ELSEVIER
Artificial Intelligence 119 (2000) 1–17
Artificial Intelligence
www.elsevier.com/locate/artint

Semantical and computational aspects of Horn approximations ☆

Marco Cadoli [a,*], Francesco Scarcello [b,1]

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

[b] *Dipartimento di Elettronica, Informatica e Sistemistica, Università della Calabria, Via P. Bucci, I-87030 Rende (CS), Italy*

Received 22 December 1997; received in revised form 23 November 1999

Abstract

Selman and Kautz proposed a method, called *Horn approximation*, for speeding up inference in propositional Knowledge Bases. Their technique is based on the *compilation* of a propositional formula into a pair of Horn formulae: a Horn Greatest Lower Bound (GLB) and a Horn Least Upper Bound (LUB). In this paper we focus on GLBs and address two questions that have been only marginally addressed so far:

(1) what is the semantics of the Horn GLBs?
(2) what is the exact complexity of finding them?

We obtain semantical as well as computational results. The major semantical result is: The set of minimal models of a propositional formula and the set of minimum models of its Horn GLBs are the same. The major computational result is: Finding a Horn GLB of a propositional formula in CNF is NP-equivalent.

Keywords: Knowledge compilation; Knowledge approximation; Computational complexity; Horn formulae

1. Introduction

Selman and Kautz proposed a method [15,23,24], called *Horn approximation*, for speeding up inference in propositional Knowledge Bases. Propositional inference is the problem of checking whether $\Sigma \models \alpha$ holds, where Σ and α are propositional formulae.

☆ This is an extended and revised version of [4].

* Corresponding author. Email: cadoli@dis.uniroma1.it.

1 Email: scarcello@deis.unical.it.

0004-3702/00/$ – see front matter
PII: S0004-3702(00)00010-2

The starting point of their technique stems from the fact that inference for general propositional formulae is co-NP-complete—hence polynomially unfeasible—while it is doable in polynomial time when Σ is a Horn formula. The fascinating question they address is the following: is it possible to *compile* a propositional formula Σ into a Horn one Σ' so that a significant amount of the inferences that are performed under Σ can be performed under Σ' in polynomial time?

Selman and Kautz notice that there exist two different ways of doing such a compilation. In the first case the compiled formula satisfies the relation $\Sigma' \models \Sigma$, or equivalently $\mathcal{M}(\Sigma') \subseteq \mathcal{M}(\Sigma)$—where $\mathcal{M}(\Phi)$ denotes the set of models of the formula Φ. For this reason Σ' is called a *Horn lower bound*—or LB—of Σ. As an example—taken from [23]—let Φ be the formula

$$(man \rightarrow person) \wedge (woman \rightarrow person) \wedge (man \vee woman).$$

The formula $\Phi_{lb} = man \wedge woman \wedge person$ is a Horn LB of Φ.

The second form of compilation is dual. The compiled version of Σ is a Horn formula Σ' that satisfies the relation $\Sigma \models \Sigma'$, or equivalently $\mathcal{M}(\Sigma) \subseteq \mathcal{M}(\Sigma')$. Σ' is called a *Horn upper bound*—or UB—of Σ. Returning to the previous example, the formula $\Phi_{ub} = (man \rightarrow person) \wedge (woman \rightarrow person)$ is a Horn UB of Φ.

The importance of having compiled forms of a Knowledge Base is in that sometimes we can use them for providing a quick answer to an inference problem. As an example, if we are faced with the problem of checking $\Sigma \models \alpha$, we may benefit from the fact that for any Horn LB Σ_{lb} of Σ, $\Sigma_{lb} \not\models \alpha$ implies $\Sigma \not\models \alpha$. Σ_{lb} is therefore a *complete approximation* of Σ. Dually, a Horn UB Σ_{ub} is a *sound approximation* of Σ, since $\Sigma_{ub} \models \alpha$ implies $\Sigma \models \alpha$.

Selman and Kautz notice that some complete approximations are better than others. In the previous example, both $\Phi_{lb1} = man \wedge woman \wedge person$ and $\Phi_{lb2} = man \wedge person$ are Horn LBs of Φ. Φ_{lb2} seems to be a better approximation than Φ_{lb1}, since $\mathcal{M}(\Phi_{lb1}) \subset \mathcal{M}(\Phi_{lb2}) \subset \mathcal{M}(\Phi)$, hence the former is in a precise sense "closer" to Φ than the latter. This consideration leads to the notion of a *Horn greatest lower bound*—or GLB—of a formula (cf. forthcoming Definition 1).

The same argument can be done for Horn upper bounds: in our example both $\Phi_{ub1} = (man \rightarrow person) \wedge (woman \rightarrow person)$ and $\Phi_{ub2} = person$ are Horn UBs of Φ, but $\mathcal{M}(\Phi) \subset \mathcal{M}(\Phi_{ub2}) \subset \mathcal{M}(\Phi_{ub1})$, hence Φ_{ub2} is a better approximation of Φ. The definition of *Horn least upper bound*—or LUB—of a formula can be found in [23,24].

Selman and Kautz's proposal is to approximate inference with respect to a propositional formula Σ by using its Horn GLBs and (the provably unique) LUB. Inference from approximations could be either unsound or incomplete. In other words, it is possible to give fast answers exploiting the approximations, or, in the worst case, give "don't know" answers. In the latter case, it is possible to spend more time and use a general inference procedure to determine the answer directly from the original formula. However, the general inference procedure could still use the approximations to prune its search space (see [23, p. 905]). It is also important to notice that Horn GLBs and LUBs can be computed off-line, hence this form of approximate reasoning is actually a *compilation*. An empirical evaluation of the reliability of the conclusions reached with the bounds, and of the computational savings they offer, is provided in [16].

As noted in [23,24], the search problem of finding a GLB is NP-hard. Anyway, as noted by Selman and Kautz, since approximations could be computed off-line, the computational cost of finding them will be amortized over the total set of subsequent queries to the Knowledge Base. In their work they propose an algorithm for finding a Horn GLB which runs in exponential time, thus leaving open the question of what is the exact complexity of the problem.

Apart from the already cited papers [15,16,23,24], the technique of Horn compilation has attracted notable interest among researchers [2,7,10,12,13,17,18,22]. A brief overview of some of their work follows. In [13] the problem of finding small-size Horn approximations of both kinds is addressed. Using learning techniques, the authors study how the knowledge of a sequence of queries to Σ may help in the design of a Horn UB and a Horn LB—not the LUB and a GLB—with a low probability of giving indefinite answers. In [22] the author performs an analysis of the computational cost of calculating the number of models of Horn approximations. In [12] the focus is on complexity of the problem of recomputing Horn approximations after the addition of a clause. In [18] an alternative approach to reasoning, based on characteristic models, is defined and compared to Horn approximations. In [14], an algorithm to produce Horn approximations out of a set of truth assignments is shown. Algorithms for computing the Horn LUB have been proposed in [7, 17]. In [2] an algorithm based on the Davis–Putnam procedure is proposed for computing a GLB. Finally, in [10], the complexity of finding a GLB is addressed for the special case in which Σ is a disjunction of Horn formulae.

