an example of a closure operator which is compactable. The reason why compactability of GCWA is interesting is that the two closure operators have similar time complexity: if T, q are propositional formulae and M is an interpretation, testing $GCWA(T) \models q$ and $EGCWA(T) \models q$ are both Π_2^p-hard problems and testing $M \models GCWA(T)$ and $M \models EGCWA(T)$ are both coNP-hard problems.

4.1. Closed-world reasoning

Generalized Closed World Assumption $GCWA(T)$ of a propositional formula T [21] is defined as follows (K is an atom and B is a clause – possibly empty – in which only positive literals occur):

$$T \cup \{\neg K \mid \forall B.\ T \not\models B \Rightarrow T \not\models B \vee K\}. \tag{4}$$

All models of $CIRC(T)$ are models of $GCWA(T)$, but not the other way around [21, Theorem 2].

A semantically more clear formalism for treatment of incomplete information is *Extended Generalized Closed World Assumption* $EGCWA(T)$ [32]. Its definition is like (4), except that K is now an arbitrary conjunction of atoms. Such conjunctions are called "free-for-negation" for T. Observe that in a reasonable representation of $EGCWA(T)$, only minimal conjunctions of atoms need to be added to T, where a free-for-negation conjunction K is minimal if any subconjunction of K is not free-for-negation. The models of $EGCWA(T)$ are exactly the models of $CIRC(T)$ [32], therefore Theorem 6 says that the size of $EGCWA(T)$ is not likely to be polynomial in $|T|$, as $|T|$ increases.

It is worth noting that $EGCWA(T)$ might be a much smarter representation of $CIRC(T)$ than listing all minimal models of T. As an example, let $a_1, \ldots, a_n$, $b_1, \ldots, b_n$ be distinct atoms and T be $(a_1 \vee b_1) \wedge \cdots \wedge (a_n \vee b_n)$. $EGCWA(T)$ is $T \wedge (\neg a_1 \vee \neg b_1) \wedge \cdots \wedge (\neg a_n \vee \neg b_n)$. The simple-minded representation of $CIRC(T)$ is the disjunction of all possible conjunctions $x_1 \wedge \cdots \wedge x_n \wedge \neg y_1 \wedge \cdots \wedge \neg y_n$, where for all i $(1 \leqslant i \leqslant n)$, x_i is a member of $\{a_i, b_i\}$, and y_i is the other member. The latter representation has clearly exponential size.

4.2. Large instances of EGCWA

We now are able to reveal an infinite set of T's, where – even when considering minimal free-for-negation conjunctions – the size of $EGCWA(T)$ is superpolynomial. Such formulae are inspired by the one built in the proof of Lemma 2. This is, to the best of our knowledge, the first example proving that such a smart technique for representing propositional circumscription outputs, in the worst case, a theory of superpolynomial size.

We use the alphabets of atoms $X = \{x_1, \ldots, x_n\}$, $C = \{c_1^+, c_1^-, \ldots, c_n^+, c_n^-\}$, $D = \{d_1^+, d_1^-, \ldots d_n^+, d_n^-\}$ and a new distinct atom u. We define a propositional formula T_n over these alphabets, with the help of two formula. Γ_n, Σ_n, which are analogous to $\Gamma_{n,m}, \Sigma_{n,m}$ of Lemma 2. To simplify notation, we use $a \neq b$ as a shorthand for $(a \vee b) \wedge (\neg a \vee \neg b)$. Let

$$\Gamma_n = \bigwedge \{x_i \vee \neg c_i^+ \mid 1 \leqslant i \leqslant n\} \wedge \bigwedge \{\neg x_i \vee \neg c_i^- \mid 1 \leqslant i \leqslant n\},$$

$$\Sigma_n = \bigwedge \{c_i^+ \neq d_i^+ \mid 1 \leqslant i \leqslant n\} \wedge \bigwedge \{c_i^- \neq d_i^- \mid 1 \leqslant i \leqslant n\},$$

$$T_n = \Sigma_n \wedge [(u \wedge x_1 \wedge \cdots \wedge x_n) \vee \Gamma_n].$$

Notice that the size of $X \cup C \cup D$ is $5n$, and the size of T_n is $O(n)$. Given a subset E of X, we define $C_E = \{c_i^+ \mid x_i \in E\} \cup \{c_i^- \mid x_i \notin E\}$, and similarly $D_E = \{d_i^+ \mid x_i \in E\} \cup \{d_i^- \mid x_i \notin E\}$. Moreover, let $\overline{D_E} = D - D_E = \{d_i^+ \mid c_i^+ \notin C_E\} \cup \{d_i^- \mid c_i^- \notin C_E\}$.

