EPiC Series in Computing

Volume 46, 2017, Pages 286–299

LPAR-21. 21st International Conference on Logic for
Programming, Artificial Intelligence and Reasoning

# From SAT to Maximum Independent Set: A New Approach to Characterize Tractable Classes

Yazid M. Boumarafi, Lakhdar Sais, and Yakoub Salhi

CRIL - CNRS, Artois University Lens France
{boumarafi,sais,salhi}@cril.fr

**Abstract**

In this paper, we propose a new approach for defining tractable classes in propositional satisfiability problem (in short SAT). The basic idea consists in transforming SAT instances into instances of the problem of finding a maximum independent set. In this context, we only consider propositional formulæ in conjunctive normal form where each clause is either positive or binary negative. Tractable classes are obtained from existing polynomial time algorithms of the problem of finding a maximum independent set in the case of different graph classes, such as claw-free graphs and perfect graphs. We show, in particular, that the pigeonhole principle belongs to one of the defined tractable classes. Furthermore, we propose a characterization of the minimal models in the largest considered fragment based on the maximum independent set problem.

## 1 Introduction

The propositional satisfiability problem (in short SAT) consists in determining whether there exists a truth assignment making a propositional formula true in classical logic. It is one of the central problems in computational complexity theory and computer science. The last years have seen remarkable breakthrough in practical solving of SAT. Indeed, modern SAT solvers are today able to solve different kind of application instances with several millions of variables and clauses.

SAT, has also attracted much attention in theoretical computer science since it is a canonical NP-complete problem [10]. Several classes of propositional formulæ have been identified for which satisfiability is solvable in polynomial time (e.g. Horn [17], Renamable Horn [27], Q-Horn [5],Krom [26]).

Characterizing new tractable classes remains important for both SAT and AI in general, since many important problems in AI can be reduced to SAT. For example, many knowledge representation systems are based on Horn formulæ for which SAT can be solved in linear time. Moreover, more general formulæ are sometimes approximated to tractable formulæ in the approach known as knowledge compilation (e.g. [37, 4, 7]). We also believe that extending current polynomial fragments of SAT might be a cornerstone step towards understanding the practical effectiveness of SAT solvers. Characterizing new tractable classes, would help to get more insights on the tractability/intractability of SAT instances.

Despite the fact that other well-known NP-Complete problems admit their own tractable classes, there are few works addressing the issue of finding their equivalent counterparts in SAT. For example several polynomial-time algorithms that solve the problem exactly for graphs belonging to special classes. For example, several known problems admits polynomial algorithms for Perfect [21], convex and chordal [15], AT-free [38] and claw-free graphs [18] (see [6] for a survey). In [32], the authors show that CNF formulæ with $\beta$-acyclic hypergraphs can be solved in polynomial time. One can also mention several recent tractable classes highlighted by different authors in constraint satisfaction problems (CSP) (e.g. [12]). As fare as we know, only few works addressed the issue of finding their equivalent SAT tractable classes (e.g. [33]).

Considering polynomial reducibility, one of the most fundamental tool in complexity theory, in this paper, we characterize new tractable classes in SAT thanks to tractability results obtained in maximum independent set problem (e.g. [31, 29]). Without loss of generality, we consider propositional formulæ in conjunctive normal form (CNF) made of a set of Positive clauses and a set of Binary Negative clauses (PBN). Indeed, any CNF formula, can be linearly translated while preserving satisfiability to PBN form using the well known Tseitin's encoding. This CNF form is more suitable for establishing connections with problems in graph theory and also with constraint satisfaction problems. For example, the direct encoding [39] of any binary CSP to CNF is a PBN formula. By reasoning on the occurrences of the variables in the positive part of the formula, we define an interesting hierarchy of propositional formulæ. Our approach for establishing new tractable classes, heavily exploits the reduction of the satisfiability of each fragment of the hierarchy to the problem of finding a maximum independent set. This allows us to find the equivalent counterparts of some well-known tractable classes of the maximum independent set problem (e.g. claw-free graphs). We also identify that the well known Pigeon-Hole problem can be mapped to a particular class, the claw-free graphs. As a summary, the contribution of this paper consists mainly in proposing a fairly simple framework to establish additional tractability results in SAT based on a connection between SAT in the case of PBN formulæ and the problem of of finding a maximum independent set. Furthermore, we provide a characterization of the minimal models in the largest considered fragment based on the maximum independent set problem. This characterization is used to highlight a connection with the problem of computing minimal models an essential task in many reasoning systems in Artificial Intelligence, including propositional circumscription [30, 28], minimal diagnosis [35, 16] and in answering queries posed on logic program (under stable model semantics) [20]. Also, this last contribution allows us to show the wealth of the connections that can be observed from the cross-fertilization between different problems.

The paper is organized as follows. After some preliminary background about propositional logic and SAT, we introduce our hierarchy of PBN formulæ. An incremental presentation of the tractable classes associated to the different fragments of the hierarchy starting from the most specific one is presented. We then describe our characterization of the minimal models in the largest considered fragment. Then, we discuss some related works, mainly those proposed in the context of constraint satisfaction problems before concluding.

## 2 Propositional Logic and SAT Problem

We assume that the reader is familiar with the main concepts of propositional logic. So, we limit our presentation to some notations used throughout the paper. A propositional formula in conjunctive normal form (CNF) is a conjunction ($\wedge$) of clauses, where a *clause* is a disjunction ($\vee$) of literals. A *literal* is a propositional variable ($p$), called positive literal, or a negated propositional variable ($\neg p$), called negative literal. A clause is *binary* if it contains only two literals. A clause is *positive* (resp. *negative*) if it contains only positive (resp. negative) literals. A CNF formula can also be seen as a set of clauses, and a clause as a set of literals. The size of the CNF formula $\phi$ corresponds to the value $\sum_{c \in \phi} |c|$ where $|c|$ is the number of literals in the clause $c$. Given a syntactic object $S$, we use $\mathcal{P}(S)$ to denote the set of propositional variables appearing in $S$.

A *Boolean interpretation* $\mathcal{I}$ of a formula $\phi$ is defined as a function from $\mathcal{P}(\phi)$ to $\{0, 1\}$ (0 corresponds to $false$ and 1 to $true$). A *model* of a formula $\phi$ is a Boolean interpretation $\mathcal{I}$ that satisfies $\phi$, i.e. $\mathcal{I}(\phi) = 1$. A formula $\phi$ is consistent if there exists a model of $\phi$. The $SAT$ *problem* consists in deciding whether a given CNF formula admits a model or not.