Knowledge compilation is an interesting area in automated theorem proving and knowledge representation (cf. [5]), which is based on the idea of shifting the burden of intractability of logical reasoning to off-line computation. Most of the work done in this area is empirical: The quality of compiled Knowledge Bases is measured in terms of the percentage of true formulae they infer, and in the time savings thus obtained. Theoretical analysis is—in our opinion—as important as the empirical perspective. In particular, it is especially important to equip an approximate reasoner with the semantics of what its conclusions are.

In this paper we focus on GLBs [2] and address two important questions that have not been addressed so far:

(i) is it possible to describe Horn GLBs with a semantics that does not rely on the syntactic notion of Horn clause?

(ii) what is the exact complexity of finding a Horn GLB?

An answer to the first question shows the exact meaning of the approximate answers. An answer to the second question tells in which cases it is reasonable—from the computational point of view—to use Horn approximations.

We obtain two different kinds of results:

(1) *Semantical*:
   - Horn GLBs of Σ are closely related to models of the circumscription of Σ;
   - reasoning with respect to Horn GLBs is the same as reasoning by counterexamples using only minimal models;

[2] Similar aspects of the LUB are partially addressed in [4].

- while *skeptical* reasoning with respect to the Horn GLBs of a formula Σ is the same as ordinary reasoning with respect to Σ, *brave* reasoning with respect to the Horn GLBs of Σ is the same as reasoning with respect to the circumscription of Σ;
- compiling more knowledge does not always give better Horn GLBs.

(2) *Computational*:
- finding a Horn GLB is not intrinsically exponential, but it is "mildly" harder than solving the original inference problem. In particular:
  - (lower bound) the problem is $\mathrm{P}^{\mathrm{NP}[\mathrm{O}(\log n)]}$-hard, and
  - (upper bound) the problem is in P^{NP}.

The structure of the paper is as follows. Section 2 contains preliminary notions about complexity classes and Horn GLBs. In Section 3 we focus on semantical aspects, showing relations existing between GLBs and minimal models. In Section 4 we focus on computational aspects, showing lower bounds and the algorithm which gives the upper bound. We give some concluding remarks in Section 5.

2. Preliminaries

2.1. Complexity theory

In this subsection we give a brief overview of complexity concepts that are used throughout the paper. We refer the reader to [21] for a thorough introduction to the field of complexity.

A *decision problem* is a problem that admits a Boolean answer. For decision problems the class P is the set of problems that can be answered by a Turing machine in polynomial time. Often we refer to computations done by non-deterministic Turing machines. The class of decision problems that can be solved by a non-deterministic Turing machine in polynomial time—where it is understood that the answer is **yes** provided *at least one* of the computations done in parallel by the machine ends in an accepting state—is denoted by NP. The class of problems whose answer is always the complement of those in NP, is denoted by co-NP. Also problems in co-NP can be solved by a non-deterministic Turing machine in polynomial time, but it is understood that the answer is **yes** provided *all* the computations done in parallel by the machine end in an accepting state. The class P is obviously contained both in NP and in co-NP.

An example of a problem in NP is testing satisfiability of a propositional formula: a formula T is satisfiable iff *at least one* truth assignment M such that $M \models T$ exists. An example of a problem in co-NP is testing if a propositional formula T entails a propositional formula γ: $T \models \gamma$ iff *for all* truth assignments M it holds that $(M \models T) \Rightarrow (M \models \gamma)$. In fact propositional satisfiability (entailment) is an NP-*hard* (co-NP-*hard*) problem, i.e., "as tough as"—with respect to many-one polynomial reducibility—any problem in the class NP (co-NP). Problems in NP (co-NP) which are also NP-hard (co-NP-hard) are called NP-*complete* (co-NP-*complete*). We recall that the best algorithms known for solving either NP-complete or co-NP-complete problems require exponential time in the worst case, and that the following relations are conjectured: $\mathrm{P} \subset \mathrm{NP} \cap \text{co-NP}$, $\mathrm{NP} \neq \text{co-NP}$.

Throughout the paper we refer to a particular type of computation called computation with *oracles*. Intuitively, oracles are subroutines with unary cost. Given a complexity class C, the class P^C is the class of decision problems that can be solved in polynomial time by a deterministic machine that uses an oracle for the problems in C, i.e., a subroutine for any problem in C that can be called several times, spending just one time-unit for each call. In particular, P^{NP} is the class of decision problems that can be computed by a polynomial-time deterministic machine which can use at unary cost an oracle that answers a set of NP-complete queries (e.g., satisfiability checks). Note that, since the machine itself is polynomial-time, the cardinality of the set of queries is bound by a polynomial function. If the cardinality of the set is bound by a logarithmic function, we have the class $\mathrm{P}^{\mathrm{NP}[\mathrm{O}(\log n)]}$. Note that both NP-complete and co-NP-complete problems can be solved with *a single* call to an oracle in NP. In fact, it is conjectured that $\mathrm{NP} \cup \text{co-NP} \subset \mathrm{P}^{\mathrm{NP}[\mathrm{O}(\log n)]} \subset \mathrm{P}^{\mathrm{NP}}$. From the practical point of view, it is reasonable to think that P^{NP}-complete problems will be always harder to compute than NP-complete ones: Even if we have a good heuristic for an NP-complete problem and we can implement an oracle that gives a quick answer to it, we still have to use the oracle a polynomial number of times for solving the P^{NP}-complete problem. As for $\mathrm{P}^{\mathrm{NP}[\mathrm{O}(\log n)]}$-complete problems, they will be "mildly" harder to compute than NP-complete or co-NP-complete ones.

Some of the problems addressed in the paper are *search* problems, i.e., their answer is more complex than just a Boolean value. As an example, finding a satisfying truth assignment for a propositional formula or finding a Horn GLB are search problems. Formally, complexity classes for search problems are different from classes cited above, that refer to decision problems. To simplify notation, we use the same complexity classes for denoting both decision and search problems. In particular, if we say that a search problem is in P^C, or C*-easy*, we mean that its output can be delivered in polynomial time by a deterministic machine that uses an oracle for the problems in C. If we say that a search problem X is P^C-hard, or simply C*-hard*, we mean that any problem in P^C can be solved in polynomial time by a deterministic machine that uses an oracle for X. A search problem which is both C-easy and C-hard is said to be C*-equivalent*.

2.2. Horn GLBs

In this subsection we give the formal definitions, some examples, and basic properties of Horn lower bound and greatest lower bound of propositional theories, following [24].

All the propositional formulae we consider in this paper are assumed to be in conjunctive normal form (CNF henceforth). Formulae will be considered either as sets of clauses, or as conjunctions of clauses. The following notation is used: given a clause $\gamma = \neg b_1 \vee \cdots \vee \neg b_n \vee a_1 \vee \cdots \vee a_m$, the symbol $B(\gamma)$ denotes $b_1 \wedge \cdots \wedge b_n$, while $H(\gamma)$ denotes $a_1 \vee \cdots \vee a_m$. With such a notation, γ can be written as $B(\gamma) \rightarrow H(\gamma)$, therefore the notation "reminds us" that a clause can be seen as a rule (B stands for "body" and H stands for "head"). The notation is useful, because in most of the transformations considered in the following sections, the "body" of a clause remains unchanged. We remind that $\neg B(\gamma)$ is the disjunction $\neg b_1 \vee \cdots \vee \neg b_n$. Sometimes, $B(\gamma)$ and $H(\gamma)$ will be used to denote the corresponding sets of literals. A clause with no negative literals is said to be positive.

