On Enumerating All the Minimal Models for Particular CNF Formula Classes

Yakoub Salhi
CRIL-CNRS, Université d'Artois, F-62307 Lens Cedex, France

Keywords: Minimal Models, SAT, Enumeration Problem, Computational Complexity.

Abstract: In this work, we propose approaches for enumerating all the minimal models for two particular classes of CNF formulæ. The first class is that of PN formulæ which are defined as CNF formulæ where each clause is either positive or negative, whereas the second class is that of PH formulæ in which each clause is either positive or a Horn clause. We first provide an approach for enumerating all the minimal models in the case of PN formulæ that is based on the use of an algorithm for generating the minimal transversals of a hypergraph. We also propose a SAT-based encoding for solving the same problem. Then, we provide a characterization of the minimal models in the case of PH formulæ, which allows us to use our approaches in the case of PN formulæ for solving the problem of minimal model enumeration for PH formulæ. Finally, we describe an application in datamining of the problem of enumerating the minimal models in the case of PN formulæ.

1 INTRODUCTION

The problem of minimal model enumeration is based on preferring models that assign false to the propositional variables (see, e.g., (Ben-Eliyahu and Dechter, 1996; Niemelä, 1996)). This problem has important applications in AI, such as propositional circumscription (McCarthy, 1980), minimal diagnosis (Reiter, 1987) and logic programming (Bidoit and Froidevaux, 1987; Gelfond and Lifschitz, 1988).

It is noteworthy that the problem of checking whether a model is minimal in the general case is coNP-complete (Cadoli, 1992b). Different algorithms have been proposed in the literature for computing minimal models. The algorithms that have been proposed in (Ben-Eliyahu and Dechter, 1996) exploit tractable classes. For instance, knowing that any Horn formula admits a single minimal model, which can be found in linear time using the procedure of unit propagation, the main idea consists in instantiating as few variables as possible so that the remaining formula is a Horn formula. In addition, an incremental algorithm for enumerating all the minimal models has been proposed in (Ben-Eliyahu-Zohary, 2005). Other algorithms and approaches for the problem of enumerating minimal models and some related problems are proposed in (Reiter, 1987; Avin and Ben-Eliyahu-Zohary, 2001; Angiulli et al., 2014; Ben-Eliyahu-Zohary et al., 2017).

In this work, we mainly propose approaches to enumerate all the minimal models for two particular classes of CNF formulæ. The first class corresponds to CNF formulæ where each clause is either positive or negative, called PN formulæ. The second class is that of CNF formulæ where each clause is either positive or a Horn clause, called PH formulæ. Clearly, the class of PH formulæ is a generalization of that of PN formulæ. From the computational complexity point of view, we show that the problem of checking whether a model is minimal is tractable in the case of PN formulæ, while it is coNP-complete in the case of PH formulæ. To the best of our knowledge, the formula classes of PN formulæ and PH formulæ have not been studied in the case of the problem of minimal model enumeration.

We provide two methods to enumerate the minimal models in the case of PN formulæ. The first one consists in using the problem of enumerating the minimal transversals of a hypergraph to obtain all the minimal models that are contained in a model. Note that we provide an interesting property that allows us to reduce the search space in solving the latter problem. Indeed, we provide in this property the part shared between all the minimal models that are contained in a given model. Our second method consists in a SAT-based encoding. The main idea consists in using a formula expressing the fact that the truth value *true* is assigned to a variable if and only if there is a

Salhi, Y.
On Enumerating All the Minimal Models for Particular CNF Formula Classes.
DOI: 10.5220/0007257104030410
In *Proceedings of the 11th International Conference on Agents and Artificial Intelligence (ICAART 2019)*, pages 403-410
ISBN: 978-989-758-350-6

positive clause which is satisfied only by this assignment. Since checking whether a Boolean interpretation is a model of a CNF formula or not is a linear time task, our encoding can be seen as a proof that the problem of checking whether a model is minimal or not is tractable in the case of PN formulæ.

Our approach in the case of PH formulæ is based on enumerating the minimal models of its greatest PN subformula (the subset of all the positive and negative clauses). In fact, we provide a simple characterization of the minimal models in the case of PH formulæ that uses the procedure of unit propagation and the minimal models of the greatest PN subformulæ. Using this characterization, we propose an algorithm that uses our SAT-based encoding in the case of PN formulæ.

2 PRELIMINARIES

2.1 Propositional Satisfiability and Minimal Models

A CNF formula is a conjunction of clauses where a *clause* is a disjunction of literals. A *literal* is a propositional variable (positive literal) or its negation (negative literal). A *positive clause* (resp. *negative clause*) is a clause that contains only positive (resp. negative) literals. In addition, a *Horn clause* is a clause that contains at most one positive literal. We denote by $Var(\phi)$ (resp. $Lit(\phi)$) the set of propositional variables (resp. literals) occurring in the formula ϕ. A CNF formula is *binary* if each clause contains at most two literals. Propositional variables are denoted by the lowercase letters p, q, r, and CNF formulæ by the Greek letters ϕ, ψ, χ.

A CNF formula can also be seen as a set of clauses and a clause as a set of literals. Thus, a CNF formula of the form $(l_1^1 \vee \cdots \vee l_{k_1}^1) \wedge \cdots \wedge (l_1^n \vee \cdots \vee l_{k_n}^n)$ can be also represented in this work by the set of clauses $\{(l_1^1 \vee \cdots \vee l_{k_1}^1), \ldots, (l_1^n \vee \cdots \vee l_{k_n}^n)\}$ and the set of sets of literals $\{\{l_1^1, \ldots, l_{k_1}^1\}, \ldots, \{l_1^n, \ldots, l_{k_n}^n\}\}$.