Let $c_i$ and $c_j$ be two clauses containing $x$ and $\neg x$ respectively. A resolvent between $c_i$ and $c_j$ on $x$ is defined as $r = c_i \cup c_j \backslash \{x, \neg x\}$. Let us note that $r$ is a logical consequence of $c_i \wedge c_j$ i.e. every model of $c_i$ and $c_j$ is also a model of $r$. Resolution rule is defined as the process of generating a resolvent from two parent clauses.

# 3 PBN formulæ

In this section, we describe forms of the propositional formulæ that we consider.

**Definition 1** (PBN formula)**.** *A PBN formula $\phi$ is a CNF formula where each clause is either a positive clause or a binary negative clause. We use $Pos(\phi)$ (resp. $Neg(\phi)$) to denote the set of its positive (resp. negative) clauses.*

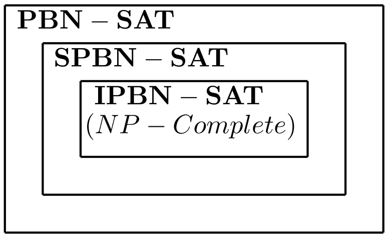

Figure 1: A hierarchy of syntactic fragments of SAT

There are different manners to transform each CNF formula into an equi-satisfiable PBN formula, in particular, using Tseitin's linear encoding. Indeed, we only have to associate to each negative literal $l$ a fresh propositional variable, denoted $r_l$, and replace each clause of the form $p_1 \vee \cdots \vee p_m \vee \neg q_1 \vee \cdots \vee \neg q_n$ with the following set of clauses: $\{p_1 \vee \cdots \vee p_m \vee r_{\neg q_1} \vee \cdots \vee r_{\neg q_n}, \neg q_1 \vee \neg r_{\neg q_1}, \ldots, \neg q_n \vee \neg r_{\neg q_n}\}$. As we can observe, the first clause can be obtained from several applications of the resolution rule on the last set of clauses. Obviously deciding the satisfiability of PBN formulæ remains NP-Complete.

We now define two particular syntactic fragments of PBN formulæ based on the occurrences of the positive literals.

**Definition 2** (SPBN formula)**.** *An SPBN formula $\phi$ is a PBN formula where, for all $c \in Pos(\phi)$, $|\mathcal{P}(Pos(\phi) \setminus \{c\}) \cap c| \leqslant 1$.*

**Definition 3** (IPBN formula)**.** *An IPBN formula $\phi$ is a PBN formula where, for all $c, c' \in Pos(\phi)$ with $c \neq c'$, $c \cap c' = \emptyset$.*

In other words, an SPBN formula is a PBN formula where each positive clause shares at most one literal with the other positive clauses, while in IPBN formula each positive clause does not share any literal with the other positive clauses.

**Example 1.** *For instance, the following formula is in SPBN form:*
$(p \vee q) \wedge (p \vee r) \wedge (p \vee s) \wedge (t \vee u) \wedge (\neg q \vee \neg r) \wedge (\neg r \vee \neg s) \wedge (\neg p \vee \neg u)$
*Indeed, each positive clause shares at most one literal with the other positive clauses.*

Let us now consider the pigeon-hole principle which is widely used in proof complexity. It asserts that there is no injective mapping from $b$ pigeons to $n$ holes as long as $b > n$. We use $php_n^b$ to denote the pigeon-hole principle with $b$ pigeons and $n$ holes. Cook proved that the Pigeon-Hole principle admits a short proof in extended resolution proof system [11]. A short proof also exists using resolution with symmetry [1, 3]. The pigeon-hole principle $php_n^b$ is expressed using the following IPBN formula $\phi_{php_n^b}$:

$$\bigvee_{j=1}^{n} p_{ij}, 1 \leqslant i \leqslant b \tag{1}$$

$$\neg p_{ij} \vee \neg p_{kj}, 1 \leqslant i < k \leqslant b \textit{ and } 1 \leqslant j \leqslant n \tag{2}$$

where $p_{ij} = 1$ iff the pigeon $i$ is put in the hole $j$.

Indeed, each positive clause of $\phi_{php_n^b}$ does not share any literals with the other positive clauses. We show later that the problem of checking the satisfiability of the formula $\phi_{php_n^b}$ belongs to a tractable class that we define.

From now on, we use PBN-SAT, SPBN-SAT and IPBN-SAT to denote the SAT problem in the cases of PBN formulæ, SPBN formulæ and IPBN formulæ respectively.

**Proposition 1** (IPBN-SAT)**.** *The problem of checking the satisfiability of an IPBN formula is NP-complete.*

*Proof.* Clearly IPBN-SAT is in NP since SAT is in NP. We now show that IPBN-SAT is an NP-hard problem using the graph coloring problem. It is well-known that deciding whether there is a 3-coloring of a graph is an NP-hard problem. 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 sets $P = \{V_{c_1}, V_{c_2}, V_{c_3}\}$ 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. In order to encode the 3-coloring problem into IPBN-SAT, we associate to each vertex $v$ and each color $c_i$ a propositional variable denoted $p_v^{c_i}$. We have $\mathcal{I}(p_v^{c_i}) = 1$, if the vertex $v$ is colored with $c_i$, and $\mathcal{I}(p_v^{c_i}) = 0$ otherwise. Then, the encoding is defined as follows:

$$\bigwedge_{v \in V} p_v^{c_1} \vee p_v^{c_2} \vee p_v^{c_3} \tag{3}$$

$$\bigwedge_{\{v,v'\} \in E} (\neg p_v^{c_1} \vee \neg p_{v'}^{c_1}) \wedge (\neg p_v^{c_2} \vee \neg p_{v'}^{c_2}) \wedge (\neg p_v^{c_3} \vee \neg p_{v'}^{c_3}) \tag{4}$$

Clearly, the formula (3) is a set of pairwise independent positive clauses and, consequently, (3) ∧ (4) is an IPBN formula. Thus, IPBN-SAT is an NP-Complete problem. □

It is worth noticing that the NP-completeness of IPBN-SAT can also be obtained from constraint satisfaction problems. Indeed, the direct encoding [39] of any binary CSP to CNF is an IPBN formula.