Models of a propositional formula will be denoted as the set of atoms occurring in the formula they map into 1. For a formula Φ, $\mathcal{M}(\Phi)$ denotes the set of models of Φ. Two formulae Σ and Π having the same models are said to be equivalent, and this is denoted as $\Sigma \equiv \Pi$. Minimal models of a propositional formula have the property that the set of atoms that they map into 1 is minimal. More formally (see [19]), given two models M, N of a formula, we write $M \subseteq N$ iff $\{x \mid M(x) = 1\} \subseteq \{x \mid N(x) = 1\}$, and we write $M \subset N$ iff the containment is strict. The models of a formula Φ that are minimal in this preorder are called the minimal models of Φ.

Definition 1 (*LB and GLB of a theory* [24]). Let Σ be a CNF formula.
- A Horn formula Σ_{lb} is a Horn LB (lower bound) of Σ if $\mathcal{M}(\Sigma_{lb}) \subseteq \mathcal{M}(\Sigma)$ (i.e., $\Sigma_{lb} \models \Sigma$).
- A Horn formula Σ_{glb} is a Horn GLB (greatest lower bound) of Σ if there exists no Horn LB Σ_{lb} of Σ such that $\mathcal{M}(\Sigma_{glb}) \subset \mathcal{M}(\Sigma_{lb}) \subseteq \mathcal{M}(\Sigma)$.

Example 1. Consider the formula $\Sigma = (\mathit{master\_student} \vee \mathit{phd\_student}) \wedge (\mathit{master\_student} \rightarrow \mathit{student}) \wedge (\mathit{phd\_student} \rightarrow \mathit{student})$ (cf. [23]). Then:
- $(\mathit{master\_student} \wedge \mathit{phd\_student} \wedge \mathit{student})$ is a Horn LB of Σ;
- $(\mathit{master\_student} \wedge \mathit{student})$ is a Horn GLB of Σ.

In this section Σ denotes a propositional formula in CNF and Σ_{glb} denotes one of its Horn GLBs.

In [23,24] a *Horn strengthening* of a clause γ is a Horn clause γ_S such that $\gamma_S \subseteq \gamma$ and there is no Horn clause γ'_S such that $\gamma_S \subset \gamma'_S \subseteq \gamma$ (here a clause is considered as a set of literals). Note that a Horn strengthening of a clause γ is either $B(\gamma) \rightarrow h$, where $h \in H(\gamma)$, or $\neg B(\gamma)$, if $H(\gamma) = \emptyset$.

As noticed by Selman and Kautz [23,24], we can always find a Horn GLB of a formula by choosing its clauses among its Horn-strengthenings.

Proposition 1 (Selman and Kautz [24]). *Let Σ_{glb} be a Horn GLB of a CNF formula $\Sigma = C_1 \wedge \cdots \wedge C_n$. Then, there exists a formula $\Sigma' = C'_1 \wedge \cdots \wedge C'_n$ where each C'_i is a Horn strengthening of C_i and such that $\Sigma_{glb} \equiv \Sigma'$.*

If a Horn GLB Σ_{glb} of Σ is composed only of Horn strengthenings of clauses in Σ (as in the proposition above), we say that Σ_{glb} is in *normal form*. By Proposition 1, it thus follows that, for any Horn GLB Σ_{glb} for Σ, there exists a Horn GLB in normal form Σ' for Σ such that $\Sigma_{glb} \equiv \Sigma'$. For this reason, henceforth we assume all GLBs are in such a form, unless explicitly stated otherwise.

Example 2. Consider again the formula $\Sigma = (\mathit{master\_student} \vee \mathit{phd\_student}) \wedge (\mathit{master\_student} \rightarrow \mathit{student}) \wedge (\mathit{phd\_student} \rightarrow \mathit{student})$ of Example 1. Then:
- $(\mathit{master\_student} \wedge \mathit{student})$ is a Horn GLB of Σ, but is not in normal form;
- $(\mathit{master\_student}) \wedge (\mathit{master\_student} \rightarrow \mathit{student}) \wedge (\mathit{phd\_student} \rightarrow \mathit{student})$ is a Horn GLB of Σ in normal form.

3. Horn GLBs and minimal models

In this section we prove that Horn GLBs of a formula Σ are closely related to the *minimal* models of Σ. Recall that Horn formulae have a unique minimal model (the *minimum* model). We will show that the minimum model of any Horn greatest lower bound Σ_{glb} is minimal for Σ. This has both semantical and computational importance. Minimal models are important in the theory of non-monotonic reasoning, since they are the semantical counterpart of circumscription [19,20]: the models of $CIRC(\Sigma)$ are exactly the minimal models of Σ.

Lemma 2. *Let Σ be a propositional formula and Σ_{glb} a Horn GLB of Σ. The minimum model of Σ_{glb} is minimal for Σ.*

Proof. First of all we notice that the minimum model M of Σ_{glb} is also a model of Σ. Now, let's assume that M is not minimal, and let N be a model of Σ such that $N \subset M$. We prove that we can build a Horn formula U such that $\mathcal{M}(\Sigma_{glb}) \subset \mathcal{M}(U) \subseteq \mathcal{M}(\Sigma)$, thus contradicting the assumption that Σ_{glb} is a Horn GLB of Σ.

The Horn formula U is built as follows:

begin
  unmark all the clauses of Σ;
  $U :=$ `true`;
  for each clause $\gamma = \neg b_1 \vee \cdots \vee \neg b_n \vee a_1 \vee \cdots \vee a_m$ of Σ **do**
    for $i := 1$ to m **do**
      if $a_i \in N$
      then begin
          (* add a Horn-strengthening of γ *)
          $U := U \wedge B(\gamma) \rightarrow a_i$;
          mark γ
        end;
  for each unmarked clause γ of Σ
  do begin
    γ':= an arbitrary Horn-strengthening of γ in Σ_{glb};
    $U := U \wedge \gamma'$;
   end;
end.

Since U is a collection of Horn-strengthenings of Σ, $\mathcal{M}(U) \subseteq \mathcal{M}(\Sigma)$ holds. Moreover, N is a model of U: Indeed,

(1) N clearly satisfies all the clauses in U that come from marked clauses of Σ; and
(2) N must satisfy at least one negative literal, i.e., $b_i \notin N$ for at least one i $(1 \leqslant 1 \leqslant n)$, of each clause γ' in U that comes from an unmarked clause γ of Σ, otherwise γ would have been marked.