A *Boolean interpretation* of a formula ϕ is an assignment that associates truth values in $\{0,1\}$ to the propositional variables in $Var(\phi)$, where 0 stands for false and 1 stands for true. It is extended to formulæ as usual. A *model* of a formula is a Boolean interpretation satisfying this formula, i.e., an interpretation making this formula true. We use $Models(\phi)$ to denote the set of models of ϕ. The problem of determining if there exists a model that satisfies a given propositional formula ($|Models(\phi)| \geq 1$), abbreviated as SAT, is one of the most studied NP-complete problems.

Given a set of literals L, we use $\phi_{|L}$ to denote the formula obtained from ϕ by removing every clauses containing a literal in L and by removing all the complement literals of those in L. More precisely, $\phi_{|L} = \{c \in \phi \mid \forall l \in L, l \notin c \ \& \ \bar{l} \notin c\} \cup \{c \setminus \{\bar{l}\} \mid c \in \phi \ \& \ l \in L \ \& \ \bar{l} \in c\}$.

For convenience purposes, we represent the Boolean interpretations as sets of variables. More precisely, the set m represents the Boolean interpretation that associates 1 to the variables in m and 0 to the variables in $Var(\phi) \setminus m$. Further, we use $(m)^-$ to denote the set of negative literals that are true by m, i.e., $(m)^- = \{\neg p \mid p \in Var(\phi) \setminus m\}$.

Definition 2.1 (Minimal Model). *Let ϕ be a CNF formula and m a model of ϕ. Then, m is a minimal model of ϕ iff there is no model m' of ϕ such that m' is a proper subset of m ($m' \subset m$).*

The problem of checking whether a model of a CNF formula is minimal is coNP-complete (Cadoli, 1992a).

We use $MinModels(\phi)$ to denote the set of minimal models of ϕ. Moreover, given a model m of ϕ, we use $MinMS(\phi, m)$ to denote the set of minimal models that are contained in m, i.e., $MinMS(\phi, m) = MinModels(\phi) \cap \{m' \in Models(\phi) \mid m' \subseteq m\}$.

2.2 PN and PH formulæ

We here describe two classes of CNF formulæ, namely PN (for Positive and Negative) and PH (for Positive and Horn) formulæ. In this work, we define different methods for enumerating all the minimal models in the case of these classes.

Definition 2.2 (PN formula). *A PN formula ϕ is a CNF formula where each clause is either positive or negative.*

We use $Pos(\phi)$ (resp. $Neg(\phi)$) to denote the set of positive (resp. negative) clauses of ϕ.

One can easily show that each CNF formula can be linearly transformed into an equisatisfiable PN formula. Indeed, we only have to use fresh propositional variables for renaming negative literals in the mixed clauses. As a consequence, the problem of checking the satisfiability of a PN formula is also NP-complete. Note that this property has been pointed out in (Boumarafi et al., 2017).

Definition 2.3 (PH formula). *A PH formula ϕ is a CNF formula where each clause is either positive or Horn.*

We use $Pos(\phi)$ (resp. $Horn(\phi)$) to denote the set of positive (resp. Horn) clauses of ϕ.

2.3 Hypergraphs and Minimal Transversals

A hypergraph H is an ordered pair (V,E) where V is a non empty set of vertices and $E \subseteq \mathcal{P}(V) \setminus \emptyset$ is a set of *hyperedges*, where $\mathcal{P}(V)$ denotes the power set of V, i.e., its set of subsets. In other words, the hypergraph notion is a generalization of that of undirected graph. Further, a *transversal* of H is a set of vertices $s \subseteq V$ that intersects every hyperedge of H, i.e., $e \cap s \neq \emptyset$ for every $e \in E$. A transversal of H is *minimal* if it is not a proper superset of any other transversal of H, i.e., $s' \not\subset s$ for every transversal s' of H. Given a hypergraph H, we use $MT(H)$ to denote the set of all the minimal transversals of H.

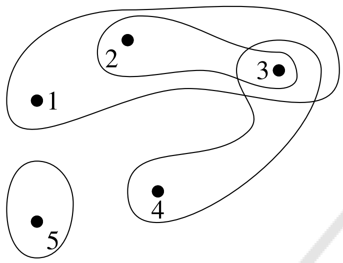

Figure 1: A Hypergraph.

Consider, for instance, the hypergraph described in Figure 1, the set $s = \{3,5\}$ is a minimal transversal since it intersects every hyperedge. The set $\{2,4,5\}$ is the only other minimal transversal.

In the literature, there are several methods for generating the minimal transversals of a hypergraph (see eg (Fredman and Khachiyan, 1996; Khachiyan et al., 2005; Khachiyan et al., 2007; Hébert et al., 2007)). In particular, the authors of (Hébert et al., 2007) have shown how efficient datamining techniques, such as the levelwise framework, can be used for generating the minimal transversals. In this work, we use an algorithm for generating the minimal transversals as a black box in our methods for generating the minimal models.

3 COMPUTATIONAL COMPLEXITY

In this section, we provide computational complexity results for the problem of checking whether a model is minimal in the case of PN formulæ and PH formulæ.