A propositional variable occurring with only one polarity in a CNF formula is called *pure.* The *pure literal rule* consists in removing the clauses containing pure literals from a CNF formula. Clearly, a CNF formula is satisfiable if and only if the CNF formula obtained from it by applying the pure literal rule is satisfiable. *From now on, we only consider PBN formulæ having no pure literals.*

Given a PBN formula $\phi$, we use $\mathcal{R}(\phi)$ to denote the PBN formula $\phi \cup \{\neg p \vee \neg q \mid p \neq q \textit{ and } \exists c \in Pos(\phi) \textit{ s.t. } \{p, q\} \subseteq c\}$.

**Proposition 2.** *Given an SPBN formula $\phi$, $\phi$ is satisfiable iff $\mathcal{R}(\phi)$ is satisfiable.*

*Proof.* Since $\phi$ is a subset of $\mathcal{R}(\phi)$, if $\mathcal{R}(\phi)$ is satisfiable then so is $\phi$. Assume now that $\phi$ is satisfiable. Let $\mathcal{I}$ be a model of $\phi$. Given a positive clause $c$ in $Pos(\phi)$, we use $l(c, \mathcal{I})$ to denote a positive literal in $c$ s.t. $(i)$ $\mathcal{I}(l(\mathcal{I}, c)) = 1$ and $(ii)$ if $|\mathcal{P}(\phi \setminus \{c\}) \cap c| = 1$ and $\mathcal{I}(\mathcal{P}(\phi \setminus \{c\}) \cap c) = 1$ then $l(\mathcal{I}, c) = \mathcal{P}(\phi \setminus \{c\}) \cap c$. We now define a Boolean interpretation $\mathcal{I}'$ of $\mathcal{R}(\phi)$ as follows:

$$\mathcal{I}'(p) = \begin{cases} 1 & if\ \exists c \in Pos(\phi), p = l(c, \mathcal{I}) \\ 0 & otherwise \end{cases}$$

Clearly, $\mathcal{I}$ satisfies exactly one literal in each positive clause of $\phi$. Indeed, assume that there exists a positive clause $c$ containing two distinct literals $p$ and $q$ s.t. $\mathcal{I}'(p) = \mathcal{I}'(q) = 1$. Then, $c$ shares the literals $p$ and $q$ with other positive clauses. Thus, we get a contradiction since $\phi$ is an SPBN formula. As a consequence, $\mathcal{I}'$ is a model of $\mathcal{R}(\phi)$. □

Intuitively, the previous proposition shows that we only need to satisfy one literal in each positive clause.

We use an incremental presentation of our results. We start by describing our approach for characterizing tractable classes in IPBN-SAT. Then, these classes are generalized by considering SPBN-SAT. Finally, we generalize our approach for characterizing tractable classes in PBN-SAT.

# 4 Tractable Classes in IPBN-SAT

In this section, we define tractable classes in IPBN-SAT. Our approach consists in transforming the IPBN-SAT instances into instances of the problem of finding a maximum independent set. The latter has polynomial time algorithms for different graph classes. In particular, we show that the pigeon-hole principle belongs to a tractable class defined using this approach.

Given an IPBN formula $\phi$, we use $G^{\mathcal{P}}_{\phi}$ to denote the undirected graph $(\mathcal{P}(\phi), E)$ where $\{p, q\} \in E$ if and only if the propositional variables $p$ and $q$ appears in a same clause in $\phi$, i.e., $\exists c \in \phi$ s.t. $\{p, q\} \subseteq \mathcal{P}(c)$. For instance, the graph corresponding to the following formula is described in Figure 2: $(p \vee q \vee r) \wedge (s \vee t) \wedge (\neg q \vee \neg s) \wedge (\neg r \vee \neg s) \wedge (\neg r \vee \neg t)$

Given an undirected graph $G = (V, E)$, an *independent set* of $G$ is a subset of vertices $S \subseteq V$ such that no two vertices in $S$ are adjacent, i.e., for all $v, v' \in S$ with $v \neq v'$, $\{v, v'\} \notin E$. A

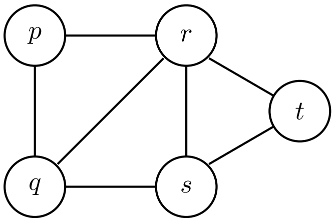

Figure 2: A graph associated to an IPBN formula

*maximum independent set* of $G$ is an independent set of largest possible size for $G$. We use $\alpha(G)$ to denote this maximum size.
The problem of finding a maximum independent set is a well-known NP-hard problem (see e.g. [6]).

Consider, for instance, the graph $G$ described in Figure 2. Then, we have $\alpha(G) = 2$ and its maximum independent sets are $\{p, s\}$, $\{p, t\}$ and $\{q, t\}$.

**Theorem 1.** *Given an IPBN formula $\phi$, $\phi$ is satisfiable iff $\alpha(G^{\mathcal{P}}_{\phi}) \geqslant n$ where $n = |Pos(\phi)|$.*

*Proof.* **Part** $\Rightarrow$. Using Proposition 2, we know that $\phi$ is satisfiable iff $\mathcal{R}(\phi)$ is satisfiable. Let $\mathcal{I}$ be a model of $\mathcal{R}(\phi)$ and $S = \{p \in \mathcal{P}(\phi) \mid \mathcal{I}(p) = 1\}$. Since $\mathcal{I}$ satisfies exactly one literal in each positive clause, $|S| \geqslant n$ holds. It suffices now to show that $S$ is an independent set of $G^{\mathcal{P}}_{\phi}$. Let us assume that there exists $p, q \in S$ such that $\{p, q\}$ is an edge in $G^{\mathcal{P}}_{\phi}$. Then, the clause $\neg p \vee \neg q$ belongs to $\mathcal{R}(\phi)$. Thus, we get a contradiction since $\mathcal{I}(p) = 0$ or $\mathcal{I}(q) = 0$.
**Part** $\Leftarrow$. Let $S$ be a maximum independent of $G^{\mathcal{P}}_{\phi}$ s.t. $|S| \geqslant n$ and $\mathcal{I}$ a Boolean interpretation of $\phi$ defined as:

$$\mathcal{I}(p) = \begin{cases} 1 & if\ p \in S \\ 0 & otherwise \end{cases}$$

Then, using the definition of $G^{\mathcal{P}}_{\phi}$, $\mathcal{I}$ satisfies all the binary negative clauses. Indeed, for all $\neg p \vee \neg q \in Neg(\phi)$, we have $p \notin S$ or $q \notin S$, since $\{p, q\}$ is an edge in $G^{\mathcal{P}}_{\phi}$. Furthermore, using a similar argument, $\mathcal{I}$ satisfies at most one literal in each positive clause. Thus, since $S \subseteq \mathcal{P}(\phi)$ and $|S| \geqslant n$, $\mathcal{I}$ satisfies all the positive clauses. □

Consider again the previous example of an IPBN formula $\phi$ (its corresponding graph $G^{\mathcal{P}}_{\phi}$ is described in Figure 2). Using Theorem 1, we know that $\phi$ is satisfiable since the number of positive clauses is equal to 2 and $\alpha(G^{\mathcal{P}}_{\phi}) = 2$. For instance, the maximum independent set $\{p, s\}$ induces the model $\{p, s, \neg q, \neg r, \neg t\}$ of $\phi$.