Now we prove that $\mathcal{M}(\Sigma_{glb}) \subset \mathcal{M}(U)$ holds. Since $N \in \mathcal{M}(U)$ and $N \notin \mathcal{M}(\Sigma_{glb})$, it is sufficient to prove that $\mathcal{M}(\Sigma_{glb}) \subseteq \mathcal{M}(U)$. Let's take a generic model P of Σ_{glb}; we prove that it is also a model of U. Since P is a model of Σ_{glb}, $M \subset P$ must hold, hence $N \subset P$ holds too. As a consequence, P satisfies all the clauses in U that come from marked clauses of Σ. As far as the other clauses of U are concerned, they are clauses of Σ_{glb} as well, therefore P satisfies all of them. □

Theorem 3. *The set of minimal models of a propositional formula Σ and the set of minimum models of the Horn GLBs of Σ are the same.*

Proof. From Lemma 2, it follows that, for any Horn GLB Σ_{glb} of Σ, the minimum model of Σ_{glb} is a minimal model of Σ.

We next prove that any minimal model of Σ is the minimum model of some GLB of Σ. Let M be a minimal model of Σ. Define $\Sigma' = \{l \mid l \in M\} \cup \{\neg l \mid l \notin M\}$. This set of clauses is a Horn formula and $\{M\} = \mathcal{M}(\Sigma')$ holds. Hence, $\mathcal{M}(\Sigma') \subseteq \mathcal{M}(\Sigma)$, and Σ' is a Horn LB of Σ. Therefore, there exists a Horn GLB Σ_{glb} of Σ such that $\Sigma' \models \Sigma_{glb} \models \Sigma$ and thus $M \in \mathcal{M}(\Sigma_{glb})$. Assume by contradiction that the minimum model of Σ_{glb}, say M', is not equal to M. Then, $M' \subset M$ holds, because M is a model of Σ_{glb}. Since every model of Σ_{glb} is a model of Σ, $M' \in \mathcal{M}(\Sigma)$. However, this is a contradiction, because we assumed M is a minimal model of Σ. □

We now address some interesting semantical consequences of the above results.

As noticed in [23] a traditional AI approach is *reasoning by counterexamples*, which consists in refuting a possible consequence of a formula by means of a suitable model that contradicts it (an example of this technique is in the early work [11]). This approach is based on the well-known property $M \not\models \alpha \Rightarrow \Sigma \not\models \alpha$, that holds for any pair of formulae α, Σ and any model M of Σ. Selman and Kautz indicate that reasoning under a specific Horn GLB is an improved version of such a reasoning schema, since a single Horn GLB captures a *set* of models of the original formula. They also briefly address the issue of what reasoning with respect to a set of Horn GLBs looks like, proving [23, Theorem 3] that a formula is equivalent to the disjunction of all its Horn GLBs.

We move further in this direction exploring some properties of reasoning with Horn GLBs. In particular, we consider the two relevant notions of *skeptical* and *brave* reasoning, which are frequently used in the AI literature: Let Σ and α be two formulae.

- α *skeptically follows* from the Horn GLBs of Σ, denoted by *skep-glb*$(\Sigma) \vdash \alpha$, if for each Horn GLB Σ_{glb} of Σ it holds that $\Sigma_{glb} \models \alpha$;
- α *bravely follows* from the Horn GLBs of Σ, denoted by *brave-glb*$(\Sigma) \vdash \alpha$, if there exists a Horn GLB Σ_{glb} of Σ such that $\Sigma_{glb} \models \alpha$ holds.

From the above mentioned result by Selman and Kautz, Theorem 3, and results of [19] relating minimal models and circumscription, the following result follows.

Corollary 4. *Let Σ be a formula.*

(i) *For any formula α, skep-glb$(\Sigma) \vdash \alpha$ iff $\Sigma \models \alpha$.*

(ii) *For any positive clause γ, brave-glb$(\Sigma) \vdash \gamma$ iff there exists a minimal model M of Σ such that $M \models \gamma$, i.e., iff CIRC$(\Sigma) \not\models \neg\gamma$.*

Thus, a formula α skeptically follows from the Horn GLBs of Σ if and only if it follows from Σ. Moreover—as far as positive clauses are concerned—brave reasoning with respect to Horn GLBs is the same as brave reasoning with respect to minimal models. Equivalently, since the minimum model of a Horn formula completely characterizes the set of its positive consequences, we can also say that reasoning under Horn GLBs is the same as reasoning by counterexamples using only minimal models. This does not hold for negative theorems.

By exploiting the relationships with classical and circumscriptive reasoning, we get the computational complexity of reasoning with Horn GLBs.

Proposition 5. *Let Σ and α be two formulae. Then,*
(i) *deciding whether skep-glb*$(\Sigma) \vdash \alpha$ *is* co-NP-*complete*;
(ii) *deciding whether brave-glb*$(\Sigma) \vdash \alpha$ *is* Σ_2^{p}*-complete.*

Proof. Point (i) trivially holds, because *skep-glb*$(\Sigma) \vdash \alpha$ iff $\Sigma \models \alpha$, and the classical inference problem is co-NP-complete.

(ii) From the Σ_2^{p}-hardness of brave reasoning under circumscription [9] and Corollary 4(ii), it follows that brave reasoning with respect to Horn GLBs is Σ_2^{p}-hard.

We next show that deciding whether *brave-glb*$(\Sigma) \vdash \alpha$ is in Σ_2^{p}. Consider the following "guess-and-check" algorithm:
(a) guess a formula Σ' such that $size(\Sigma') \leqslant size(\Sigma)$;
(b) verify that Σ' is a Horn GLB of Σ and that $\Sigma' \models \alpha$.
This algorithm correctly decides whether *brave-glb*$(\Sigma) \vdash \alpha$. Indeed, the size limitation in step (a) is sound because we can consider just Horn GLBs in normal form, whose size is bounded by the size of Σ. Moreover, the algorithm can be implemented on a nondeterministic Turing machine with an oracle in co-NP, because checking whether Σ' is a Horn GLB of Σ is in co-NP, and checking whether $\Sigma' \models \alpha$ is polynomial, because Σ' is Horn if the first check is successful. □

Let us see how the relation with non-monotonicity just shown affects approximate inference under Horn GLBs.

We recall that reasoning using a generic Horn GLB is complete and unsound with respect to reasoning using the original formula. Let Σ and α be two formulae, and assume that *brave-glb*$(\Sigma) \nvdash \alpha$. Then, for each Horn GLB Σ' of Σ, $\Sigma' \not\models \alpha$ holds. This means that whatever Horn GLB we compute, we can disprove α. Now, assume we get "more knowledge", in form of a set of clauses $C_1, \ldots, C_n$ to be added to Σ. Let $\Sigma^+ = \Sigma \wedge C_1 \wedge \cdots \wedge C_n$ be the resulting "bigger" knowledge base. Clearly, $\Sigma^+ \models \Sigma$. Suppose that α is not a consequence of Σ^+, i.e., $\Sigma^+ \not\models \alpha$ holds for Σ^+, too. A desirable property of Horn GLBs would be to preserve the possibility of disproving α with any Horn GLB of Σ^+. The following example shows that

$$\textit{brave-glb}(\Sigma) \nvdash \alpha \text{ and } \Sigma^+ \not\models \alpha \quad \not\Rightarrow \quad \textit{brave-glb}(\Sigma^+) \nvdash \alpha.$$

Thus, the above property does not hold for Horn GLBs. This means that, in general, compiling more knowledge does not always give "better" complete approximations.