Theorem 3.1. *The problem of checking whether a model of a PN formula is minimal or not is in P.*

Proof. Using the SAT-based encoding defined in Section 5, we know that the problem of checking whether a model of a PN formula is minimal or not is in P. Indeed, it is well known that the problem of checking whether a Boolean interpretation is a model of a formulæ can be solved in linear time. □

Let $G = (V,E)$ be an undirected graph such that V is the set of its vertices and E is the set of its edges. A 3-coloring of G corresponds to a partition of V into 3 subsets $P = \{V_{c_1}, V_{c_2}, V_{c_3}\}$ so that no two vertices in a same subset are adjacent, and V_{c_1}, V_{c_2} and V_{c_3} refer to three different colors c_1, c_2 and c_3 respectively. It is well known that deciding whether an undirected graph admits a 3-coloring is an NP-complete problem. In the following theorem, we show that the complement of the 3-coloring problem can be transformed into the problem of checking whether a model of PH formula is minimal.

Theorem 3.2. *The problem of checking whether a model of PH formula is minimal is coNP-complete.*

Proof. Using the fact that the same problem in the general case is coNP-complete, we know that it is also in coNP in the case of PH formulæ. In order to show the coNP-completeness, we reduce the complement of the 3-coloring problem to the problem of checking whether a model of a PH formula is minimal. Let $G = (V,E)$ be an undirected graph such that $V = \{1,\ldots,n\}$. We associate to each vertex $i \in V$ and each color $c \in \{r,g,b\}$ a propositional variable denoted p_i^c. We also use an additional fresh variable denoted s. Then, the encoding is defined as the conjunction of the following formulæ:

$$\bigwedge_{i \in 1..n} (p_i^r \vee p_i^g \vee p_i^b) \tag{1}$$

$$\begin{array}{c} \bigwedge_{i \in 1..n} [(p_i^r \vee \neg p_i^g \vee \neg p_i^b) \wedge \\ (\neg p_i^r \vee p_i^g \vee \neg p_i^b) \wedge (\neg p_i^r \vee \neg p_i^g \vee p_i^b)] \end{array} \tag{2}$$

$$\begin{array}{c} \bigwedge_{i \in 1..n-1} [\neg p_i^r \vee \neg p_i^g \vee \neg p_i^b \vee (p_{i+1}^r \wedge p_{i+1}^g \wedge p_{i+1}^b)] \\ \wedge [\neg p_n^r \vee \neg p_n^g \vee \neg p_n^b \vee (p_1^r \wedge p_1^g \wedge p_1^b)] \end{array} \tag{3}$$

$$\bigwedge_{i<j, \{i,j\} \in E, c \in \{r,g,b\}} (\neg p_i^c \vee \neg p_j^c \vee s) \tag{4}$$

$$(\neg s \vee p_1^r) \wedge (\neg s \vee p_1^g) \wedge (\neg s \vee p_1^b) \tag{5}$$

(1) is used to express that each vertex has at least one color, and (2) expresses that if a vertex has more than one color, then it has the three considered colors. (3) is used to add the fact that if there is a vertex that has three colors, then every other vertex has also three colors. (4) means that s is true if there are two adjacent vertices that have the same color. Finally, (5)

expresses that the fact that s is true implies that the vertex 1 has three colors. In fact, we show that our encoding captures the complement of the 3-coloring problem by showing that there is no 3-coloring iff the model that associates three colors to every vertex is a minimal model.

Using the fact that each formula of the form $\neg p_1 \vee \cdots \vee \neg p_k \vee (q_1 \wedge \cdots \wedge q_l)$ is equivalent to the conjunction of Horn clauses $(\neg p_1 \vee \cdots \vee \neg p_k \vee q_1) \wedge \cdots \wedge (\neg p_1 \vee \cdots \vee \neg p_k \vee q_l)$, we know that the encoding $\phi_G = (1) \wedge (2) \wedge (3) \wedge (4) \wedge (5)$ corresponds to a PH formula.

Knowing that each clause in our encoding contains at least one positive literal, we have $m = \{s, p_1^r, p_1^g, p_1^b, \dots, p_n^r, p_n^g, p_b^b\}$ is a model of this encoding. In this context, the proof of coNP-completeness comes from the fact that m is a minimal model of our encoding iff G does not admit a 3-coloring. Using $(1) \wedge (2)$, we know that each model of ϕ_G satisfies either a unique positive literal or the three literals in $\{p_i^r, p_i^g, p_i^b\}$ for every $i \in 1..n$. In addition, using (3), we know that if a model of ϕ_G satisfies the three positive literals $\{p_i^r, p_i^g, p_i^b\}$ for some $i \in 1..n$, then this model satisfies all the positive literals in $\{p_j^r, p_j^g, p_j^b\}$ for every $j \in 1..n$. In other words, every model of ϕ_G different from m associates a single color to each vertex of G. $(4) \wedge (5)$ is used to force every model of ϕ_G to satisfy all the literals in $\{s, p_1^r, p_1^g, p_1^b\}$ if it associates the same color to two adjacent vertices. Thus, every model of ϕ_G which is different from m (and consequently contained in m) corresponds to a 3-coloring of G. Conversely, every 3-coloring of G can be easily used for defining a model of ϕ_G different from m. Thus, m is not a minimal model of ϕ_G if and only if G admits a 3-coloring. □

4 PN FORMULÆ AND MINIMAL MODELS

We here provide interesting properties that allow us to reduce the search space for solving in the case of PN formulæthe problem of enumerating all the minimal models that are contained in a given model. We also show how the problem of listing the minimal transversals can be directly used for enumerating these minimal models. Then, we use this result for defining an algorithm for enumerating all the minimal models of a PN formula.