There are polynomial time algorithms for the problem of finding a maximum independent set for different classes of graphs, such us claw-free graphs, perfect graphs, graphs of bounded clique-width, etc (e.g. see [31, 22, 2, 6, 14]). Consequently, Theorem 1 shows that we can obtain tractable classes of IPBN-SAT from the maximum independent set problem. Consider, for instance, the class of claw-free graphs. A graph $G = (V, E)$ is *claw-free* if no vertex has three pairwise nonadjacent neighbours, i.e., for all $v \in V$, if there exist three distinct vertices $v_1, v_2, v_3 \in V$ s.t. $\{\{v, v_1\}, \{v, v_2\}, \{v, v_3\}\} \subseteq E$, then at least one of the edges $\{v_1, v_2\}$, $\{v_2, v_3\}$ and $\{v_1, v_3\}$ is in $E$. A sub-graph induced by four vertices where a vertex has three pairwise nonadjacent neighbours is called a *claw*. We say that an IPBN formula is claw-free if its corresponding graph $G^{\mathcal{P}}_{\phi}$ is claw-free.

**Theorem 2** (Claw-free IPBN-SAT)**.** *Checking the satisfiability of a claw-free IPBN formula is tractable.*

*Proof.* A direct consequence of Theorem 1 and the fact that the problem of finding a maximum independent set in the class of claw-free graphs is tractable. □

It is easy to see that checking whether an undirected graph is claw-free is tractable. Then, we obtain the following theorem:

**Theorem 3.** *Checking whether a CNF formula is claw-free is tractable.*

In the following proposition, we show that checking the satisfiability of the pigeon-hole principle belongs to Claw-free IPBN-SAT.

**Proposition 3.** *The pigeon-hole principle is an instance of claw-free IPBN-SAT.*

*Proof.* As we said before, the formula $\phi_{php_n^b}$ is clearly an IPBN formula. Let $p, q, r, s$ be four different vertices in its associated graph $(V, E)$ s.t. $\{\{p, q\}, \{p, r\}, \{p, s\}\} \subseteq E$. If two of the variables from $q$, $r$ and $s$ belongs to a same positive clause, then it is easy to see that the sub-graph induced by $p$, $q$, $r$ and $s$ is not a claw. We now consider the case where $q$, $r$ and $s$ belongs to three different positive clauses. Then, there are two variables from $q$, $r$ and $s$ representing a same hole for two different pigeons, since $\{\{p, q\}, \{p, r\}, \{p, s\}\} \subseteq E$. As a consequence, one of the following negative clauses belongs to $\phi_{php_n^b}$: $\neg q \vee \neg r$, $\neg q \vee \neg s$ and $\neg r \vee \neg s$. Thus, the sub-graph induced by $p$, $q$, $r$ and $s$ is not a claw. □

It is interesting to note that there are graph classes, other than claw-free graphs, characterizing tractable classes in the problem of finding a maximum independent set and which can be recognized in polynomial time. For instance, bipartite graphs and chordal graphs, which are perfect graphs, can be recognized in polynomial time.

# 5 Tractable Classes in SPBN-SAT

In this section, we extend the previous results by characterizing tractable classes in SPBN-SAT. Indeed, we show that each instance of SPBN-SAT can be translated into an instance of the problem of finding a maximum independent set. The basic idea consists in taking into account the number of occurrences of each positive literal, since positive clauses may share literals.

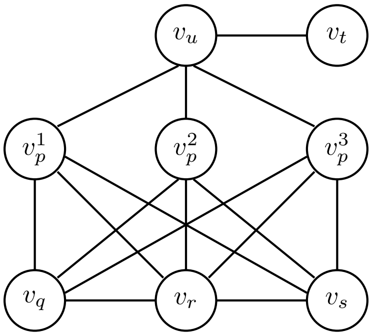

Figure 3: A graph associated to an SPBN formula

Given a PBN formula $\phi$ and $p \in \mathcal{P}(\phi)$, we use $Nb(p, \phi)$ to denote the value $|\{c \in Pos(\phi) \mid p \in c\}|$. Furthermore, we associate a set of $Nb(p, \phi)$ elements, denoted $S(p, \phi)$, to each propositional variable $p \in \mathcal{P}(\phi)$ so that, for all $p, q \in \mathcal{P}(\phi)$ with $p \neq q$, $S(p, \phi) \cap S(q, \phi) = \emptyset$.

Given a PBN formula $\phi$, we use $G^{\mathcal{O}}_{\phi}$ to denote the undirected graph $(V, E)$ where $V = \bigcup_{p \in \mathcal{P}(\phi)} S(p, \phi)$ and, for all $v, v' \in V$, $\{v, v'\} \in E$ if and only if there exist $p, q \in \mathcal{P}(\phi)$ s.t. $p \neq q$, $v \in S(p, \phi)$, $v' \in S(q, \phi)$ and $p$ and $q$ appears in a same clause.

For instance, the graph described in Figure 3 corresponds to the SBPN formula given in Example 1. Let us explain further how such a graph is obtained. The set of vertices $V$ is obtained by considering the union of $S(x, \phi)$ for all $x \in \mathcal{P}(\phi)$. We have for example $S(p, \phi) = \{v^1_p, v^2_p, v^3_p\}$, the other variables occurs only once in the positive clauses (e.g. $S(q, \phi) = \{v_q\}$ ). For the edges, we have for instance $p$ and $q$ occuring in the same clause, consequently, edges are added between all occurrences of $p$ ($\{v^1_p, v^2_p, v^3_p\}$) and $q$ ($\{v_q\}$).