Example 3. Consider the formulae $\Sigma = \neg a \vee \neg b$ and $\alpha = a$. Moreover, let $\Sigma^+ = \Sigma \wedge (a \vee b)$ be a new, "bigger", knowledge base. Clearly, $\Sigma^+ \not\models a$ and *brave-glb*$(\Sigma) \nvdash a$. However, *brave-glb*$(\Sigma^+) \vdash a$. Indeed, Σ^+ has two different Horn GLBs: $(a \wedge \neg b)$ and

$(\neg a \wedge b)$. The former, $(a \wedge \neg b)$, entails a and thus is not able to disprove this atomic formula.

Note that this observation reflects the relationships between brave reasoning with Horn GLBs and circumscription. In fact it is well known that, since circumscription is a non-monotonic formalism, for a generic formula α, $\mathit{CIRC}(\Sigma) \models \neg\alpha$ does not imply $\mathit{CIRC}(\Sigma^+) \models \neg\alpha$, even if $\Sigma^+ \not\models \alpha$ holds.

For the sake of completeness, we notice that $\textit{brave-glb}(\Sigma) \vdash \alpha$ does not imply $\textit{brave-glb}(\Sigma^+) \vdash \alpha$: the implication does not hold when, e.g., $\Sigma = a \vee b$, $\Sigma^+ = \Sigma \wedge b$, and $\alpha = a$.

4. The complexity of finding Horn GLBs

In this section we analyze the computational complexity of finding a Horn GLB of a CNF formula. We find a lower bound and an upper bound of this search problem, and we give a precise characterization of its complexity in terms of polynomial-time Turing reductions, by showing the problem to be NP-equivalent.

Lemma 2 implies that if we have a Horn GLB Σ_{glb} of Σ, then we can obtain in time linear in the size of Σ_{glb} a minimal model of Σ: just compute the minimum model of Σ_{glb} using the well known algorithm of Dowling and Gallier (see [8]). More technically, the theorem shows a polynomial-time (Turing) reduction from the search problem of finding a minimal model of Σ to the search problem of finding a Horn GLB of Σ. The computational complexity of the search problem of finding a minimal model of a propositional formula has been analyzed in [3,6]. One of the results in these papers is that finding a minimal model of a formula Σ is hard (using many-one reductions) with respect to the class $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$.

As mentioned in Section 2.1, $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$-hard problems are in a precise sense computationally harder both than NP-complete problems and co-NP-complete problems. We recall that the problem of deciding whether $\Sigma \models \alpha$ holds, i.e., the original problem we want to solve, is co-NP-complete.

As shown in [3], $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$-hardness of finding a minimal model holds even if a model of Σ is known. This fact can be compared with a consideration in [23, Theorem 1]: Σ_{glb} is satisfiable iff Σ is satisfiable, hence finding a Horn GLB is NP-hard. By Lemma 2, it follows that even if we know that Σ is satisfiable and have one of its models in hand, finding a Horn GLB is still $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$-hard. We recall that finding a model (not necessarily minimal) of a propositional formula is *per se* an NP-hard task.

Corollary 6. *Finding a Horn GLB of a propositional formula Σ is $\mathrm{P}^{\mathrm{NP}[O(\log n)]}$-hard. This holds even if a model of Σ is already known.*

Corollary 6 gives a lower bound that holds even for the case when a model of Σ is known. One may wonder whether the problem becomes easier if we have more information, e.g., a *minimal* model of Σ (in the sequel the importance of having minimal models in order to find GLBs is highlighted). The following theorem shows that this is not the case.

Theorem 7. *Finding a Horn GLB of a propositional formula Σ is* $\mathrm{P}^{\mathrm{NP}[\mathrm{O}(\log n)]}$*-hard. This holds even if*

(1) *a minimal model of Σ is already known*; *or*

(2) *a Horn LB of Σ whose minimum model is a minimal model of Σ is already known.*

Proof. (1) We reduce—by means of a polynomial-time transformation—the problem of finding a Horn GLB of a propositional formula Σ to the problem of finding a Horn GLB of a propositional formula Σ_p, with a minimal model of Σ_p given. Σ_p is defined as $\{\gamma \vee \neg p \mid \gamma$ is a clause in $\Sigma\} \equiv (\Sigma \vee \neg p)$, where p is a new propositional variable, not occurring in Σ. The only minimal model for Σ_p is $\emptyset$.

We prove that, given any Horn GLB Σ_{glb} of Σ_p, we can determine in linear time (in the size of Σ_{glb}) a Horn GLB of Σ. Here, we do not make any assumption about the syntactic form of Σ_{glb}. In particular, we do not require Σ_{glb} to contain only Horn-strengthenings of Σ. Next, we show that $\Sigma_g = \{(B - \{p\} \rightarrow h) \mid B \rightarrow h$ is a clause in Σ_{glb}, and $h \neq p\}$ is a Horn GLB of Σ. Note that p does not occur in Σ_g. Now, we need three useful properties of Σ_g.

Fact a. $\mathcal{M}(\Sigma_g) = \{M - \{p\} \mid p \in M$ *and* $M \in \mathcal{M}(\Sigma_{glb})\}$.

Let $\mathcal{M}_p$ denote the set of models for Σ_{glb} which contain the atom p, i.e., $\mathcal{M}_p = \{M \in \mathcal{M}(\Sigma_{glb}) \mid p \in M\}$. Now, consider the formula $\Sigma_{glb} \wedge p$. It can be verified that $\mathcal{M}(\Sigma_{glb} \wedge p) = \mathcal{M}_p$. Furthermore, we have $\Sigma_{glb} \wedge p \equiv \Sigma_g \wedge p$. Indeed, clauses of Σ_{glb} having p in their head are subsumed by the clause p; clauses of Σ_{glb} in which p occurs in a negative literal, i.e., of the form $\gamma \vee \neg p$ can be clearly resolved with the clause p to get γ, which belongs to Σ_g, by definition. Then, $\mathcal{M}(\Sigma_g \wedge p) = \mathcal{M}_p$. Since p does not appear in the formula Σ_g, this entails $\mathcal{M}(\Sigma_g) = \{M - \{p\} \mid M \in \mathcal{M}_p\}$.

Fact b. $\Sigma_g \models \Sigma$.

Assume there exists a model $M \in \mathcal{M}(\Sigma_g)$ such that $M \notin \mathcal{M}(\Sigma)$. Then, from Fact a, $M \cup \{p\}$ belongs to $\mathcal{M}(\Sigma_{glb})$, but $M \cup \{p\} \notin \mathcal{M}(\Sigma \vee \neg p)$. This contradicts the hypothesis that $\Sigma_{glb} \models \Sigma_p$.

Fact c. $\Sigma_{glb} \models (\Sigma_g \vee \neg p)$.

Models of Σ_{glb} which map p into 0 are models of $\neg p$. Models of Σ_{glb} which map p into 1 are models of Σ_g (cf. Fact a, plus the fact that p does not occur in Σ_g).