Definition 4.1. *Let ϕ be a PN formula, m a model of ϕ and $n \in \mathbb{N}$. Then, we define $\pi(\phi, m)$ and $\nu(\phi, m)$ as follows: $\pi(\phi, m) = \{p \in Var(\phi) \mid p \in \phi_{|(m)^-}\}$ and $\nu(\phi, m) = \{\neg p \mid p \in Var(\phi)$ and $\forall c \in \psi$ w. $p \in c, \exists c' \in \psi : c' \subseteq c \setminus \{p\}\}$, where $\psi = \phi_{|(m)^- \cup \pi(\phi,m)}$.*

Note that $\phi_{|(m)^- \cup \pi(\phi,m)}$ contains only positive clauses, since ϕ is a PN formula and all the negative clauses are satisfied by satisfying the literals in $(m)^-$.

Proposition 4.1. *Given a PN formula ϕ and a model m of ϕ, $(m)^- \subseteq \nu(\phi, m)$ holds.*

Proof. This property is a direct consequence of the fact that there is no clause in $\phi_{|(m)^- \cup \pi(\phi,m)}$ that contains a variable occurring in $(m)^-$. □

One can easily see that $\pi(\phi, m)$ is included in every model which is contained in m, since this set corresponds to unit clauses occurring in ϕ after assigning the truth value 0 to the variables in $Var(\phi) \setminus m$. Moreover, from the definition of $\nu(\phi, m)$, the formula obtained by assigning 0 to the variables in $S = \{p \in Var(\phi) \mid \neg p \in \nu(\phi, m)\}$ is equisatisfiable to $\psi = \phi_{|(m)^- \cup \pi(\phi,m)}$. As a consequence, we get $m' \cap S = \emptyset$ for every minimal model $m' \subseteq m$.

Let us recall that a positive monotone formula is defined as a CNF formula that contains only positive clauses.

Proposition 4.2. *Let ϕ be a positive monotone formula s.t., for every clause $c \in \phi$, c contains at least two literals. Then, for every positive literal $p \in Lit(\phi)$, if $\exists c \in \phi$ s.t. $p \in c$ and $c' \not\subseteq c$ for every $c' \in \phi$, then there exists a minimal model m of ϕ s.t. $p \in m'$.*

Proof. Let p a literal in $Lit(\phi)$ and $c \in \phi$ s.t. $p \in c$ and $c' \not\subseteq c$ for every $c' \in \phi$. Let us consider the formula $\psi = \phi_{|\{\neg q \mid q \in c \setminus \{p\}\}}$. Clearly, ψ does not contain the empty clause since there is no clause $c' \in \phi$ s.t. $c' \subseteq c \setminus \{p\}$. Knowing that all the models of ψ contain the literal p since it is a unit clause in ψ, we deduce that there exists a minimal model of ϕ that contains p. □

The following theorem allows us to reduce the search space for solving the problem of enumerating all the minimal models in the case of PN formulæ.

Theorem 4.1. *Let ϕ be a PN formula and m a model of ϕ. Then, the following properties are satisfied: (i) $\bigcap_{m' \in MinMS(\phi,m)} m' = \pi(\phi, m)$ and (ii) $\bigcap_{m' \in MinMS(\phi,m)} (m')^- = \nu(\phi, m)$.*

Proof. We here use S (resp. S') to denote the set $\bigcap_{m' \in MinMS(\phi,m)} m'$ (resp. $\bigcap_{m' \in MinMS(\phi,m)} (m')^-$). We only consider the part $\supseteq$, the other being similar. For this part, it suffices to show the two following properties: (*i*) $\pi(\phi, m) \subseteq S$ and (*ii*) $\nu(\phi, m) \subseteq S'$. Clearly, all the minimal models in $MinMS(\phi, m)$ satisfy the literals in $(m)^-$. Moreover, all the models of $\phi_{|(m)^-}$ includes all the unit clause occurring in $\phi_{|(m)^-}$. As a consequence, $\pi(\phi, m) \subseteq S$ holds. Further, let p be a propositional variable s.t. $\forall c \in Pos(\phi_{|(m)^- \cup \pi(\phi,m)})$

Algorithm 1: An algorithm for computing the minimal models of a PN formula.

```
1: procedure MINMODSPN(φ)
2:     r ← ∅
3:     ψ ← φ
4:     while SAT(ψ) do
5:         tmp ← MinMS(φ,m)            ▷ m is a model of ψ
6:         ψ ← ψ ∧ ⋀_{m′∈tmp} ¬m′
7:         r ← r ∪ tmp
8:     return r
```

with $p \in c$, $\exists c' \in Pos(\phi_{|(m)^- \cup \pi(\phi,m)})$ s.t. $c' \subseteq c \setminus \{p\}$. Then, for every model m' of $\phi_{|(m)^- \cup \pi(\phi,m)}$, $m' \setminus \{p\}$ satisfies also $\phi_{|(m)^- \cup \pi(\phi,m)}$. Thus, $\nu(\phi,m) \subseteq S'$ holds. □

Given a positive monotone formula ϕ, we use H_ϕ to denote a hypergraph associated to ϕ and defined as the ordered pair $(Lit(\phi),\phi)$. Clearly, using the definition of H_ϕ and the notion of minimal transversal, we get the following property.

Proposition 4.3. *Let ϕ be a positive monotone CNF formula. Then, we have $MinModels(\phi) = MT(H_\phi)$.*

Thus, using Theorem 4.1, we obtain a simple characterization of the minimal models that are contained in a given model in the case of PN formulæ. Indeed, the following property shows that enumerating these minimal models is reduced to enumerating the minimal transversals of a hypergraph.