**Proposition 4.** *Let $\phi$ be a PBN formula, $G^{\mathcal{O}}_{\phi} = (V, E)$ its associated graph, $p \in \mathcal{P}(\phi)$ and $v \in S(p, \phi)$. If $M$ is a maximum independent set s.t. $v \in M$, then $S(p, \phi) \subseteq M$.*

*Proof.* Let us assume that there exists $v' \in S(p, \phi)$ s.t. $v \neq v'$ and $v' \notin M$. Then, there exists $v'' \in M$ s.t. $\{v', v''\} \in E$ since $M$ is a maximum independent set. Thus, by definition, $\{v, v''\} \in E$ holds and we get a contradiction. □

Theorem 4 is a generalization of Theorem 1. Intuitively, using Proposition 2, we know that we only need to satisfy exactly one literal in each positive clause to obtain a model of an SPBN formula. Combining this property with Proposition 4, we get a definition of each SPBN-SAT instance as an instance of maximum independent set problem.

**Theorem 4.** *Given an SPBN formula $\phi$, $\phi$ is satisfiable iff $\alpha(G^{\mathcal{O}}_{\phi}) \geqslant n$ where $n = |Pos(\phi)|$.*

*Proof.* The proof is similar to that of Theorem 1. The main difference is in taking into account the occurrences of the positive literals since positive clauses may share literals in SPBN formulas.
**Part $\Rightarrow$.** Using Proposition 2, we know that $\phi$ is satisfiable iff $\mathcal{R}(\phi)$ is satisfiable. Let $\mathcal{I}$ be a Boolean interpretation s.t. $\mathcal{I}(\mathcal{R}(\phi)) = 1$ and $M = \{v \in S(p, \phi) \mid p \in \mathcal{P}(\phi)\ and\ \mathcal{I}(p) = 1\}$. Since $\mathcal{I}$ satisfies all the positive clauses, $|M| \geqslant n$ holds. Let us assume that there exists $v, v' \in M$ s.t. $\{v, v'\}$ is an edge in $G^{\mathcal{O}}_{\phi}$. Then, there exists a clause containing $\neg p \vee \neg q$ in $\mathcal{R}(\phi)$ s.t. $v \in S(p, \phi)$ and $v' \in S(q, \phi)$. Thus, we get a contradiction since $\mathcal{I}(p) = 0$ or $\mathcal{I}(q) = 0$.
**Part** $\Leftarrow$. Let $M$ be a maximum independent set of $G^{\mathcal{O}}_{\phi}$ s.t. $|M| \geqslant n$. We define the Boolean interpretation $\mathcal{I}$ of $\phi$ as:

$$\mathcal{I}(p) = \begin{cases} 1 & if\ S(p, \phi) \subseteq M \\ 0 & otherwise \end{cases}$$

Then, using the definition of $G^{\mathcal{O}}_{\phi}$, $\mathcal{I}$ satisfies all the negative binary clauses in $\phi$ since we have, for all $\neg p \vee \neg q \in Neg(\phi)$, $S(p, \phi) \cap M = \emptyset$ or $S(q, \phi) \cap M = \emptyset$. Furthermore, we have $M \subseteq \bigcup_{p \in \mathcal{P}(\phi)} S(p, \phi)$ and, using Proposition 4, we have either $S(p, \phi) \subseteq M$ or $S(p, \phi) \cap M = \emptyset$ for every $p \in \mathcal{P}(\phi)$. Moreover, using the definition of $G^{\mathcal{O}}_{\phi}$, for all $c \in Pos(\phi)$ and for all $p, q \in c$ with $p \neq q$, $S(p, \phi) \cap M = \emptyset$ or $S(q, \phi) \cap M = \emptyset$. Thus, $\mathcal{I}$ satisfies at most one literal in each positive clause. Since $|M| \geqslant n$, we deduce that $\mathcal{I}$ satisfies all the positive clauses in $\phi$. □

Let us consider the SPBN formula $\phi$ described in Example 1. The graph $\mathcal{G}^{\mathcal{O}}_{\phi}$ is described in Figure 3. The set of vertices $S = \{v^1_p, v^2_p, v^3_p, v_t\}$ is a maximum independent set of $\mathcal{G}^{\mathcal{O}}_{\phi}$. Using Theorem 4, we know that $\phi$ is satisfiable since it contains 4 positive clauses and $\alpha(\mathcal{G}^{\mathcal{O}}_{\phi}) = 4$. For example, the model of $\phi$ induced by $S$ is $\{p, t, \neg q, \neg r, \neg s, \neg u\}$.

As in Theorem 1, Theorem 4 allows to define tractable classes in SPBN-SAT using the tractable classes in the problem of finding a maximum independent set.

## 6 Clause-Covering based Generalization

In this section we describe a generalization of our previous results. To this end, we define a particular class of PBN formulæ, called Clause-Covering PBN (in short CC-PBN) formulæ, generalizing that of SPBN formulæ. This generalization allows the positive clauses in PBN formulæ to share more than one variable. In the same way, we establish a connection between the satisfiability of CC-PBN formulæ and the problem of finding a maximum independent set.

Let $\psi$ be a set of positive clauses and $p$ a propositional variable. We use $\mathcal{C}(p, \psi)$ to denote the set of clauses in $\psi$ where $p$ appears, i.e., $\mathcal{C}(p, \psi) = \{c \in S \mid p \in c\}$. We say that $\mathcal{C}(p, \psi)$ is the *cover* of $p$ in $\psi$.

**Definition 4** (CC-PBN formula)**.** *A* CC-PBN formula $\phi$ *is a PBN formula where, for all* $p, q \in \mathcal{P}(\phi)$ *with* $p, q \in c$ *for a clause* $c \in Pos(\phi)$*, we have* $\mathcal{C}(p, Pos(\phi)) \subseteq \mathcal{C}(q, Pos(\phi))$ *or* $\mathcal{C}(q, Pos(\phi)) \subseteq \mathcal{C}(p, Pos(\phi))$*.*

A CC-PBN formula is a PBN formula where for each two distinct literals in a positive clause, the cover of one is included in the cover of the other in the positive part. One can polynomially check if a PBN formula is in CC-PBN form.

It is easy to see that the SPBN formulæ are CC-PBN formulæ. Indeed, if $\phi$ is an SPBN formula, then for all $c \in Pos(\phi)$, there is at most one positive literal $p$ with $\{c\}$ is a proper subset of $\mathcal{C}(p, Pos(\phi))$, i.e., $\mathcal{C}(p, Pos(\phi)) \supset \{c\}$.

