Generalising and unifying SLUR and unit-refutation completeness

Matthew Gwynne[1] and Oliver Kullmann[2]

[1] Computer Science Department, Swansea University
http://cs.swan.ac.uk/~csmg/
[2] Computer Science Department, Swansea University
http://cs.swan.ac.uk/~csoliver

Abstract. The class $\mathcal{SLUR}$ (*Single Lookahead Unit Resolution*) was introduced in [22] as an umbrella class for efficient SAT solving. [7,2] extended this class in various ways to hierarchies covering all of CNF (all clause-sets). We introduce a hierarchy $\mathcal{SLUR}_k$ which we argue is the natural "limit" of such approaches.
The second source for our investigations is the class $\mathcal{UC}$ of *unit-refutation complete clause-sets* introduced in [10]. Via the theory of (tree-resolution based) "hardness" of clause-sets as developed in [19,20,1] we obtain a natural generalisation $\mathcal{UC}_k$, containing those clause-sets which are "unit-refutation complete of level k", which is the same as having hardness at most k. Utilising the strong connections to (tree-)resolution complexity and (nested) input resolution, we develop fundamental methods for the determination of hardness (the level k in $\mathcal{UC}_k$).
A fundamental insight now is that $\mathcal{SLUR}_k = \mathcal{UC}_k$ holds for all k. We can thus exploit both streams of intuitions and methods for the investigations of these hierarchies. As an application we can easily show that the hierarchies from [7,2] are strongly subsumed by $\mathcal{SLUR}_k$. We conclude with a discussion of open problems and future directions.

1 Introduction

The boolean satisfiability problem, SAT for short, in its core version is the problem of deciding satisfiability of a conjunctive normal form (clause-set); see the handbook [4] for further information. We bring together two previously unconnected streams of research:

SLUR The search for classes of clause-sets for which one can decide satisfiability in polynomial time. Especially we consider the SLUR algorithm and class.

UC The search for target classes of clause-sets with good knowledge compilation properties, i.e., where the clausal entailment problem can be decided quickly. Especially we consider the class UC of unit-refutation complete clause-sets.

In the year 1995 in [22] the SLUR algorithm was introduced, a simple *incomplete* non-deterministic SAT-decision algorithm, together with the class $\mathcal{SLUR}$ of inputs where it *always* succeeds. $\mathcal{SLUR}$ contains various classes with polynomial-time SAT decision, where previously only rather complicated algorithms were known. The natural question arises, whether $\mathcal{SLUR}$ can be turned into a hierarchy, covering in the limit all clause-sets. In [7,2] the authors finally proved

that membership decision of $\mathcal{SLUR}$ is coNP-complete, and presented three hierarchies, $\mathcal{SLUR}(k), \mathcal{SLUR}*(k)$ and CANON(k). It still seemed that none of these hierarchies is the final answer, though they all introduce a certain natural intuition. We now present what seems the natural "limit hierarchy", which we call $\mathcal{SLUR}_k$, and which unifies the two basic intuitions embodied in $\mathcal{SLUR}(k), \mathcal{SLUR}*(k)$ on the one hand and CANON(k) on the other hand.

In the year 1994 in [10] the class $\mathcal{UC}$ was introduced, containing clause-sets F such that *clause entailment*, that is, whether $F \models C$ holds (clause C follows logically from F, i.e., C is an implicate of F), can be decided by unit-clause propagation. A second development is important here, namely the development of the notion of "hardness" $\mathrm{hd}(F)$ in [19,20,1]. As we show in Theorem 14, $\mathrm{hd}(F) \le k$ is equivalent to the property of F, that all implicates of F (i.e., all clauses C with $F \models C$) can be derived by k-times nested input resolution from F, a generalisation of input resolution as introduced and studied in [19,20]. We obtain that $\mathcal{UC}$ is precisely the class of clause-sets F with $\mathrm{hd}(F) \le 1$! It is then natural to define the hierarchy $\mathcal{UC}_k$ via the property $\mathrm{hd}(F) \le k$. The hierarchy CANON(k) is based on resolution trees of height at most k, which is a special case of k-times nested input resolution, and so we have CANON$(k) \subset \mathcal{UC}_k$.

The hardness-notion provides the *proof-theoretic side* of our investigations. The *algorithmic side* is given by the reductions $\mathrm{r}_k(F)$, which perform certain forced assignments, with r_1 being UCP (unit-clause propagation) as the most well-known case. For unsatisfiable F the hardness $\mathrm{hd}(F)$ is equal to the minimal k such that $\mathrm{r}_k(F)$ detects unsatisfiability of F. This yields the basic observation $\mathcal{UC} \subseteq \mathcal{SLUR}$ — and actually we have $\mathcal{UC} = \mathcal{SLUR}$!