Proposition 4.4. *Let ϕ be a PN formula and m a model of ϕ. Then, we have $MinMS(\phi,m) = \{s \cup \pi(\phi,m) \mid s \in MT(H_\psi)\}$ where $\psi = \phi_{|\pi(\phi,m)\cup\nu(\phi,m)}$.*

Algorithm 1 shows how the method described in Proposition 4.4 can be simply used for enumerating all the minimal models of a PN formula. Indeed, given a PN formula ϕ, MINMODSPN(ϕ) uses $MinMS(\phi,m)$ to get all the minimal models that are contained in m in every step of the while loop (see Line 5). The soundness of Algorithm 1 is a direct consequence of Proposition 4.4. It is worth mentioning that we add in Line 6 the negations of the found minimal models to avoid finding the same minimal model more than once.

As a side note, connections between minimal transversals and minimal models have already been made in the literature (e.g. (Reiter, 1987)). Moreover, the graph structure has already been used for computing minimal models through the notion of dependency graph (Ben-Eliyahu-Zohary, 2000; Ben-Eliyahu-Zohary, 2005; Angiulli et al., 2014).

Algorithm 2: An algorithm for computing a minimal models of a PN formula contained in a given model.

```
procedure SOLVEMMP(φ, m)
    r ← m
    for p ∈ m do
        if MOD(r \ {p}, φ) then
            r ← r \ {p}
    return r
```

5 A SAT-BASED APPROACH FOR ENUMERATING MINIMAL MODELS

In this section, we provide a SAT-based encoding for enumerating all the minimal models of a PN formula. the main idea consists in using a formula expressing the fact that the truth value 1 is assigned to a propositional variable if and only if there is a positive clause which is satisfied only by this assignment.

The following proposition provides a characterization of the minimal models in the case of PN formulæ.

Proposition 5.1. *Given a PN formula ϕ and a model m of ϕ, m is a minimal model of ϕ iff for all $p \in m$, there exists a clause $c \in Pos(\phi)$ s.t. $p \in c$ and $c \setminus \{p\} \subseteq Var(\phi) \setminus m$.*

Proof.
Part $\Rightarrow$. Let m be a minimal model of ϕ. Assume that there exists $p \in m$ s.t. for every clause c in $Pos(\phi)$ containing p, we have $c \setminus \{p\} \nsubseteq Var(\phi) \setminus m$. Thus, $m \setminus \{p\}$ satisfies all the clauses containing p, and consequently, it satisfies ϕ. Therefore, we get a contradiction since m is a minimal model of ϕ.
Part $\Leftarrow$. Let m be a model of ϕ s.t., for all $p \in m$, there exists a clause $c \in Pos(\phi)$ s.t. $p \in c$ and $c \setminus \{p\} \subseteq Var(\phi) \setminus m$. Then, for all $p \in m$, $\phi \wedge (\bigwedge_{q \in Var(\phi)\setminus m} \neg q) \wedge \neg p$ is unsatisfiable. As a consequence, m is a minimal model of ϕ. □

Now, using the characterization provided in Proposition 5.1, we propose a SAT-based encoding for enumerating all the minimal models of a PN formula. Given a PN formula ϕ, we use $\mathcal{E}_{PN}(\phi)$ to denote the following SAT encoding: $\phi \wedge (\bigwedge_{p \in Var(\phi)} p \rightarrow \bigvee_{c \in Pos(\phi), q \in c} \neg(c \setminus \{q\}))$.

Let us recall that any propositional formula can be translated to CNF using Tseitin's linear encoding (Tseitin, 1968). To translate our encoding to CNF, one can associate to each subformula of the form $c \setminus \{p\}$ a fresh propositional variable r with the clauses $(\neg r \vee c \setminus \{p\}) \wedge \bigwedge_{q \in c \setminus \{p\}} (r \vee \neg q)$ that represent the equivalence $r \leftrightarrow c \setminus \{p\}$.

Theorem 5.1 (Soundness). *Let ϕ be a PN formula. Then, $Models(\mathcal{E}_{PN}(\phi)) = MinModels(\phi)$ holds.*

Proof. The soundness of the SAT encoding $\mathcal{E}_{PN}(\phi)$ is a consequence of Porposition 5.1. Indeed, the subformula $(\bigwedge_{p \in Var(\phi)} p \rightarrow \bigvee_{c \in Pos(\phi), p \in c} \overline{c \setminus \{p\}})$ allows us to get the property that for all $p \in m$, there exists a clause $c \in Pos(\phi)$ s.t. $p \in c$ and $\{\neg q \mid q \in c \setminus \{p\}\} \subseteq m$. □

As mentioned in our proof of Theorem 3.1, this encoding shows that the problem of checking whether a model is minimal (called minimal model checking problem) in the case of PN formulæ is tractable, since checking whether a Boolean interpretation is a model of a CNF formula or not can be performed in linear time.

Let us now consider a problem related to minimal model computation, called model minimization problem (Angiulli et al., 2014).

Definition 5.1 (Model Minimization Problem). *Given a propositional formula ϕ and a model m of ϕ, compute a minimal model m' for ϕ contained in the model m.*

The model minimization problem is $P^{NP[O(log(n))]}$-hard in the general case where n is the number of variables (Angiulli et al., 2014). This means that this problem is as hard as a problem that can be solved using a polynomial deterministic algorithm which uses $O(log(n))$ calls of an NP oracle.