Now, observe that Fact b implies that Σ_g is a Horn LB of Σ. To conclude, assume Σ_g is not a Horn GLB of Σ, i.e., there exists some Horn formula Σ' such that $\Sigma_g \models \Sigma' \models \Sigma$ and $\Sigma' \not\models \Sigma_g$. Then, since p occurs neither in Σ' nor in Σ_g, $(\Sigma' \vee \neg p) \not\models (\Sigma_g \vee \neg p)$ holds, and, as a consequence of Fact c, $(\Sigma' \vee \neg p) \not\models \Sigma_{glb}$. Furthermore, by applying Fact c, we get $\Sigma_{glb} \models (\Sigma_g \vee \neg p) \models (\Sigma' \vee \neg p) \models (\Sigma \vee \neg p)$. Since Σ' is Horn, the formula obtained by adding the atom p to the body of each clause belonging to Σ' (i.e., the formula $\{B \wedge p \to h \mid B \to h \in \Sigma'\}$), which is equivalent to $\Sigma' \vee \neg p$, is a Horn formula too. Thus, we contradict the hypothesis that Σ_{glb} is a Horn GLB for Σ_p.

(2) Note that $\neg p$ is a Horn LB of Σ_p, and that its minimum model is a minimal model of Σ_p. Therefore the problem of finding a Horn GLB of Σ reduces to the problem of finding a Horn GLB of Σ_p given its Horn LB $\neg p$. □

Theorem 7 shows that even having a "good approximation" of a GLB, i.e., a minimal model of Σ, or even a LB that has as its minimum model a minimal model of the original formula, does not make the problem of finding a GLB any easier.

We notice that Corollary 6 and Theorem 7 give us just a lower bound. It is reasonable to ask *how easy* it is to find a Horn GLB, i.e., to give an upper bound to the complexity of the problem. In [23] an algorithm for computing a Horn GLB of a formula Σ is shown. The algorithm performs an exponential number of polynomial steps.

We next show that a Horn GLB can be found in polynomial time by a deterministic Turing machine with access to an NP oracle, i.e., we prove that the problem is in the class P^{NP}. This means that we only need a polynomial number of queries to the GLB in order to "pay off" the overhead of the knowledge compilation. In particular, we prove that it is possible to build a Horn GLB of Σ by using a linear number of times a subroutine that returns a minimal model of an arbitrary formula. This result allows us to obtain a precise upper bound on the complexity of finding a Horn GLB.

We propose an algorithm that is based on the idea of transforming each clause γ of Σ into one of its Horn-strengthenings. In this case, each non-Horn rule is treated separately, using a different minimal model of a formula related to Σ. The transformation uses a function which is defined here. Given a formula Σ and a set of atoms M, we denote by $Str(\Sigma, M)$ the following set of clauses

$$\{B(\gamma) \to (H(\gamma) \cap M) \mid \gamma \in \Sigma \text{ and } B(\gamma) \subseteq M\}.$$

$Str(\Sigma, M)$ returns strengthenings (not necessarily Horn) of some of the clauses in Σ. As an example, if

$$\Sigma = \{(a \to b \vee c \vee d), (b \wedge c \to a \vee d), (d \to a)\}$$

and $M = \{a, b, c\}$, then

$$Str(\Sigma, M) = \{(a \to b \vee c), (b \wedge c \to a)\}.$$

The following lemma shows that, if M is appropriately chosen, we can transform Σ by replacing some of its clauses with the corresponding ones in $Str(\Sigma, M)$, thus being sure that we do not miss all Horn GLBs of Σ.

Lemma 8. *Let Σ be a formula, γ a non-Horn clause of Σ, M a minimal model of $\Sigma \wedge B(\gamma)$, and h an atom in $H(\gamma) \cap M$. Then, each Horn GLB of $\Sigma \wedge Str(\Sigma, M) \wedge (B(\gamma) \to h)$ is a Horn GLB of Σ.*

Proof. Let Σ' be the formula obtained from $\Sigma \wedge Str(\Sigma, M) \wedge (B(\gamma) \to h)$ by removing subsumed clauses, and Π a Horn GLB of Σ' in normal form. We remind that each clause of Π is a Horn-strengthening of some clause in Σ'.

First of all, note that Π is a Horn LB of Σ. Assume now Π is not a Horn GLB of Σ. Then, there exists a Horn GLB of Σ, say Π', such that $\Pi \models \Pi' \models \Sigma$ and $\Pi' \not\models \Pi$. From hypothesis and assumptions above, the following facts follow.

Fact a. $M \in \mathcal{M}(\Pi \wedge B(\gamma))$.

By definition, $M \in \mathcal{M}(B(\gamma))$. Assume $M \notin \mathcal{M}(\Pi)$. Then, there exists a Horn-strengthening of some clause $\alpha \in \Sigma'$, say α', such that $\alpha' \in \Pi$, and α' is violated by M, i.e., $B(\alpha') \subseteq M$ and $H(\alpha') \cap M = \emptyset$. Since $B(\alpha) = B(\alpha')$ is a subset of M, and no subsumed clause occurs in Σ', then $\alpha \in Str(\Sigma, M)$, by construction of $Str(\Sigma, M)$. Moreover, M is a model for Σ, then $H(\alpha) \neq \emptyset$. Now, recall that α' is a Horn-strengthening of α. Therefore, we have $H(\alpha') \neq \emptyset$ and $H(\alpha') \subseteq H(\alpha) \subseteq M$, which contradicts the hypothesis that α' is violated by M.

Fact b. *M is the minimum model of the Horn formula $\Pi' \wedge B(\gamma)$.*

Consider the conjunction of Π, Π', and Σ with the same formula $B(\gamma)$. Then the following relation holds: $(\Pi \wedge B(\gamma)) \models (\Pi' \wedge B(\gamma)) \models (\Sigma \wedge B(\gamma))$. Fact a and the above relation entail that M is a model of $\Pi' \wedge B(\gamma)$. Moreover, since $(\Pi' \wedge B(\gamma)) \models (\Sigma \wedge B(\gamma))$, M is the minimal model for $\Pi' \wedge B(\gamma)$, otherwise the minimality of M for $\Sigma \wedge B(\gamma)$ would be contradicted.

Fact c. $\Pi' \models B(\gamma) \rightarrow h$.

From Fact b we know that $\Pi' \wedge B(\gamma) \models \bigwedge_{m \in M} m$. Hence, we get $\Pi' \models B(\gamma) \rightarrow \bigwedge_{m \in M} m$ and, in particular, $\Pi' \models B(\gamma) \rightarrow h$.

Fact d. *Π' is a Horn LB of Σ'.*

We proceed by contradiction. If $\Pi' \not\models \Sigma'$, then there exists a model M' for Π' which is not a model for Σ'. Since $\Pi' \models \Sigma$ (by definition), and $\Pi' \models B(\gamma) \rightarrow h$ (cf. Fact c), the violated clauses must belong to the set of clauses $Str(\Sigma, M)$. Let $\gamma' = B' \rightarrow H'$ be one of such violated clauses. By definition of $Str(\Sigma, M)$, it holds that $B' \subseteq M$; furthermore, γ' is violated by M', hence $H' \cap M' = \emptyset$ and $B' \subseteq M'$ hold. Note that, since $\gamma' \in Str(\Sigma, M)$, γ' corresponds to some clause $(B' \rightarrow \bar{H}) \in \Sigma$ such that $H' = \bar{H} \cap M$, i.e., H' contains all the atoms in the head $\bar{H}$ belonging to M. Now, since Π' is in normal form, it contains at least one Horn-strengthening of the clause $(B' \rightarrow \bar{H}) \in \Sigma$. Let $B' \rightarrow h'$ $(h' \in \bar{H})$ be such a Horn-strengthening. We have $h' \in M'$, because M' is a model for Π' and $B' \subseteq M'$. On the other hand, since $B' \subseteq M$ and $\Pi' \wedge B(\gamma) \models \bigwedge_{m \in M} m$, we get $\Pi' \wedge B(\gamma) \models \bigwedge_{m \in M} m \wedge h'$. This entails $h' \in M$, because from Fact b we know M is the minimal model of $\Pi' \wedge B(\gamma)$. Then, $h' \in \bar{H} \cap M$ and, as a consequence, $h' \in H'$, i.e., h' belongs to the head of γ' and to M', contradicting the hypothesis that γ' is violated by M'.