So by replacing the use of r_1 in the SLUR algorithm by r_k (using a refined, semantic, analysis) we obtain a natural hierarchy $\mathcal{SLUR}_k$, which includes the previous SLUR-hierarchies $\mathcal{SLUR}(k)$ and $\mathcal{SLUR}*(k)$, and where we have $\mathcal{SLUR}_k = \mathcal{UC}_k$. This equality of these two hierarchies is our argument that we have found the "limit hierarchy" for SLUR.

The underlying report of this paper is [15], where all missing proofs can be found, and where examples are provided. Also the anticipated main application of the classes $\mathcal{UC}_k$ as target classes for SAT translation is discussed there.

2 Preliminaries

We follow the general notions and notations as outlined in [18]. Based on an infinite set $\mathcal{VA}$ of variables, we form the set $\mathcal{LIT} := \mathcal{VA} \dot\cup \overline{\mathcal{VA}}$ of positive and negative literals, using complementation. A clause $C \subset \mathcal{LIT}$ is a finite set of literals without clashes, i.e., $C \cap \overline{C} = \emptyset$, where for $L \subseteq \mathcal{LIT}$ we set $\overline{L} := \{\overline{x} : x \in L\}$. The set of all clauses is denoted by $\mathcal{CL}$. A clause-set $F \subset \mathcal{CL}$ is a finite set of clauses, the set of all clause-sets is $\mathcal{CLS}$. A special clause is the empty clause $\bot := \emptyset \in \mathcal{CL}$, and a special clause-set is the empty clause-set $\top := \emptyset \in \mathcal{CLS}$. By $\mathrm{lit}(F) := \bigcup F \cup \overline{\bigcup F}$ we denote the set of literals occurring at least in one polarity in F. We use $\mathrm{var} : \mathcal{LIT} \to \mathcal{VA}$ for the underlying variable of a literal, $\mathrm{var}(C) := \{\mathrm{var}(x) : x \in C\} \subset \mathcal{VA}$ for the variables in a clause, and $\mathrm{var}(F) := \bigcup_{C \in F} \mathrm{var}(C)$ for the variables in a clause-set. The number of variables in a clause-set is $n(F) := |\mathrm{var}(F)| \in \mathbb{N}_0$, the number of clauses is $c(F) := |F| \in \mathbb{N}_0$, and the number of literal occurrences is $\ell(F) := \sum_{C \in F} |C| \in \mathbb{N}_0$. The set of Horn clause-sets is $\mathcal{HO} \subset \mathcal{CLS}$, where every clause contains at most one positive

literal. A partial assignment $\varphi : V \to \{0,1\}$ maps $V \subset \mathcal{VA}$ to truth-values, the set of all partial assignments is $\mathcal{PASS}$. We construct partial assignments via $\langle v_1 \to \varepsilon_1, \ldots, v_n \to \varepsilon_n \rangle \in \mathcal{PASS}$ for $v_i \in \mathcal{VA}$ and $\varepsilon_i \in \{0,1\}$. We use $\mathrm{var}(\varphi) := V$. For a partial assignment $\varphi \in \mathcal{PASS}$ and a clause-set $F \in \mathcal{CLS}$ the application of φ to F is denoted by $\varphi * F \in \mathcal{CLS}$, which results from F by removing all satisfied clauses (containing at least one satisfied literal), and removing all falsified literals from the remaining clauses. A clause-set F is satisfiable (i.e., $F \in \mathcal{SAT} \subset \mathcal{CLS}$) if there exists a partial assignment φ with $\varphi * F = \top$, otherwise F is unsatisfiable (i.e., $F \in \mathcal{USAT} := \mathcal{CLS} \setminus \mathcal{SAT}$). Two clauses $C, D \in \mathcal{CL}$ are resolvable if they clash in exactly one literal x, that is, $C \cap \overline{D} = \{x\}$, in which case their resolvent is $(C \cup D) \setminus \{x, \overline{x}\}$ (with resolution literal x). A resolution tree is a binary tree formed by the resolution operation. We write $\boldsymbol{T : F \vdash C}$ if T is a resolution tree with axioms (the clauses at the leaves) all in F and with derived clause (at the root) C. By $\mathbf{Comp^*_R(F)}$ for unsatisfiable F the minimum number of leaves in a tree-resolution-refutation $T : F \vdash \bot$ is denoted. Finally, by $\mathrm{r}_1 : \mathcal{CLS} \to \mathcal{CLS}$ unit-clause propagation is denoted, that is applying $F \rightsquigarrow \langle x \to 1 \rangle * F$ as long as there are unit-clauses $\{x\} \in F$, and reducing $F \rightsquigarrow \{\bot\}$ in case of $\bot \in F$.