Theorem 5.2. *The model minimization problem is tractable in the case of PN formulæ.*

Proof. We here show that Algorithm 2 is sound. In this algorithm the procedure $\text{MOD}(r \setminus \{p\}, \phi)$ in Line 4 is used for checking whether $r \setminus \{p\}$ is a model of ϕ or not. Clearly, the returned Boolean interpretation r is a model of ϕ and contained in m. Assume now that this returned interpretation is not a minimal model of ϕ. Then using Proposition 5.1, there exists $p \in r$ s.t. $c \setminus \{p\} \cap r \neq \emptyset$ for every clause c containing the literal p. Thus, for all m' with $r \subseteq m'$, we get $c \setminus \{p\} \cap m \neq \emptyset$ for every clause c containing the literal p. As a consequence, for all m' with $r \subseteq m'$, $m' \setminus \{p\}$ is a model of ϕ. Therefore, $p \notin r$ holds and we get a contradiction. □

6 MINIMAL MODEL ENUMERATION FOR PH FORMULÆ

In this section, we propose an approach for enumerating all the minimal models in the case of PH formulæ.

Algorithm 3: An algorithm for computing the minimal models of a PH formula.

procedure MINMODSPH(ϕ)
  $r \leftarrow \emptyset$
  $\psi \leftarrow \mathcal{E}_{PN}(Pos(\phi) \cup Neg(\phi))$
  while SAT(ψ) **do** ▷ m is a minimal model of $Pos(\phi) \cup Neg(\phi)$
    $m' \leftarrow m \cup pup(M(\phi)_{|m})$
    if $(\forall c \in Neg(\phi),\ \{p \mid \neg p \in c\} \nsubseteq m')$ **then**
      $r \leftarrow r \cup \{m'\}$
    $\psi \leftarrow \psi \wedge \overline{m'}$
  return r

This approach is based on enumerating the minimal models of its greatest PN subformula (the subformula consisting of all the positive and negative clauses), which means that our methods for enumerating the minimal models in the case of PN formulæ can be used in the case of PH formulæ.

Unit propagation is a procedure consisting in applying the following two rules for every unit clause l occurring in the CNF formula: (*i*) every clause containing the literal l, except the unit clause l, is removed; and (*ii*) all the occurrences of $\bar{l}$ are removed. For instance, applying unit propagation to the formula $\{\neg p \vee \neg q, p, p \vee r, q \vee r\}$ produce the formula $\{\neg q, p, r\}$. Indeed, using the two rules with the literal p produces the unit clause $\neg q$ and removes $p \vee r$, and then, we get the literal r from the clause $q \vee r$ by propagating $\neg q$. It is worth noting that an efficient implementation of unit propagation in linear time is provided in (Crawford and Auton, 1993). In this work, we use $pup(\phi)$ to denote the set of positive literals that are propagated in the procedure of unit propagation. Considering again the previous example, we get $pup(\{\neg p \vee \neg q, p, p \vee r, q \vee r\}) = \{p, r\}$.

Given a PH formula ϕ, we use $M(\phi)$ to denote the set of clauses $\phi \setminus (Pos(\phi) \cup Neg(\phi))$. Note that each clause in $M(\phi)$ contains at least one negative literal and exactly one positive literal.

Theorem 6.1. *Let ϕ be a PH formula. Then, $MinModels(\phi) = \{m \cup pup(M(\phi)_{|m}) \mid m \in MinModels(Pos(\phi) \cup Neg(\phi))\ \&\ \forall c \in Neg(\phi), \{p \mid \neg p \in c\} \nsubseteq m \cup pup(M(\phi)_{|m})\}$ holds.*

Proof. We use in this proof S to denote the set $\{m \cup pup(M(\phi)_{|m}) \mid m \in MinModels(Pos(\phi) \cup Neg(\phi))\ \&\ \forall c \in Neg(\phi), \{p \mid \neg p \in c\} \nsubseteq m \cup pup(M(\phi)_{|m})\}$.
Part $\subseteq$. Let m be a minimal model in $MinModels(\phi)$. Then, we know that there exists a minimal model m' of $Pos(\phi) \cup Neg(\phi)$ s.t. $m' \subseteq m$ since $Pos(\phi) \cup Neg(\phi)$ is a subformula of ϕ. Clearly, we have $pup(M(\phi)_{|m'})) \subseteq m \setminus m'$. Moreover, knowing that $\forall c \in Neg(\phi), \{p \mid \neg p \in c\} \nsubseteq m$ since m satisfies ϕ, $\{p \mid \neg p \in c\} \nsubseteq m' \cup pup(M(\phi)_{|m'})$ holds for every

$c \in Neg(\phi)$. As a consequence, using the fact that m is a minimal model of ϕ, we get $m = m' \cup pup(\phi_{|m'})$. Thus, m belongs to the set S.
Part $\supseteq$. Let $m = m' \cup pup(M(\phi)_{|m'})$ an element of S. Then, using the property $\forall c \in Neg(\phi), \{p \mid \neg p \in c\} \not\subseteq m$ and the fact that m' satisfies $\psi = Pos(\phi) \cup Neg(\phi)$, we know that m is a model of ϕ. Further, for all $p \in m'$, $\phi \wedge \bigwedge_{q \notin m} \neg q \wedge \neg p$ is unsatisfiable since $m' \in MinModels(\psi)$. We know also that for all $p \in pup(M(\phi)_{|m})$, $\phi \wedge \bigwedge_{q \notin m} \neg q \wedge \neg p$ is unsatisfiable since all the elements of $pup(M(\phi)_{|m})$ are logical consequences of $\phi_{|m'}$. Thus, m is a minimal model of ϕ. □

Theorem 6.1 shows how a method for enumerating the minimal models of a PN formula can be used to define a method in the case of PH formulæ. For instance, Algorithm 3, shows how our SAT-based encoding proposed in Section 5 can be used for generating all the minimal models of a PH formula. Indeed, given a PH formula ϕ, MINMODSPH(ϕ) uses the SAT-based encoding $\mathcal{E}_{PN}(Pos(\phi) \cup Neg(\phi))$ for generating the minimal models of $Pos(\phi) \cup Neg(\phi)$.