To conclude, remember we assumed $\Pi \models \Pi'$ and $\Pi' \not\models \Pi$. From Fact d, we know $\Pi' \models \Sigma'$. Then, the Horn formula Π cannot be a Horn GLB for Σ', a contradiction. □

```
Algorithm MinimalModel2GLB
Input  a CNF formula Σ
Output  a Horn GLB of Σ
begin
    while Σ is not a Horn formula
    do begin
        γ := an arbitrary non-Horn clause of Σ;
        if  Σ ∧ B(γ) is satisfiable
        then begin
                M := an arbitrary minimal model of Σ ∧ B(γ);
                h := an arbitrary atom in H(γ) ∩ M;
                Σ := Σ ∧ Str(Σ, M) ∧ (B(γ) → h)
            end
        else
            Σ := Σ ∧ ¬B(γ);
        remove subsumed clauses from Σ
      end;
    return Σ
end.
```

Fig. 1. The algorithm *MinimalModel2GLB*.

Fig. 1 shows the algorithm *MinimalModel2GLB* that finds a Horn GLB of a formula. Σ is a variable initialized with the input (non-Horn) formula. At the end of *MinimalModel2GLB*, Σ will contain a Horn GLB of the input formula. At each step of the **while** loop, a non-Horn clause of Σ is selected and replaced by one of its Horn-strengthenings. We exploit the fact that each Horn-strengthening γ' of a clause γ contains all the atoms in the body, i.e., $B(\gamma) \subseteq \gamma'$. Thus, to get the Horn clause γ' we just determine its head atom (if any), performing a suitable choice among the atoms in $H(\gamma)$. To this end, we look at the formula $\Sigma \wedge B(\gamma)$. If this formula is unsatisfiable, no head atom can be "derived" from the selected clause, and we can simply replace γ by the Horn clause containing only its body, i.e., $\neg B(\gamma)$. Otherwise, we have to select an atom from the head of γ. The idea is to choose an atom h which also belongs to a minimal model M of $\Sigma \wedge B(\gamma)$. Intuitively, in the Horn GLB which covers M, if $B(\gamma)$ is true then h should be necessarily derived. Thus, choosing h gives a clause $\gamma' = (B(\gamma) \rightarrow h)$ which is less restrictive than any other Horn-strengthening of γ, and hence leads to a formula covering one of the largest sets of models of Σ.

M is also used to perform further simplifications of Σ. Indeed, the clauses belonging to $Str(\Sigma, M)$ will replace the clauses of Σ they subsume. This step is fundamental for the soundness of *MinimalModel2GLB*. It guarantees that M will continue to be a model of the (current) formula Σ—and of $\Sigma \wedge B(\gamma)$—also in the following iterations of the algorithm.

Eventually, the variable Σ will contain only Horn clauses and the algorithm ends returning Σ as a Horn GLB of the input formula.

Example 4. Consider the following formula

$$\Sigma = \{(a \rightarrow b \vee c), (a \vee b)\}.$$

We find a Horn GLB for Σ by applying the algorithm of Fig. 1. Σ contains two non-Horn clauses. We arbitrarily select the clause $\gamma = a \rightarrow b \vee c$. Now, we have to look for a minimal model of the formula $\Sigma \wedge a$. For instance, we could compute the model $M = \{a, b\}$. Since $H(\gamma) \cap \{a, b\} = \{b\}$, the only possible head for the Horn-strengthening of γ is the atom b. Moreover, $Str(\Sigma, M) = \{(a \rightarrow b), (a \vee b)\}$. By removing the subsumed clauses in $\Sigma \wedge (a \rightarrow b) \wedge Str(\Sigma, M)$, we get the new formula $\Sigma' = \{(a \rightarrow b), (a \vee b)\}$.

Now we have only one non-Horn clause, namely $\gamma' = a \vee b$. Since the body of γ' is true, we look for a minimal model of the formula Σ', without any additional clause. Σ' has just one minimal model, namely $\{b\}$, hence we get $\Sigma'' = \Sigma' \wedge b \equiv \{b\}$. $\{b\}$ is clearly a Horn formula, and it is also a Horn GLB of the input formula Σ.

Another Horn GLB of Σ, i.e., the formula $\{a, a \rightarrow c\}$, could be obtained by selecting the other minimal model of $\Sigma \wedge a$, i.e., $\{a, c\}$. Both GLBs could be also computed by initially choosing clause $a \vee b$ instead of $a \rightarrow b \vee c$.

Theorem 9. *Algorithm MinimalModel2GLB is correct.*

Proof. The variable Σ initially contains the input formula (let's name it Σ^0). *MinimalModel2GLB* modifies Σ until it becomes a Horn formula. We show by induction that, at each step, every Horn GLB of the formula Σ is a Horn GLB of Σ^0. This clearly holds when the algorithm starts, because $\Sigma = \Sigma^0$.

Assume this property holds at the beginning of some execution of the **while** loop. We select a non-Horn clause γ of Σ, and look for a minimal model of the formula $\Sigma \wedge B(\gamma)$, i.e., the formula Σ with the additional constraint that the "body" of clause γ is true.

If such a model does not exist (i.e., $\Sigma \wedge B(\gamma)$ is unsatisfiable), then $\Sigma \models \neg B(\gamma)$, and we can get a formula equivalent to Σ—and hence with the same set of Horn GLBs—by replacing the non-Horn clause γ by the (Horn) clause $\neg B(\gamma)$. Otherwise, let M be a minimal model for $\Sigma \wedge B(\gamma)$. We select an atom $h \in H(\gamma) \cap M$. Note that $H(\gamma) \cap M \neq \emptyset$, because $B(\gamma) \subseteq M$ and M is a model for Σ.

Let Σ' be the formula obtained by adding to Σ all the clauses in $Str(\Sigma, M)$ plus the additional clause $B(\gamma) \rightarrow h$, and by removing subsumed clauses. From Lemma 8, it follows that every Horn GLB of Σ' is a Horn GLB of Σ, and hence, by the induction hypothesis, is a Horn GLB of Σ^0, too. This concludes the induction proof. Note that Σ' will contain fewer non-Horn clauses than Σ, since at least γ will be deleted, because it is subsumed by $B(\gamma) \rightarrow h$.