3 The SLUR class and extensions

The SLUR-algorithm and the class $\mathcal{SLUR} \subset \mathcal{CLS}$ have been introduced in [22]. For input $F \in \mathcal{CLS}$ we get an incomplete polynomial-time SAT algorithm, which either returns "SAT" or "UNSAT" (in both cases correctly) or gives up. This algorithm is non-deterministic, and $\mathcal{SLUR}$ is the class of clause-sets where it never gives up (whatever the choices are). Thus SAT-decision for $F \in \mathcal{SLUR}$ can be done in polynomial time, and due to an observation attributed to Truemper in [11], the SLUR-algorithm can actually be implemented such that it runs in linear time. Decision of membership, that is whether $F \in \mathcal{SLUR}$ holds, by definition is in coNP, but only in [7] it was finally shown that this decision problem is coNP-complete. The original motivation was that $\mathcal{SLUR}$ contains several other classes, including renamable Horn, extended Horn, hidden extended Horn, simple extended Horn and CC-balanced clause-sets, where for each class it was known that the SAT problem is solvable in polynomial time, but with in some cases rather complicated proofs, while it is trivial to see that the SLUR-algorithm runs in polynomial time. In [11,12] probabilistic properties of $\mathcal{SLUR}$ have been investigated. In this section we first give a semantic definition of $\mathcal{SLUR}$ in Subsection 3.1. In a nutshell, $\mathcal{SLUR}$ is the class of clause-sets where either UCP (unit-clause propagation aka r_1) creates the empty clause, or where otherwise iteratively making assignments followed by UCP will always yield a satisfying assignment, given that these transitions do not obviously create unsatisfiable results, i.e., do not create the empty clause. In order to understand this definition clearly, we present a precise mathematical (non-algorithmic) definition, based on the transition relation $F \xrightarrow{\mathrm{SLUR}} F'$ (Definition 2), which represents one non-deterministic step of the SLUR algorithm: If r_1 on input $F \in \mathcal{CLS}$ does not determine unsatisfiability (in which case we have $F \in \mathcal{SLUR}$), then $F \in \mathcal{SLUR}$ iff $\top$ can be reached by this transition relation, while everything else reachable from F is not an end-point of this transition relation. In [7,2] recently three approaches towards generalising $\mathcal{SLUR}$ have been considered, and we discuss

them in Subsection 3.2. Our generalisation, called $\mathcal{SLUR}_k$, which we see as the natural completion of these approaches, will be presented in Section 6.

3.1 SLUR

The idea of the SLUR-algorithm ("Single Lookahead Unit Resolution") for input $F \in \mathcal{CLS}$ is as follows: First run UCP, that is, reduce $F \rightsquigarrow \mathrm{r}_1(F)$. If now $\bot \in F$ then we determined unsatisfiable. If not, then the algorithm guesses a satisfying assignment for F, by repeated transitions $F \xrightarrow{\text{SLUR}} F'$, where F' is obtained by assigning one variable and then performing UCP. The "lookahead" means that for $F' = \{\bot\}$ this transition is not performed. The algorithm might find a satisfying assignment in this way, or it gets stuck, in which case it "gives up". The SLUR class is defined as the class of clause-sets where this algorithm never gives up. The precise details are as follows. First we define the underlying transition relation (one non-failing transition from F to F'):

Definition 1. *For clause-sets $F, F' \in \mathcal{CLS}$ the relation $\boldsymbol{F \xrightarrow{SLUR} F'}$ holds if there is $x \in \mathrm{lit}(F)$ such that $F' = \mathrm{r}_1(\langle x \to 1\rangle * F)$ and $F' \neq \{\bot\}$. The transitive-reflexive closure is denoted by $\boldsymbol{F \xrightarrow{SLUR}_* F'}$.*

Via the transition-relation $F \xrightarrow{\text{SLUR}} F'$ we can now easily define the class $\mathcal{SLUR}$, which will find a natural generalisation in Definition 26 to $\mathcal{SLUR}_k$ for $k \in \mathbb{N}_0$:

Definition 2. *The set of reduced clause-sets reachable from $F \in \mathcal{CLS}$ is denoted by $\boldsymbol{\mathrm{slur}(F)} := \{F' \in \mathcal{CLS} \mid F \xrightarrow{SLUR}_* F' \wedge \neg\exists\, F'' \in \mathcal{CLS} : F' \xrightarrow{SLUR} F''\}$. The class of all clause-sets which are either identified by UCP to be unsatisfiable, or where by SLUR-reduction always a satisfying assignment is found, is denoted by $\boldsymbol{\mathcal{SLUR}} := \{F \in \mathcal{CLS} : \mathrm{r}_1(F) \neq \{\bot\} \Rightarrow \mathrm{slur}(F) = \{\top\}\}$.*

3.2 Previous approaches for SLUR hierarchies

In [7,2] three hierarchies $\mathcal{SLUR}(k), \mathcal{SLUR}*(k)$ $(k \in \mathbb{N})$ and $\mathrm{CANON}(k)$ $(k \in \mathbb{N}_0)$ have been introduced. In Section 4 of [2] it is shown that $\mathcal{SLUR}(k) \subset \mathcal{SLUR}*(k)$ for all $k \in \mathbb{N}$ and so we restrict our attention to $\mathcal{SLUR}*(k)$ and $\mathrm{CANON}(k)$. $\mathrm{CANON}(k)$ is defined to be the set of clause-sets F such that every prime implicate of F can be derived from F by a resolution tree of height at most k. Note that basically by definition (using stability of resolution proofs under application of partial assignments) we get that each $\mathrm{CANON}(k)$ is stable under application of partial assignments and under variable-disjoint union. The $\mathcal{SLUR}*(k)$ hierarchy is derived in [2] from the $\mathcal{SLUR}$ class by extending the reduction r_1. We provide an alternative formalisation here, in the same manner as in Section 3.1. The main question is the transition relation $F \rightsquigarrow F'$. The $\mathcal{SLUR}*(k)$-hierarchy provides stronger and stronger witnesses that F' might be satisfiable, by longer and longer assignments (making "k decisions") not yielding the empty clause:

Definition 3. *That partial assignment $\varphi \in \mathcal{PASS}$ **makes** k **decisions** for some $k \in \mathbb{N}_0$ w.r.t. $F \in \mathcal{CLS}$ is defined recursively as follows: For $k = 0$ this relation holds if $\varphi * F = \mathrm{r}_1(F)$, while for $k > 0$ this relation holds if either there is $k' < k$ such that φ makes k' decision w.r.t. F and $\varphi * F = \top$, or there exists $x \in \mathrm{lit}(F)$*

and a partial assignment φ' making $k-1$ decision for $\mathrm{r}_1(\langle x \to 1\rangle * F)$, and where $\varphi * F = \varphi' * \mathrm{r}_1(\langle x \to 1\rangle * F)$. Now $\boldsymbol{F} \xrightarrow{SLUR*\boldsymbol{k}} \boldsymbol{F'}$ for $k \geq 1$ by definition holds if there is a partial assignment φ making k decision w.r.t. F with $F' = \varphi * F$, where $F' \neq \{\bot\}$. The reflexive-transitive closure is $\xrightarrow{SLUR*k}_*$.

Finally we can define the hierarchy:

$$\begin{aligned}\mathbf{slur*(k)(F)} &:= \{F' \in \mathcal{CLS} \mid F \xrightarrow{\mathrm{SLUR}*k}_* F' \wedge \neg\exists\, F'' : F' \xrightarrow{\mathrm{SLUR}*k} F''\}\\ \boldsymbol{\mathcal{SLUR}*(k)} &:= \{F \in \mathcal{CLS} : \mathrm{slur}*(k)(F) \neq \{F\} \Rightarrow \mathrm{slur}*(k)(F) = \{\top\}\}.\end{aligned}$$

The unsatisfiable elements of $\mathcal{SLUR}*(k)$ are those $F \neq \top$ with $\mathrm{slur}*(k)(F) = \{F\}$. By definition each $\mathcal{SLUR}*(k)$ is stable under application of partial assignments, but not stable under variable-disjoint union, since the number of decision variables is bounded by k (in Lemma 21 we will see that our hierarchy is stable under variable-disjoint union, which is natural since it strengthens the CANON(k)-hierarchy).