Lemma 8. *Let T_n be as above, and for any $E \subseteq X$ let $M_E = E \cup C_E \cup \overline{D_E}$. M_E is a minimal model of T_n.*

Proof. Since $E \cup C_E$ satisfies Γ_n, and $C_E \cup \overline{D_E}$ satisfies Σ_n, M_E is a model of T_n. Suppose $M \subset M_E$ is also a model of T_n, and let $M_C = M \cap C$, $M_D = M \cap D$, and $M_X = M \cap X$. Since M satisfies Σ_n, if $M_C \subset C_E$ and $M_D \supset D_E$, and vice versa if $M_D \subset D_E$ then $M_C \supset C_E$. Hence, $M_C = C_E$ and $M_D = D_E$. Therefore, it should be $M_X \subset E$, so let $x_i \in E - M_X$. By definition of $C_E, c_i^+ \in C_E$, hence the clause $x_i \vee \neg c_i^+$ is not satisfied by M, hence Γ_n is not satisfied. Since also the conjunction $u \wedge x_1 \wedge \cdots \wedge x_n$ is not satisfied, we conclude that any such M is not a model of T_n, therefore M_E is minimal. □

We exploit the previous property in the proof of our next theorem. To simplify notation, we denote by $\overline{D_E} \wedge u$ the formula obtained as a conjunction of all atoms in the set $\overline{D_E}$ and u.

Theorem 9. *Let E be any subset of X; then $\overline{D_E} \wedge u$ is a minimal free-for-negation formula for T_n.*

Proof. First of all, we show by contradiction that $\overline{D_E} \wedge u$ is free-for-negation: Assume there exists a minimal model M of T_n satisfying $\overline{D_E} \wedge u$. Now, M must also satisfy $x_1, \ldots, x_n$, because otherwise M satisfies Γ_n, and $M - \{u\}$ would be a model, contradicting minimality of M. Therefore $M \supseteq \overline{D_E} \cup \{x_1, \ldots, x_n\} \cup \{u\}$.

Let $M_C = M \cap C$, $M_D = \cap D$, so that M can be partitioned as $M_C \cup M_D \cup \{x_1, \ldots, x_n\} \cup \{u\}$. Since M satisfies $\overline{D_E} \wedge u$, then $M_D \supseteq \overline{D_E}$. Then $M_C \subseteq C_E$, since M satisfies also Σ_n. Now let $N = E \cup M_C \cup M_D$. We show that N satisfies T_n. First, N satisfies Σ_n because it gives the same interpretation as M to literals in C and D. We now show that N satisfies each clause of Γ_n.

1. Let $x_i \in E$. Then the clause $x_i \vee \neg c_i^+$ is satisfied by N. By definition of C_E, $c_i^- \notin C_E$. Since $M_C \subseteq C_E$, also $c_i^- \notin M_C$. Hence the clause $\neg x_i \vee \neg c_i^-$ is satisfied too.

2. Let $x_i \notin E$. Then the clause $\neg x_i \vee \neg c_i^-$ is satisfied. By definition of C_E, this time $c_i^+ \notin C_E$, so $c_i^+ \notin M_C$. Hence the clause $x_i \vee \neg c_i^+$ is satisfied.
Since $N \subset M$, M is not minimal, contradicting the hypothesis. We conclude that $\overline{D_E} \wedge u$ is free-for-negation.

We now show that if we remove one conjunct from $\overline{D_E} \wedge u$, the resulting formula is not free-for-negation, thus showing that $\overline{D_E} \wedge u$ is a minimal free-for-negation formula.