**Example 2.** *The following formula* $\phi$ *is in CC-PBN form:*
$(p_1 \vee p_2 \vee p_3 \vee p_4)\wedge$
$(p_1 \vee p_2 \vee p_3 \vee p_5)\wedge$
$(p_1 \vee p_2 \vee p_6 \vee p_7)\wedge$
$(p_1 \vee p_8 \vee p_9 \vee p_{10})\wedge$
$(\neg p_1 \wedge \neg p_4)\wedge$
$(\neg p_3 \vee \neg p_7)\wedge$
$(\neg p_5 \vee \neg p_6)\wedge$
$(\neg p_2 \vee \neg p_{10})$.

*Let us note that* $\mathcal{C}(p_3, Pos(\phi)) \subset \mathcal{C}(p_2, Pos(\phi)) \subset \mathcal{C}(p_1, Pos(\phi))$ *and the other variables appear once in* $Pos(\phi)$*.*

We now show that we only need to satisfy exactly one literal in each positive clause to obtain a model of a CC-PBN formula.

**Proposition 5.** *Given an CC-PBN formula* $\phi$*,* $\phi$ *is satisfiable iff* $\mathcal{R}(\phi)$ *is satisfiable.*

*Proof.* It suffices to show that if $\phi$ is satisfiable then $\mathcal{R}(\phi)$ is satisfiable, the converse being trivial. Let $\mathcal{I}$ be a model of $\phi$ and $\preceq$ a preorder relation on $\mathcal{P}(\phi)$ satisfying the following property: for all $p, q \in \phi$ with $p \neq q$, if $\mathcal{C}(q, Pos(\phi))$ is a subset of $\mathcal{C}(p, Pos(\phi))$ ($\mathcal{C}(q, Pos(\phi)) \subseteq \mathcal{C}(p, Pos(\phi))$), then $p \preceq q$. Given a positive clause $c$ in $Pos(\phi)$, we use $l(c, \mathcal{I})$ to denote one of the smallest propositional variable in $c$ w.r.t. $\preceq$ s.t. $\mathcal{I}(l(\mathcal{I}, c)) = 1$. Let $\mathcal{I}'$ be a Boolean interpretation of $\mathcal{R}(\phi)$ defined as follows:

$$\mathcal{I}'(p) = \begin{cases} 1 & if\ \exists c \in Pos(\phi), p = l(c, \mathcal{I}) \\ 0 & otherwise \end{cases}$$

Using the fact that, for all $p, q \in \mathcal{P}(\phi)$ with $p, q \in c$ for a clause $c \in Pos(\phi)$, we have $\mathcal{C}(p, Pos(\phi)) \subseteq \mathcal{C}(q, Pos(\phi))$ or $\mathcal{C}(q, Pos(\phi)) \subseteq \mathcal{C}(p, Pos(\phi))$, $\mathcal{I}$ satisfies exactly one literal in each positive clause. As a consequence, $\mathcal{I}'$ is a model of $\mathcal{R}(\phi)$. □

We now provide a generalization of Theorem 4. It is obtained by combining Proposition 4 and Proposition 5.

**Theorem 5.** *Given a CC-PBN formula $\phi$, $\phi$ is satisfiable iff $\alpha(G^{\mathcal{O}}_{\phi}) \geqslant n$ where $n = |Pos(\phi)|$.*

*Proof.* The proof is similar to that of Theorem 4.
**Part** $\Rightarrow$. The proof is obtained by replacing Proposition 2 with Proposition 5 and using the same arguments.
**Part** $\Leftarrow$, the same arguments works and the following Boolean interpretation built from an independent set $M$ of $G^{\mathcal{O}}_{\phi}$ is a model of $\phi$ if its size is greater than or equal to $n$:

$$\mathcal{I}(p) = \begin{cases} 1 & if\ S(p, \phi) \subseteq M \\ 0 & otherwise \end{cases}$$

□

For all the fragments of the hierarchy, IPBN, SPBN and CC-PBN, Theorem 5, gives us a simple way to find the SAT counterparts of the different tractable classes of the maximum independent set problem. Our approach can be seen as a new way to deal with tractability in SAT by a carful analysis of the positive parts of PBN formulæ.

## 7 A Connection with Minimal Models

The minimal model problem has important applications in AI, such as propositional circumscription and minimal diagnosis [30, 35]. We here provide a characterization of the minimal models in the case of CC-PBN formulæ using the maximum independent set problem.

Let $\mathcal{I}$ and $\mathcal{I}'$ be two interpretations of a propositional formula $\phi$. Then, $I$ is said to be smaller than $I'$, written $\mathcal{I} \preceq \mathcal{I}'$, if $\{p \in \mathcal{P}(\phi) \mid \mathcal{I}(p) = 1\} \subseteq \{p \in \mathcal{P}(\phi) \mid \mathcal{I}'(p) = 1\}$.

**Definition 5.** *Let $\phi$ be a propositional formula and $\mathcal{I}$ a model of $\phi$. Then $\mathcal{I}$ is said to be a minimal model of $\phi$ if there is no model strictly smaller than $I$ w.r.t. $\preceq$.*

**Proposition 6.** *Let $\phi$ be a CC-PBN formula and $\mathcal{I}$ a model of $\phi$. Then, $\mathcal{I}$ is a minimal model of $\phi$ iff $\mathcal{I}$ satisfies exactly one literal in each positive clause.*

*Proof.* **Part** $\Leftarrow$. We assume that there exists a model $\mathcal{I}$' of $\phi$ s.t. $\{p \in \mathcal{P}(\phi) \mid \mathcal{I}'(p) = 1\} \subset \{p \in \mathcal{P}(\phi) \mid \mathcal{I}(p) = 1\}$. Then, we get a contradiction since there exists at least one clause in $Pos(\phi)$ which is not satisfied. Indeed, since $\mathcal{I}$ satisfies exactly one literal in each positive clause, the positive clauses containing the literals in $\{p \in \mathcal{P}(\phi) \mid \mathcal{I}(p) = 1\} \setminus \{p \in \mathcal{P}(\phi) \mid \mathcal{I}'(p) = 1\}$ are not satisfied by $\mathcal{I}'$. We recall that we consider PBN formulæ without pure literals.
**Part** $\Rightarrow$. We assume that there exists a positive clause in $Pos(\phi)$ containing two literals $p$ and $q$

s.t. $\mathcal{I}(p) = \mathcal{I}(q) = 1$. Since $\phi$ is a CC-PBN formula, we know that $\mathcal{C}(p, Pos(\phi)) \subseteq \mathcal{C}(q, Pos(\phi))$ or $\mathcal{C}(q, Pos(\phi)) \subseteq \mathcal{C}(p, Pos(\phi))$. We here consider w.l.o.g. that $\mathcal{C}(p, Pos(\phi)) \subseteq \mathcal{C}(q, Pos(\phi))$. As a consequence, the following interpretation $\mathcal{I}'$ is a model of $\phi$: $\mathcal{I}'(p) = 0$ and $\mathcal{I}'(r) = \mathcal{I}(r)$ for every propositional $r \neq p$. Thus, $\mathcal{I}$ is not a minimal model of $\phi$ and we get a contradiction. □