4 Generalised unit-clause propagation

In this section we review the approximations of forced assignments, as computed by the hierarchy of reductions $\mathrm{r}_k : \mathcal{CLS} \to \mathcal{CLS}$ from [19,20] for $k \in \mathbb{N}_0$. For further discussions of these reductions, in the context of SAT decision and in their relations to various consistency and width-related notions, see [19,20] and Section 3 in [21]. Fundamental is the notion of a **forced literal** of a clause-set, which are literals which must be set to true in order to satisfy the clause-set. If x is a forced literal for F, then the **forced assignment** $\langle x \to 1\rangle * F$ yields a satisfiability-equivalent clause-set. We denote by $\mathbf{r_\infty(F)} \in \mathcal{CLS}$ the result of applying all forced assignments to F. Note that F is unsatisfiable iff $\mathrm{r}_\infty(F) = \{\bot\}$. We now present the hierarchy $\mathrm{r}_k : \mathcal{CLS} \to \mathcal{CLS}$, $k \in \mathbb{N}_0$, of reductions ([19]), which achieves approximating r_∞ by poly-time computable functions.

Definition 4 ([19]). *The maps $\mathbf{r_k} : \mathcal{CLS} \to \mathcal{CLS}$ for $k \in \mathbb{N}_0$ are defined as follows (for $F \in \mathcal{CLS}$):*

$$\mathrm{r}_0(F) := \begin{cases}\{\bot\} & \text{if } \bot \in F\\ F & \text{otherwise}\end{cases}$$

$$\mathrm{r}_{k+1}(F) := \begin{cases}\mathrm{r}_{k+1}(\langle x \to 1\rangle * F) & \text{if } \exists\, x \in \mathrm{lit}(F) : \mathrm{r}_k(\langle x \to 0\rangle * F) = \{\bot\}\\ F & \text{otherwise}\end{cases}.$$

r_1 is unit-clause propagation, r_2 is (full) failed literal elimination. In general we call r_k **generalised unit-clause-propagation of level** k. In [19] one finds the following basic observations proven (for $k \in \mathbb{N}_0$ and $F \in \mathcal{CLS}$):

- $\mathrm{r}_k : \mathcal{CLS} \to \mathcal{CLS}$ is well-defined (does not depend on the choices involved).
- r_k applies only forced assignments.
- $\mathrm{r}_k(F)$ is computable in time $O(\ell(F) \cdot n(F)^{2(k-1)})$ and linear space.

Definition 5 ([19,20]). *For $k \in \mathbb{N}_0$, clause-sets F and clauses C the relation $\boldsymbol{F \models_k C}$ holds if $\mathrm{r}_k(\varphi_C * F) = \{\bot\}$, where $\varphi_C := \langle x \to 0 : x \in C\rangle$.*

$F \models_1 C$ iff some subclause of C follows from F via input resolution. In [19] the *levelled height* "$h(T)$" of branching trees T has been introduced, which was further generalised in [20]. It handles satisfiable as well as unsatisfiable clause-sets. Here we will only use the unsatisfiable case. Then this measure reduces to a well-known measure which only considers the structure of the tree. [1] used the term "Horton-Strahler number", which is the oldest source (from 1945).

Definition 6. *The* ***Horton-Strahler*** *number* $\mathbf{hs(T)} \in \mathbb{N}_0$ *for a resolution tree* T *is defined as* $\mathrm{hs}(T) := 0$*, if* T *is trivial, while otherwise we have two subtrees* T_1, T_2*, and we set* $\mathrm{hs}(T) := \max(\mathrm{hs}(T_1), \mathrm{hs}(T_2))$ *if* $\mathrm{hs}(T_1) \neq \mathrm{hs}(T_2)$*, while in case of* $\mathrm{hs}(T_1) = \mathrm{hs}(T_2)$ *we set* $\mathrm{hs}(T) := \max(\mathrm{hs}(T_1), \mathrm{hs}(T_2)) + 1$.

See Sections 4.2, 4.3 in [19] for various characterisations of $\mathrm{hs}(T)$. In [19], Chapter 7 (generalised in [20], Chapter 5), *generalised input resolution* was introduced:

Definition 7 ([19,20]). *For a clause-set* F *and a clause* C *the relation* $\boldsymbol{F \vdash_k C}$ *(*C *can be derived from* F *by* k***-times nested input resolution****) holds if there exists a resolution tree* T *and* $C' \subseteq C$ *with* $T : F \vdash C'$ *and* $\mathrm{hs}(T) \leq k$.

By Parts 1 and 2 of Theorem 7.5 in [19], generalised in Corollary 5.12 in [20]:

Lemma 8 ([19,20]). *For clause-sets* F*, clauses* C *and* $k \in \mathbb{N}_0$ *we have* $F \models_k C$ *if and only if* $F \vdash_k C$.