First observe that if we remove u, then $\overline{D_E}$ (considered as a conjunction) is not free-for-negation because M_E satisfies it, and by Lemma 8 M_E is a minimal model. Secondly, we prove that if we take out a literal $d_i^- \in D$ from $\overline{D_E} \wedge u$ the resulting formula is not free-for-negation. In fact, let $M = (\overline{D_E} - \{d_i^-\}) \cup C_E \cup \{c_i^-\} \cup \{u\} \cup X$ be an interpretation satisfying the smaller conjunction. It holds that M satisfies Σ_n, hence M is also a model of T_n because it satisfies $u \wedge x_1 \wedge \cdots \wedge x_n$. We now show that M is also a minimal model of T_n, by proving that for any model N such that $N \subseteq M$, it results $N = M$. Since N satisfies Σ_n, if $N \cap C \subset M \cap C$ then $N \cap D \supset M \cap D$. Hence to be $N \subseteq M$, it must be $N \cap C = M \cap C$, and also $N \cap D = M \cap D$. Therefore N and M can differ at most on $X \cup \{u\}$. But notice that both c_i^+ and c_i^- belong to M, hence they belong to N too. Observe that Γ_n contains the two clauses $x_i \vee \neg c_i^+$ and $\neg x_i \vee \neg c_i^-$, which cannot both be satisfied by N, for any possible interpretation of x_i. Hence N cannot satisfy Γ_n. Therefore to satisfy T_n, N must satisfy $u \wedge x_1 \wedge \cdots \wedge x_n$. But this implies $N = M$. $\square$

Since there are exponentially many subsets of X, there are also exponentially many distinct free-for-negation conjuncts. So $EGCWA(T_n)$ contains at least 2^n clauses, each clause having $n+1$ disjuncts. Therefore $|EGCWA(T_n)|$ is $\Omega(n2^n)$, while $|T_n|$ is $O(n)$. Observe also that T_n could be rewritten as a 3CNF-formula (by distributing the conjunction $u \wedge x_1 \wedge \cdots \wedge x_n$ over Γ_n) having $O(n^2)$ clauses. Hence, even when T_n is in 3CNF, the above line of reasoning yields a super-polynomial lower bound for the size of $EGCWA(T_n)$.

4.3. Generality of main result

In Theorems 4, 6 and 7 we used the reduction of NP-hard problems – deciding whether $M \not\models CIRC(T)$ or not – and Π_2^p-hard ones – deciding whether $CIRC(T) \models Q$ or not – to show that a polynomial-size representation of $CIRC(T)$ is unlikely to exist, regardless of the effort we spend for doing the "compilation" of $CIRC(T)$.

The technique employed readily applies to a much wider spectrum of reasoning problems in knowledge bases. Using well-known reductions of circumscription into other reasoning problems, we were able to extend our result to the explicit representations of disjunctive databases under the stable [9] or well-founded semantics [30] as extended by Przymusinski [25], skeptical reasoning in default logic [27] and autoepistemic logics [18]. Furthermore, in [5] we apply this method to skeptical and credulous reasoning in (fragments of) default logic, while in [2] we analyze the space complexity of most operators for belief revision and update introduced in the literature.

In this paper we have shown that the existence of a polynomial-size representation of $CIRC(T)$ is unlikely to exist, regardless of time needed and equivalence criterion adopted. In [2] we presented belief revision operators which do not admit a compact representation if we require model-equivalence, but they do if we only go for query-equivalence. Therefore, the three equivalence criteria have different impacts on the existence of compact representations and it might very well be the case that, for restricted languages, circumscription admits a compact representation w.r.t. some equivalence criteria but not w.r.t. all of them.

However, the technique is not applicable to all reductions of NP-hard problems in knowledge representation. As an example, we now show that model checking under GCWA is coNP-hard, but the closure of a theory under GCWA has always a representation of polynomial size.

The reduction for GCWA rephrases the one showing that $M \not\models CIRC(T)$ is NP-hard. Given any formula F on alphabet $X = \{x_1, \ldots x_n\}$, and another atom $u \notin X$, define $T = (F \wedge \neg u) \vee (u \wedge x_1 \wedge \cdots \wedge x_n)$. Let $M = \{u\} \cup X$. It can be shown that F is satisfiable iff $M \not\models GCWA(T)$. Hence model checking under GCWA is coNP-hard.

Nevertheless, there exists a simple polynomial-size explicit representation of $GCWA(T)$: for every atom K, simply decide if $\neg K$ must be added or not to T, and if so add it. Hence, if T is fixed then $GCWA(T)$ can be "compiled", once and for all. Observe that this does not prove NP $\subseteq$ P/poly, since the compilation of $GCWA(T)$ depends on T itself, and not only on its size.