It is noteworthy that, using the fact that each binary CNF formula is also a PH formula, we get the following property.

Corollary 6.1. *Let ϕ be a binary CNF formula. $MinModels(\phi) = \{m \cup pup(M(\phi)_{|m}) \mid m \in MinModels(Pos(\phi) \cup Neg(\phi))$ & $\forall c \in Neg(\phi), \{p \mid \neg p \in c\} \not\subseteq m \cup pup(M(\phi)_{|m})\}$ holds.*

7 A MODELING EXAMPLE

We here describe a modeling example showing that the problem of enumerating the minimal models in the case of PN formulæ can be used for solving the datamining problem of enumerating the minimal generators (Zaki, 2004).

Let $\mathcal{T}$ be a set of items. From now on, we assume $\mathcal{T}$ fixed. A *transaction* over $\mathcal{T}$ is an ordered pair (tid, I) where $tid \in \mathbb{N}$ is its *identifier* and $I \subseteq \mathcal{T}$ is an *itemset*. A *transaction database* is a finite set of transactions where each identifier refers to only one transaction. Given an itemset J, we say that (tid, I) supports J if we have $J \subseteq I$. Further, the *cover* of an itemset J in a database D, denoted $\mathcal{C}(J, D)$, is defined as follows: $\mathcal{C}(J, D) = \{tid \mid (tid, I) \in D \,\&\, J \subseteq I\}$.

Definition 7.1 (Closed Itemset)**.** *Let D be a transaction database and I an itemset. Then, I is closed if, for every itemset J with $I \subset J$, $\mathcal{C}(J, D) \subset \mathcal{C}(I, D)$ holds.*

Definition 7.2 (Minimal Generator)**.** *Let D be a transaction database and I an itemset. Then, I is a minimal generator if, for every itemset J with $J \subset I$, $\mathcal{C}(I, D) \subset \mathcal{C}(J, D)$ holds.*

Table 1: Transaction database D.

tid	itemset
1	a,b,c,d,e
2	a,b,c,f
3	a,b,c,m
4	a,c,d,f,j
5	j,l
6	e,f

Note that for every minimal generator I, there exists a single closed itemset J such that $\mathcal{C}(I, D) = \mathcal{C}(J, D)$, and we say that I generates J.

Consider the transaction database D described in Table 1. In this database, $\{a,b,c\}$ and $\{a,c\}$ are closed itemsets and $\{a,b\}$ and $\{a\}$ are minimal generators that generate $\{a,b,c\}$ and $\{a,c\}$ respectively.

To define the encoding of the problem of generating all the minimal generators, we associate to each item a a variable denoted p_a which is used to represent the fact that a is in the minimal generator. Further, for all $i \in 1..n$, the variable q_i is used to represent the fact that the itemset appears in the transaction t_i. The encoding $\mathcal{MG}(D)$ is $\bigwedge_{i=1}^{n}(\neg q_i \leftrightarrow \bigvee_{a \notin \mathcal{T} \setminus t_i} p_a)$. Clearly, each subformula of the form $\neg q_i \leftrightarrow \bigvee_{a \notin t_i} \neg p_a$ is equivalent to the following PN formula: $(q_i \vee \bigvee_{a \notin \mathcal{T} \setminus t_i} p_a) \wedge \bigwedge_{a \notin \mathcal{T} \setminus t_i}(\neg p_a \vee \neg q_i)$.

It is important to note that the encoding $\mathcal{MG}(D)$ is similar to that proposed in (Jabbour et al., 2017) for generating the frequent itemsets. Furthermore, a SAT-based encoding for generating the minimal generators has been proposed in (Boudane et al., 2017).

Proposition 7.1. *Let $D = \{(1,t_1), \dots, (n,t_n)\}$ be a transaction database over the set of items $\mathcal{T}$. Then, m is a minimal model of $\mathcal{MG}(D)$ iff $I_m = \{a \mid p_a \in m\}$ is a minimal generator in D such that $\mathcal{C}(I_m, D) = \{i \in 1..n \mid q_i \in m\}$.*

Proof. We only consider the *only if part*, the other part being similar. Let m be a minimal model of $\mathcal{MG}(D)$. Then, $\mathcal{C}(I_m, D) = \{i \in 1..n \mid m(q_i) = 1\}$ holds. Assume that I_m is not a minimal generator. Then there exists $a \in I_m$ such that $\mathcal{C}(I_m, D) = \mathcal{C}(I_m \setminus \{a\}, D)$. Then, $m \setminus \{p_a\}$ is a model of $\mathcal{MG}(D)$. Thus, we get a contradiction since m is a minimal model of $\mathcal{MG}(D)$. Therefore, I_m is a minimal generator in the transaction database D. □