**Theorem 6.** *Let $\phi$ be a CC-PBN formula and $\mathcal{I}$ a Boolean interpretation of $\phi$. $\mathcal{I}$ is a minimal model of $\phi$ iff $\alpha(G^{\mathcal{O}}_{\phi}) = |Pos(\phi)|$ and $\bigcup_{p \in E} S(p, \phi)$ is a maximum independent set of $G^{\mathcal{O}}_{\phi}$ where $E = \{p \in \mathcal{P}(\phi) \mid \mathcal{I}(p) = 1\}$.*

*Proof.* **Part** $\Leftarrow$. Using Theorem 5, we know that $\phi$ is satisfiable since $\alpha(G^{\mathcal{O}}_{\phi}) = |Pos(\phi)|$. We know also that $\mathcal{I}$ is a model of $\phi$ (see the proof of Theorem 5). Using Proposition 6, $\mathcal{I}$ is a minimal model of $\phi$ since $\mathcal{I}$ satisfies exactly one literal in each positive clause ($\alpha(G^{\mathcal{O}}_{\phi}) = |Pos(\phi)|$).
**Part** $\Rightarrow$. Using Proposition 6, $\mathcal{I}$ satisfies exactly one literal in each positive clause. As a consequence, $|\bigcup_{p \in E} S(p, \phi)| = |Pos(\phi)|$ and $\bigcup_{p \in E} S(p, \phi)$ is an independent set of $G^{\mathcal{O}}_{\phi} = (V, E)$. Using the definition of $G^{\mathcal{O}}_{\phi}$, we know that every node in $V$ is either in $\bigcup_{p \in E} S(p, \phi)$ or adjacent to a node in $\bigcup_{p \in E} S(p, \phi)$. Thus, the latter set is a maximum independent set of $G^{\mathcal{O}}_{\phi}$. □

# 8 Related Works

Several works have investigated resolution, inference and tractability relationships between SAT and CSP through several encodings (e.g. [39, 33]). The relationship between our framework and binary CSP concerns only the particular case of IPBN form. Indeed, this particular case concerns PBN formulæ where the positive clauses do not share variables (SAT direct encoding of CSP). Our framework extends this, by introducing nested forms obtained by considering some properties on the occurrences of variables in the positive part of a PBN formulæ. This suggests that tractability results can be derived by a careful analysis of the positive part of the PBN formula. This is the most important point of the paper. Our framework presents tractability results, but most importantly propose a new way to deal with tractability in SAT.

It is well known that solving a binary CSP instance $P$ is equivalent to finding an independent set in the microstructure complement of $P$ [34]. It is also well known that a binary CSP instance can be encoded as an IPBN-SAT instance using direct encodings and vice-versa [39]. Therefore, tractable classes on the maximum independent set problem for various classes of graphs can be easily used to characterize tractable CSP classes [8, 13]. Consequently, tractable classes in IPBN-SAT can be derived in the same way.

Let us recall that the microstructure of a binary CSP instance is the graph $(V, E)$ where $V$ is the set of possible assignments of values to variables and $E$ connect all pairs of allowed assignments. The microstructure complement is the complement of this graph. It is worth noticing that the microstructure graph representation plays an important role in characterizing tractable classes in CSPs by imposing conditions in the microstructure (e.g. [36, 9]).

More generally, several results are obtained thanks to polynomial reductions between different NP-Complete or NP-Hard problems (e.g. [25]). Let us mention the work by Jégou and Paris [24], where a SAT instance is represented as an undirected graph issued from a polynomial transformation of SAT to CLIQUE problem. By exploiting the well known properties of chordal graphs, the authors characterize a new SAT tractable class. A well known tractable class, called matched CNF formulas [19], is characterized through graph theoretic notions. A CNF formula is matched if its associated bipartite graph (whose vertices are clauses and variables, and edges connect variables to clauses where they appear) has a matching that covers all clauses. Other polynomial reductions from a problem $\pi$ to a problem $\pi'$ are also designed to benefit from $\pi'$ efficient solving techniques. Let us mention for example the recent reduction of the maximal independent set to minimum satisfiability problem proposed by Ignatiev et al. [23].

## 9 Conclusion and perspectives

In this paper, we proposed a new framework for characterizing tractable classes in SAT problem. Without loss of generality, we focus on a target form of Boolean formulæ with only positive clauses and negative binary clauses. We defined a hierarchy of fragments characterized by some specific constraints on the positive occurrences of the variables. For each fragment and for each of its propositional formula, we provide a reduction to a graph, where the satisfiability of this formula is equivalent to the existence of a maximum independent set of size greater than or equal to the number of positive clauses. This property comes from the fact that in the considered form we only need to satisfy one literal in each positive clause. This original framework allows us to characterize new tractable classes from existing polynomial time algorithms of the problem of finding a maximum independent set in the case of different graph classes, such as claw-free graphs and perfect graphs. Furthermore, we provide a characterization of the minimal models based on the maximum independent set problem.

Our framework can be used for a broad cross-fertilization between propositional satisfiability, graphs theory and constraint satisfaction problems. Determining other properties on the positive part of PBN formulae that might lead to new tractable classes is clearly an interesting research avenue.

## References

[1] Krishnamurthy Balachander. Shorts proofs for tricky formulas. *Acta Informatica*, 22:253–275, 1985.

[2] Egon Balas and Chang Sung Yu. On graphs with polynomially solvable maximum-weight clique problem. *Networks*, 19(2):247–253, 1989.