5 Hardness

This section is devoted to the discussion of $\mathrm{hd} : \mathcal{CLS} \to \mathbb{N}_0$. It is the central concept of the paper, from which the hierarchy $\mathcal{UC}_k$ is derived (Definition 13). The basic idea is to start with some measurement $h : \mathcal{USAT} \to \mathbb{N}_0$ of "the complexity" of unsatisfiable F. This measure is extended to arbitrary $F \in \mathcal{CLS}$ by maximising over all "sub-instances" of F, that is, over all unsatisfiable $\varphi * F$ for (arbitrary) partial assignments φ. A first guess for $h : \mathcal{USAT} \to \mathbb{N}_0$ is to take something like the logarithm of the tree-resolution complexity of F. However this measure is too fine-grained, and doesn't yield a hierarchy like $\mathcal{UC}_k$. Another approach is algorithmical, measuring how far F is from being refutable by unit-clause propagation. As shown in [19,20], actually these two lines of thought can be brought together by the hardness measure $\mathrm{hd} : \mathcal{USAT} \to \mathbb{N}_0$.

Definition 9 ([19,20]). *The* ***hardness*** $\mathrm{hd}(F)$ *of an unsatisfiable* $F \in \mathcal{CLS}$ *is the minimal* $k \in \mathbb{N}_0$ *such that* $\mathrm{r}_k(F) = \{\bot\}$.

As shown in [19], $\mathrm{hd}(F)+1$ is precisely the clause-space complexity of F regarding tree-resolution. From [16] we gain the insight that for $F \in \mathcal{USAT}$ holds $\mathrm{hd}(F) \leq 1$ iff there exists $F' \subseteq F$ which is an unsatisfiable renamable Horn clause-set. By Theorem 7.8 (and Corollary 7.9) in [19] (or, more generally, Theorem 5.14 in [20]) we have for $F \in \mathcal{USAT}$ that $2^{\mathrm{hd}(F)} \leq \mathrm{Comp}^*_{\mathrm{R}}(F) \leq (n(F)+1)^{\mathrm{hd}(F)}$. Lemma 8 yields:

Lemma 10 ([19,20]). *For an unsatisfiable clause-set* F *and* $k \in \mathbb{N}_0$ *we have* $\mathrm{hd}(F) \leq k$ *iff* $F \models_k \bot$ *iff* $F \vdash_k \bot$.

By applying partial assignments we can reach all hardness-levels in a clause-set, as the following lemma shows (see [15] for the straightforward proof):

Lemma 11. *For an unsatisfiable clause-set F and every $0 \le k \le \mathrm{hd}(F)$ there exists a partial assignment φ with $n(\varphi) = k$ and $\mathrm{hd}(\varphi * F) = \mathrm{hd}(F) - k$.*

The hardness $\mathrm{hd}(F)$ of arbitrary clause-sets can now be defined as the maximum hardness over all unsatisfiable instances obtained by partial assignments.

Definition 12. *The **hardness** $\mathbf{hd}(F) \in \mathbb{N}_0$ for $F \in \mathcal{CLS}$ is the minimal $k \in \mathbb{N}_0$ such that for all clauses C with $F \models C$ we have $F \models_k C$ (recall Definition 5; by Lemma 8 this is equivalent to $F \vdash_k C$).*

In other words, if $F \neq \top$ then $\mathrm{hd}(F)$ is the maximum of $\mathrm{hd}(\varphi * F)$ for partial assignments φ such that $\varphi * F \in \mathcal{USAT}$. The measure $\mathrm{hd}(F)$ for satisfiable F apparently was mentioned the first time in the literature in [1], Definition 8, where there in Lemma 9 it was related to another hardness-alternative for satisfiable F. Note that one can restrict attention in Definition 12 to prime implicates C. Hardness 0 means that all prime clauses are there, i.e., $\mathrm{hd}(F) = 0$ iff $\mathrm{prc}_0(F) \subseteq F$, where $\mathrm{prc}_0(F)$ is the set of prime implicates of F.