5. Conclusions and open problems

In this paper we have investigated the size of representations equivalent to the circumscription $CIRC(T)$ of a propositional formula T, taking into account three different definitions of equivalence. We have found that necessary and sufficient condition for the existence of polynomial-size representations equivalent to $CIRC(T)$ in the three cases is inclusion of NP into three different non-uniform complexity classes. As such conditions imply the collapse of the polynomial hierarchy at a low level, it is likely that the size of a propositional representation of $CIRC(T)$ grows faster than any polynomial as the size of T increases.

We want to point out that we identified the exact conditions under which compact representations exist. These results cannot be easily strengthened as proving the existence of compact representations implies a collapse in the polynomial hierarchy, while, for example, proving their non-existence under the model equivalence criterium implies that P $\neq$ NP.

This result has a negative side: It is unfeasible to (off-line) compile a knowledge base so that (on-line) reasoning under circumscription becomes easier. On the other side, our results imply that circumscription allows more compact representation of knowledge. As a consequence, circumscription may be used to produce a compact

Table 1
Necessary and sufficient conditions for the existence of polynomially-sized representations equivalent to *CIRC*(T)

T' logically equivalent	$\Leftrightarrow$	$\mathrm{NP} \subseteq \mathrm{NC}^1/\mathrm{poly}$	Theorem 6
D model equivalent	$\Leftrightarrow$	$\mathrm{NP} \subseteq \mathrm{P}/\mathrm{poly}$	Theorems 4, 5
T' query equivalent	$\Leftrightarrow$	$\mathrm{NP} \subseteq \mathrm{coNP}/\mathrm{poly}$	Theorem 7

representation of some boolean functions whose propositional representation is inherently super-polynomial w.r.t. the number of boolean variables.

The results presented in this paper are summarized in Table 1.

Some interesting questions that we did not consider in the present work are briefly listed:

1. Are there syntactically restricted classes of formulae for which, e.g., polynomial-sized query equivalent formulae exist, while logically equivalent formulae do not?
2. Why formalisms with similar time complexity (e.g., GCWA and EGCWA) have different compactability properties?
3. The degree of undecidability of infinitary propositional (sentential) circumscription has been analyzed in [24], where it is proven that the inference problem is more difficult than the corresponding problem in infinitary propositional logic. What is the impact of such a result from the point of view of the size of the representation?

Acknowledgements

The authors are grateful to Pierluigi Crescenzi for an interesting discussion on the non-uniform polynomial hierarchy and to Phokion Kolaitis for suggesting to us to investigate "inverse" results, such as Theorem 5.

References

[1] M. Cadoli, The complexity of model checking for circumscriptive formulae, *Inform. Process. Lett.* **44** (1992) 113–118.
[2] M. Cadoli, F.M. Donini, P. Liberatore and M. Schaerf, The size of a revised knowledge base, in: *Proc. 14th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems* (*PODS-95*) (1995) 151–162.
[3] M. Cadoli, F.M. Donini and M. Schaerf, Is intractability of non-monotonic reasoning a real drawback?, in: *Proc. 12th National Conf. on Artifical Intelligence* (*AAAI-94*) (1994) 946–951. Extended version as RAP.09.95 DIS, Univ. of Roma "La Sapienza", 1995.
[4] M. Cadoli, F.M. Donini and M. Schaerf, On compact representations of propositional circumscription, in: *12th Symp. on Theoretical Aspects of Computer Science* (*STACS-95*) (1995) 205–216. Extended version as RAP.14.95 DIS, Univ. of Roma "La Sapienza", July 1995.
[5] M. Cadoli, F.M. Donini and M. Schaerf, Is intractability of non-monotonic reasoning a real drawback?, *Artifical Intelligence J.* **88** (1996) 215–251.

[6] P. Doherty, W. Lukaszewicz and A. Szalas, Computing circumscription revisited: a reduction algorithm, in: *Proc. 14th Internat. Joint Conf. on Artifical Intelligence* (*IJCAI-95*) (1995) 1502–1508.

[7] T. Eiter and G. Gottlob, Propositional circumscription and extended closed world reasoning are Π_2^p-complete, *Theoret. Comput. Sci.* **114** (1993) 231–245.