[3] Belaid Benhamou and Lakhdar Sais. Theoretical study of symmetries in propositional calculus and applications. In *11th International Conference on Automated Deduction (CADE'1992)*, volume 607 of *Lecture Notes in Computer Science*, pages 281–294. Springer, 1992.

[4] Alex Borgida and David W. Etherington. Hierarchical knowledge bases and efficient disjunctive reasoning. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 33–43, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[5] Endre Boros, Peter L. Hammer, and Xiaorong Sun. Recognition of q-Horn formulae in linear time. *Annals of Mathematics and Artificial Intelligence*, 1:21–32, 1990.

[6] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph Classes: A Survey.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[7] Marco Cadoli. Panel on "knowledge compilation and approximation": Terminology, questions, references. In *In Proc. of the Fourth Int. Symp. on Artificial Intelligence and Mathematics, AI/Math-96*, pages 183–186, 1996.

[8] David A. Cohen. A new classs of binary csps for which arc-constistency is a decision procedure. In *Principles and Practice of Constraint Programming - CP 2003, 9th International Conference, CP 2003, Kinsale, Ireland, September 29 - October 3, 2003, Proceedings*, pages 807–811, 2003.

[9] David A. Cohen, Martin C. Cooper, Páidí Creed, Dániel Marx, and András Z. Salamon. The tractability of CSP classes defined by forbidden patterns. *J. Artif. Intell. Res. (JAIR)*, 45:47–78, 2012.

[10] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[11] Stephen A. Cook. A short proof of the pigeon hole principle using extended resolution. *SIGACT News*, 8(4):28–32, October 1976.

[12] Martin Cooper, Achref El Mouelhi, Cyril Terrioux, and Bruno Zanuttini. On Broken Triangles (regular paper). In *International Conference on Principles and Practice of Constraint Programming (CP'2014)*, pages 9–24, 2014.

[13] Martin C. Cooper and Stanislav Zivny. A new hybrid tractable class of soft constraint problems. In *Principles and Practice of Constraint Programming - CP 2010 - 16th International Conference, CP 2010, St. Andrews, Scotland, UK, September 6-10, 2010. Proceedings*, pages 152–166, 2010.

[14] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.*, 33(2):125–150, 2000.

[15] Peter Damaschke, Haiko Müller, and Dieter Kratsch. Domination in convex and chordal bipartite graphs. *Information Processing Letter*, 36(5):231–236, December 1990.

[16] Johan de Kleer, Alan K. Mackworth, and Raymond Reiter. Characterizing diagnoses and systems. *Artif. Intell.*, 56(2-3):197–222, 1992.

[17] Wiliam F. Dowling and Jean H. Gallier. Linear-time algorithms for testing satisfiability of propositional Horn formulae. *Journal of Logic Programming*, pages 267–284, 1984.

[18] Ralph Faudree, Evelyne Flandrin, and Zdenk Ryjek. Claw-free graphs  a survey. *Discrete Mathematics*, 164(13):87 – 147, 1997.

[19] John Franco and Allen Van Gelder. A perspective on certain polynomial-time solvable classes of satisfiability. *Discrete Applied Mathematics*, 125(23):177 – 214, 2003.

[20] Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3):365–385, 1991.

[21] Martin Charles Golumbic, editor. *Algorithmic Graph Theory and Perfect Graphs*. Annals of Discrete Mathematics, 2004.

[22] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[23] Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. On reducing maximum independent set to minimum satisfiability. In *17th International Conference on Theory and Applications of Satisfiability Testing (SAT'2014)*, pages 103–120, 2014.

[24] Philippe Jégou and Lionel Paris. A new formula rewriting by reasoning on a graphical representation of SAT instances. In *Eighth Symposium on Abstraction, Reformulation, and Approximation, SARA 2009, Lake Arrowhead, California, USA, 8-10 August 2009*, 2009.

[25] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.

[26] Melven R. Krom. The decision problem for a class of first-order formulas in which all disjunctions are binary. *Mathematical Logic Quarterly*, 13(1-2):15–20, 1967.

[27] Harry R. Lewis. Renaming a set of clauses as a horn set. *J. ACM*, 25(1):134–135, January 1978.

[28] Vladimir Lifschitz. Computing circumscription. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence (IJCAI'85)*, pages 121–127, 1985.

[29] Vadim V. Lozin and Martin Milani. A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *Journal of Discrete Algorithms*, 6(4):595 – 604, 2008.

[30] John McCarthy. Circumscription - a form of non-monotonic reasoning. *Artif. Intell.*, 13(1-2):27–39, 1980.

[31] George J. Minty. On maximal independent sets of vertices in claw-free graphs. *J. Comb. Theory,*

*Ser. B*, pages 284–304, 1980.

[32] Sebastian Ordyniak, Daniel Paulusma, and Stefan Szeider. Satisfiability of acyclic and almost acyclic {CNF} formulas. *Theoretical Computer Science*, 481(0):85 – 99, 2013.

[33] Justyna Petke and Peter Jeavons. The order encoding: From tractable CSP to tractable SAT. In *Theory and Applications of Satisfiability Testing - SAT 2011 - 14th International Conference, SAT 2011, Ann Arbor, MI, USA, June 19-22, 2011. Proceedings*, pages 371–372, 2011.

[34] Jégou Philippe. Decomposition of domains based on the micro-structure of finite constraint-satisfaction problems. In *Proceedings of the 11th National Conference on Artificial Intelligence. Washington, DC, USA, July 11-15, 1993.*, pages 731–736, 1993.

[35] Raymond Reiter. A theory of diagnosis from first principles. *Artif. Intell.*, 32(1):57–95, 1987.

[36] András Z. Salamon and Peter G. Jeavons. Perfect constraints are tractable. In *Principles and Practice of Constraint Programming, 14th International Conference, CP 2008, Sydney, Australia, September 14-18, 2008. Proceedings*, pages 524–528, 2008.

[37] Bart Selman and Henry Kautz. Knowledge compilation using horn approximations. In *Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, pages 904–909. MIT Press, 1991.

[38] Juraj Stacho. 3-colouring at-free graphs in polynomial time. In *Algorithms and Computation*, volume 6507 of *Lecture Notes in Computer Science*, pages 144–155. Springer Berlin Heidelberg, 2010.

[39] Walsh Toby. SAT vs. CSP. In *Principles and Practice of Constraint Programming - CP 2000, 6th International Conference, Singapore, September 18-21, 2000, Proceedings*, pages 441–456, 2000.