Definition 13. *For $k \in \mathbb{N}_0$ let $\boldsymbol{\mathcal{UC}_k} := \{F \in \mathcal{CLS} : \mathrm{hd}(F) \le k\}$ (the class of **unit-refutation complete clause-sets of level** k).*

The class $\mathcal{UC}_1$ has been introduced in [10] for knowledge compilation. Various (resolution-based) algorithms computing for clause-sets F some equivalent set $F' \in \mathcal{UC}_1$ of prime implicates are discussed. Based on the results from [19,20], we can now give a powerful proof-theoretic characterisation for all classes $\mathcal{UC}_k$:

Theorem 14. *For $k \in \mathbb{N}_0$ and $F \in \mathcal{CLS}$ holds $F \in \mathcal{UC}_k$ if and only if $\forall\, C \in \mathrm{prc}_0(F) : F \vdash_k C$. Thus if every $C \in \mathrm{prc}_0(F)$ has a tree-resolution refutation using at most $2^{k+1} - 1$ leaves (i.e., $\mathrm{Comp}^*_{\mathrm{R}}(\varphi_C * F) < 2^{k+1}$), then $\mathrm{hd}(F) \le k$.*

Proof. The equivalence $F \in \mathcal{UC}_k \Leftrightarrow \forall\, C \in \mathrm{prc}_0(F) : F \vdash_k C$ follows from Lemma 8. And if $\mathrm{hd}(F) > k$, then there is $C \in \mathrm{prc}_0(F)$ with $F \nvdash_k C$, and then every tree-resolution derivation of C from F needs at least 2^{k+1} leaves due to $2^{\mathrm{hd}(\varphi_C * F)} \le \mathrm{Comp}^*_{\mathrm{R}}(\varphi_C * F)$ (as stated before). □

The following basic lemma follows directly by definition:

Lemma 15. *If two clause-sets F and F' are variable-disjoint, then we have:*

1. *If $F, F' \in \mathcal{SAT}$, then $\mathrm{hd}(F \cup F') = \max(\mathrm{hd}(F), \mathrm{hd}(F'))$.*
2. *If $F \in \mathcal{SAT}$ and $F' \in \mathcal{USAT}$, then $\mathrm{hd}(F \cup F') = \mathrm{hd}(F')$.*
3. *If $F, F' \in \mathcal{USAT}$, then $\mathrm{hd}(F \cup F') = \min(\mathrm{hd}(F), \mathrm{hd}(F'))$.*

Via full clause-sets A_n (clause-sets such that each clause contains all variables) with n variables and 2^n clauses we obtain (unsatisfiable, simplest) examples with $\mathrm{hd}(A_n) = n$, and when removing one clause for $n \ge 1$, then we obtain satisfiable examples A'_n with $\mathrm{hd}(A'_n) = n - 1$ (see [15] for the proof):

Lemma 16. *Consider a full clause-set F (each clause contains all variables).*

1. *If F is unsatisfiable then $\mathrm{hd}(F) = n(F)$.*
2. *If $F \neq \top$, then $\mathrm{hd}(F) = n(F) - \min_{C \in \mathrm{prc}_0(F)} |C|$.*
3. *If for F no two clauses are resolvable, then $\mathrm{hd}(F) = 0$.*

The next lemma yields a way of increasing hardness (see [15] for the proof):

Lemma 17. *Consider $F \in \mathcal{CLS}$ and $v \in \mathcal{VA} \setminus \mathrm{var}(F)$. Let $F' := \{C \cup \{v\} : C \in F\} \cup \{C \cup \{\overline{v}\} : C \in F\}$. Then $\mathrm{hd}(F') = \mathrm{hd}(F) + 1$.*

5.1 Containment and stability properties

The following fundamental lemma is obvious from the definition:

Lemma 18. *Consider $\mathcal{C} \subseteq \mathcal{CLS}$ stable under partial assignment and $k \in \mathbb{N}_0$ such that for $F \in \mathcal{C} \cap \mathcal{USAT}$ we have* $\mathrm{hd}(F) \leq k$. *Then* $\mathrm{hd}(F) \leq k$ *for all $F \in \mathcal{C}$.*

We apply Lemma 18 to various well-known classes $\mathcal{C}$ (stating in brackets the source for the bound on the unsatisfiable cases).