[8] D. Gabbay and H.J. Ohlbach, Quantifier elimination in second-order predicate logic, in: *Proc. 3rd Internat. Conf. on the Principles of Knowledge Representation and Reasoning* (*KR-92*) (1992) 425–435.

[9] M. Gelfond and V. Lifschitz, The stable model semantics for logic programming, in: *Proc. 5th Logic Programming Symp.* (MIT Press, Cambridge, MA, 1988) 1070–1080.

[10] M. Gelfond, H. Przymusinska and T. Przymusinsky, On the relationship between circumscription and negation as failure, *Artificial Intelligence J.* **38** (1989) 49–73.

[11] D.S. Johnson, A catalog of complexity classes, in: J. van Leeuwen, ed., *Handbook of Theoretical Computer Science*, Vol. A, Ch. 2 (Elsevier, Amsterdam, 1990).

[12] R.M. Karp and R.J. Lipton, Some connections between non-uniform and uniform complexity classes, in: *Proc. 12th ACM Symp. on Theory of Computing* (*STOC-80*) (1980) 302–309.

[13] H.A. Kautz and B. Selman, Forming concepts for fast inference, in: *Proc. 10th Natl. Conf. on Artifcial Intelligence* (*AAAI-92*) (1992) 786–793.

[14] P.G. Kolaitis and C.H. Papadimitriou, Some computational aspects of circumscription, *J. ACM* **37** (1990) 1–14.

[15] T. Krishnaprasad, On the computability of circumscription, *Inform. Process. Lett.* **27** (1988) 237–243.

[16] H.J. Levesque, Making believers out of computers, *Artifical Intelligence J.* **30** (1986) 81–108.

[17] V. Lifschitz, Computing circumscription, in: *Proc. 9th Internat. Joint Conf. on Artificial Intelligence* (*IJCAI-85*) (1985) 121–127.

[18] W. Marek and M. Truszczyński, Autoepistemic logic, *J. ACM* **38** (1991) 588–619.

[19] J. McCarthy, Circumscription – a form of non-monotonic reasoning, *Artifical Intelligence J.* **13** (1980) 27–39.

[20] J. McCarthy, Applications of circumscription to formalizing common-sense knowledge, *Aritifical Intelligence J.* **28** (1986) 89–116.

[21] J. Minker, On indefinite databases and the closed world assumption, in: *Proc. 6th Internat. Conf. on Automated Deduction* (*CADE-82*) (1982) 292–308.

[22] Y. Moses and M. Tennenholtz, Off-line reasoning for on-line efficiency, in: *Proc. 13th Internat. Joint Conf. on Artifical Intelligence* (*IJCAI-93*) (1993) 490–495.

[23] A. Nerode, R.T. Ng and V.S. Subrahmanian, Computing circumscriptive databases. I: Theory and algorithms, *Inform. and Comput.* **116** (1995) 58–80.

[24] M. Papalaskari and S. Weinstein, Minimal consequence in sentential logic, *J. Logic Programming* **9** (1990) 19–31.

[25] M. Przymusinski, Stable semantics for disjunctive programs, *New Generation Comput.* **9** (1991) 401–424.

[26] A. Rabinov, A generalization of collapsible cases of circumscription, *Artificial Intelligence J.* **38** (1989) 111–117.

[27] R. Reiter, A logic for default reasoning, *Artifical Intelligence J.* **13** (1980) 81–132.

[28] B. Selman and H.A. Kautz, Knowledge compilation using Horn approximations, in: *Proc. 9th Natl. Conf. on Artifical Intelligence* (*AAAI-91*) (1991) 904–909.

[29] P.M. Spira, On time hardware complexity tradeoffs for boolean functions, in: *Proc. 4th Hawaii Symp. on Systems Science* (1971) 525–527.

[30] A. van Gelder, K.A. Ross and J.S. Schlipf, The well-founded semantics for general logic programs, *J. ACM* **38** (1991) 620–650.

[31] C.K. Yap, Some consequences of non-uniform conditions on uniform classes, *Theoret. Comput. Sci.* **26** (1983) 287–300.

[32] A. Yahya and L.J. Henschen, Deduction in non-Horn databases, *J. Autom. Reasoning* **1** (1985) 141–160.