Let n be the number of non-Horn clauses in Σ^0. After at most n iteration of the **while** loop, we get a Horn formula Σ whose unique Horn GLB (up to logical equivalence) is Σ itself. From the above induction property, it follows that Σ is a Horn GLB of the input formula Σ^0, and thus *MinimalModel2GLB* is correct. □

The method described by algorithm *MinimalModel2GLB* gives an upper bound on the complexity of the problem of finding a Horn GLB, as specified by the next theorem.

Theorem 10. *Finding a Horn GLB of a propositional formula Σ in CNF is in* P^{NP}.

Proof. Let $|\Sigma|$ and $\|\Sigma\|$ denote the number of clauses and the number of propositional variables in Σ, respectively. Referring to the algorithm in Fig. 1, the **while** cycle can be executed at most $|\Sigma|$ times, because at the end of each iteration, at least the selected non-Horn clause γ will be subsumed and removed from Σ. All the performed operations are polynomial-time computable, except the problem of finding a minimal model for $\Sigma \wedge B(\gamma)$ (which subsumes the satisfiability test for $\Sigma \wedge B(\gamma)$).

A minimal model of a formula Σ can be determined with $O(\|\Sigma\|)$ calls to an NP oracle, cf. [1]. Then, the whole execution of the algorithm requires at most $O(\|\Sigma\| \cdot |\Sigma|)$ calls to an NP oracle, i.e., a number of calls polynomial in the size of the input formula Σ. □

Theorem 10 and Corollary 6 immediately give a complete characterization of the computational complexity of this problem in terms of NP-equivalence, as described in Section 2.

Corollary 11. *Finding a Horn GLB of a propositional formula Σ in CNF is* NP-*equivalent.*

5. Conclusions

Research on GLBs has so far focused on algorithms for computing them and on empirical evaluation of the quality of the approximation. In this paper we have addressed some formal issues about GLBs. In particular, we have investigated their semantics and the intrinsic complexity of finding them.

From the semantical point of view, we have discovered an interesting relation between GLBs and a popular form of non-monotonic reasoning, i.e., circumscription. This relation essentially tells that reasoning with respect to Horn GLBs is the same as reasoning by counterexamples using only minimal models, and also explains why compiling more knowledge does not always give better approximations. Moreover, the relation gave us the basis for the subsequent computational analysis.

From the computational point of view, we showed that the problem of finding a GLB is not intrinsically exponential, and gave an upper and a lower bound to its complexity. The upper bound is reasonably close to the lower bound: finding a Horn GLB requires to solve a number of propositional satisfiability problems which is at least logarithmic and at most polynomial in the size of the input. The upper bound was obtained by means of an algorithm which uses as oracle a procedure for finding a minimal model. Empirical evaluation of the performance of the algorithm deserves future research.

Acknowledgements

Anonymous referees' comments and suggestions led to substantial improvements in the quality of this paper. We are grateful to H. Kautz and B. Selman for fruitful discussions on their method. We also thank L.C. Aiello, M. Lenzerini, E. Omodeo and M. Schaerf for reading previous versions of the paper. We also had interesting discussions with

L. Palopoli. This work has been supported by ASI (Italian Space Agency), MURST (Italian Ministry for University and Scientific and Technological Research) and CNR (Italian Research Council).

References

[1] R. Ben-Eliyahu, R. Dechter, On computing minimal models, Ann. Math. Artificial Intelligence (1996) 3–27.
[2] Y. Boufkhad, Algorithms for propositional KB approximation, in: Proc. AAAI-98, Madison, WI, 1998, pp. 280–285.
[3] M. Cadoli, On the complexity of model finding for nonmonotonic propositional logics, in: A. Marchetti Spaccamela, P. Mentrasti, M. Venturini Zilli (Eds.), Proc. 4th Italian Conference on Theoretical Computer Science, World Scientific, Singapore, October 1992, pp. 125–139.
[4] M. Cadoli, Semantical and computational aspects of Horn approximations, in: Proc. IJCAI-93, Chambéry, France, 1993, pp. 39–44.
[5] M. Cadoli, F.M. Donini, A survey on knowledge compilation, AI Communications—The European Journal on Artificial Intelligence 10 (1997) 137–150.
[6] Z. Chen, S. Toda, The complexity of selecting maximal solutions, Inform. and Comput. 119 (1995) 231–239.
[7] A. del Val, An analysis of approximate knowledge compilation, in: Proc. IJCAI-95, Montreal, Quebec, 1995, pp. 830–836.
[8] W.P. Dowling, J.H. Gallier, Linear-time algorithms for testing the satisfiability of propositional Horn formulae, J. Logic Programming 1 (1984) 267–284.
[9] T. Eiter, G. Gottlob, Propositional circumscription and extended closed world reasoning are Π_2^p-complete, Theoret. Comput. Sci. 114 (1993) 231–245.
[10] T. Eiter, T. Ibaraki, K. Makino, Disjunctions of Horn theories and their cores, in: Proc. 9th Annual Symposium on Algorithms and Computation (ISAAC-98), 1998, pp. 49–58.
[11] H. Gelernter, Realization of a geometry theorem-proving machine, in: Proc. International Conference on Information Processing, UNESCO House, Paris, 1959, pp. 273–282.
[12] G. Gogic, C. Papadimitriou, M. Sideri, Incremental recompilation of knowledge, J. Artificial Intelligence Res. 8 (1998) 23–37.
[13] R. Greiner, D. Schuurmans, Learning useful Horn approximations, in: Proc. Third International Conference on the Principles of Knowledge Representation and Reasoning (KR-92), Cambridge, MA, 1992, pp. 383–392.
[14] H.A. Kautz, M.J. Kearns, B. Selman, Horn approximations of empirical data, Artificial Intelligence 74 (1995) 129–145.
[15] H.A. Kautz, B. Selman, Forming concepts for fast inference, in: Proc. AAAI-92, San Jose, CA, 1992, pp. 786–793.
[16] H.A. Kautz, B. Selman, An empirical evaluation of knowledge compilation by theory approximation, in: Proc. AAAI-94, Seattle, WA, 1994, pp. 155–161.
[17] D. Kavvadias, C.H. Papadimitriou, M. Sideri, On Horn envelopes and hypergraph transversals, in: Proc. 4th Annual Symposium on Algorithms and Computation (ISAAC-93), 1993, pp. 399–405.
[18] R. Khardon, D. Roth, Reasoning with models, Artificial Intelligence 87 (1996) 187–213.
[19] V. Lifschitz, Computing circumscription, in: Proc. IJCAI-85, Los Angeles, CA, 1985, pp. 121–127.
[20] J. McCarthy, Circumscription—A form of non-monotonic reasoning, Artificial Intelligence 13 (1980) 27–39.
[21] C.H. Papadimitriou, Computational Complexity, Addison Wesley, Reading, MA, 1994.
[22] D. Roth, On the hardness of approximate reasoning, in: Proc. IJCAI-93, Chambéry, France, 1993, pp. 613–618.
[23] B. Selman, H.A. Kautz, Knowledge compilation using Horn approximations, in: Proc. AAAI-91, Anaheim, CA, 1991, pp. 904–909.
[24] B. Selman, H.A. Kautz, Knowledge compilation and theory approximation, J. ACM 43 (1996) 193–224.