8 CONCLUSION AND PERSPECTIVES

We proposed approaches for generating all the minimal models in two particular classes of CNF formulæ. The first class is that of PN formulæ which are defined as CNF formulæ where each clause is either positive or negative, whereas the second class is that of PH formulæ in which each clause is either positive or a Horn clause. The first contribution consists in showing that the problem of determining whether a model is minimal is tractable in the case of PN formulæ, whereas it is coNP-complete in the case of PH formulæ. Then, we introduced our first approach for enumerating all the minimal models of a PN formula, which is based on the use of an algorithm for generating the minimal transversals. We also proposed a SAT-based encoding for enumerating all the minimal models of a PN formula. Next, we provided an interesting characterization of the minimal models in the case of PH formulæ that allows us to use our approaches in the case of PN formulæ for enumerating the minimal models for the PH formulæ. Finally, we described a simple modeling example in datamining.

As a future work, we intend to implement and evaluate the proposed methods for generating the minimal models. We also plan to use similar approaches for other formula classes.

REFERENCES

Angiulli, F., Ben-Eliyahu-Zohary, R., Fassetti, F., and Palopoli, L. (2014). On the tractability of minimal model computation for some CNF theories. *Artificial Intelligence*, 210:56–77.

Avin, C. and Ben-Eliyahu-Zohary, R. (2001). Algorithms for computing x-minimal models. In *Logic Programming and Nonmonotonic Reasoning, 6th International Conference, LPNMR 2001, Vienna, Austria, Proceedings*, pages 322–335.

Ben-Eliyahu, R. and Dechter, R. (1996). On computing minimal models. *Annals of Mathematics and Artificial Intelligence*, 18(1):3–27.

Ben-Eliyahu-Zohary, R. (2000). A demand-driven algorithm for generating minimal models. In *Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on on Innovative Applications of Artificial Intelligence, USA*, pages 267–272.

Ben-Eliyahu-Zohary, R. (2005). An incremental algorithm for generating all minimal models. *Artificial Intelligence*, 169(1):1–22.

Ben-Eliyahu-Zohary, R., Angiulli, F., Fassetti, F., and Palopoli, L. (2017). Modular construction of minimal models. In *Logic Programming and Nonmonotonic Reasoning - 14th International Conference, LPNMR 2017, Finland, Proceedings*, pages 43–48.

Bidoit, N. and Froidevaux, C. (1987). Minimalism subsumes default logic and circumscription in stratified logic programming. In *Proceedings of the Symposium on Logic in Computer Science (LICS '87), USA*, pages 89–97.

Boudane, A., Jabbour, S., Sais, L., and Salhi, Y. (2017). Enumerating non-redundant association rules using satisfiability. In *Advances in Knowledge Discovery and Data Mining - 21st Pacific-Asia Conference, PAKDD 2017, South Korea, Proceedings, Part I*, pages 824–836.

Boumarafi, Y., Sais, L., and Salhi, Y. (2017). From SAT to maximum independent set: A new approach to characterize tractable classes. In *LPAR-21, 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Botswana*, pages 286–299.

Cadoli, M. (1992a). The complexity of model checking for circumscriptive formulae. *Information Processing Letters*, 44(3):113–118.

Cadoli, M. (1992b). On the complexity of model finding for nonmonotonic propositional logics. In *4th Italian conference on theoretical computer science*, pages 125–139.

Crawford, J. M. and Auton, L. D. (1993). Experimental results on the crossover point in satisfiability problems. In *Proceedings of the 11th National Conference on Artificial Intelligence (AAAI-93), USA.*, pages 21–27.

Fredman, M. L. and Khachiyan, L. (1996). On the complexity of dualization of monotone disjunctive normal forms. *Journal of Algorithms*, 21(3):618–628.

Gelfond, M. and Lifschitz, V. (1988). The stable model semantics for logic programming. In *Logic Programming, Proceedings of the Fifth International Conference and Symposium, Seattle, Washington, (2 Volumes)*, pages 1070–1080.

Hébert, C., Bretto, A., and Crémilleux, B. (2007). A data mining formalization to improve hypergraph minimal transversal computation. *Fundamenta Informaticae*, 80(4):415–433.

Jabbour, S., Sais, L., and Salhi, Y. (2017). Mining top-k motifs with a sat-based framework. *Artificial Intelligence*, 244:30–47.

Khachiyan, L., Boros, E., Elbassioni, K. M., and Gurvich, V. (2005). A new algorithm for the hypergraph transversal problem. In *Computing and Combinatorics, 11th Annual International Conference, COCOON 2005, China, Proceedings*, pages 767–776.

Khachiyan, L., Boros, E., Elbassioni, K. M., and Gurvich, V. (2007). A global parallel algorithm for the hypergraph transversal problem. *Information Processing Letters*, 101(4):148–155.

McCarthy, J. (1980). Circumscription - a form of nonmonotonic reasoning. *Artificial Intelligence*, 13(1-2):27–39.

Niemelä, I. (1996). A tableau calculus for minimal model reasoning. In *Theorem Proving with Analytic Tableaux and Related Methods, 5th International Workshop, TABLEAUX'96, Italy, 1996, Proceedings*, pages 278–294.

Reiter, R. (1987). A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–95.

Tseitin, G. (1968). On the complexity of derivations in the propositional calculus. In Slesenko, H., editor, *Structures in Constructives Mathematics and Mathematical Logic, Part II*, pages 115–125.

Zaki, M. J. (2004). Mining non-redundant association rules. *Data Mining and Knowledge Discovery*, 9(3):223–248.