Lemma 19. *Consider $F \in \mathcal{CLS}$.*

1. *For $\varphi \in \mathcal{PASS}$ we have* $\mathrm{hd}(\varphi * F) \leq \mathrm{hd}(F)$ *(by Lemma 3.11 in [19]).*
2. $\mathrm{hd}(F) \leq n(F)$ *(by Lemma 3.18 in [19]).*
3. *If $F \in 2\text{–}\mathcal{CLS} = \{F \in \mathcal{CLS} \mid \forall\, C \in F : |C| \leq 2\}$, then* $\mathrm{hd}(F) \leq 2$ *(by Lemma 5.6 in [19]).*
4. *If $F \in \mathcal{HO} = \{F \in \mathcal{CLS} \mid \forall\, C \in F : |C \cap \mathcal{VA}| \leq 1\}$ (Horn clause-sets), then* $\mathrm{hd}(F) \leq 1$ *by (Lemma 5.8 in [19]).*
5. *More generally, if $F \in \mathcal{QHO}$, the set of q-Horn clause-sets (see Section 6.10.2 in [8], and [23]), then* $\mathrm{hd}(F) \leq 2$ *(by Lemma 5.12 in [19]).*
6. *Generalising Horn clause-sets to the hierarchy $\mathcal{HO}_k$ from [17] (with $\mathcal{HO}_1 = \mathcal{HO}$): if $F \in \mathcal{HO}_k$ for $k \in \mathbb{N}$, then* $\mathrm{hd}(F) \leq k$ *(by Lemma 5.10 in [19]).*

By a standard autarky-argument for $2\text{–}\mathcal{CLS}$ (see [18]) we can sharpen the hardness-upper-bound 2 for satisfiable clause-sets:

Lemma 20. *For $F \in 2\text{–}\mathcal{CLS} \cap \mathcal{SAT}$ we have* $\mathrm{hd}(F) \leq 1$.

Proof. Consider $\varphi \in \mathcal{PASS}$ with unsatisfiable $\varphi * F$. We have $\mathrm{r}_1(\varphi * F) = \{\bot\}$, since otherwise $\mathrm{r}_1(\varphi * F) \subseteq F$, and thus $\mathrm{r}_1(\varphi * F)$ would be satisfiable. □

We have the following stability properties:

Lemma 21. *Consider $k \in \mathbb{N}_0$.*

1. *$\mathcal{UC}_k$ is stable under application of partial assignments (with Lemma 19, Part 1; this might reduce hardness).*
2. *$\mathcal{UC}_k$ is stable under variable-disjoint union (with Lemma 15).*
3. *$\mathcal{UC}_k$ is stable under renaming variables and switching polarities.*
4. *$\mathcal{UC}_k$ is stable under subsumption-elimination.*
5. *$\mathcal{UC}_k$ is stable under addition of inferred clauses (this might reduce hardness).*

A fundamental tool, underlying all inclusion relations presented here, are the hierarchies $G_k(\mathcal{U}, \mathcal{S}) \subseteq \mathcal{CLS}$ introduced in [19,20], using oracles $\mathcal{U} \subseteq \mathcal{USAT}$, $\mathcal{S} \subseteq \mathcal{SAT}$ for (un)satisfiability decision. Only the unsatisfiable instances are relevant here (and thus $\mathcal{S}$ is not employed in our context): The G_k-hierarchies contain (finally all) satisfiable instances, as does $\mathcal{UC}_k$, but with a different aim, as can be seen from the fact, that for fixed k, membership in $G_k(\mathcal{U}, \mathcal{S})$ is decidable in polynomial time, while it is coNP-complete for $\mathcal{UC}_k$. See [15] for more details, and for the relations to the hierarchies presented in [6], which can be understood as special cases. These considerations lead to

Lemma 22. *For all $k \in \mathbb{N}_0$ we have $\Pi_k \subset \mathcal{UC}_{k+1}$ and $\Upsilon_k \subset \mathcal{UC}_{k+2}$ for the hierarchies Π_k, Υ_k introduced in [6].*

5.2 Determining hardness computationally

By the well-known computation of $\mathrm{prc}_0(F)$ via resolution-closure we obtain:

Lemma 23. *Whether for $F \in \mathcal{CLS}$ we have $\mathrm{hd}(F) = 0$ or not can be decided in polynomial time, namely $\mathrm{hd}(F) = 0$ holds if and only if F is stable under resolution modulo subsumption (which means that for all resolvable $C, D \in F$ with resolvent R there exists $E \in F$ with $E \subseteq R$).*

Thus if the hardness is known to be at most 1, we can compute it efficiently:

Corollary 24. *Consider a class $\mathcal{C} \subseteq \mathcal{CLS}$ of clause-sets where $\mathcal{C} \subseteq \mathcal{UC}_1$ is known. Then for $F \in \mathcal{C}$ one can compute $\mathrm{hd}(F) \in \{0, 1\}$ in polynomial time.*

Examples for $\mathcal{C}$ are given by $\mathcal{HO} \subset \mathcal{UC}_1$ and in Subsection 3.1. Another example class with known hardness is given by 2–$\mathcal{CLS} \subset \mathcal{UC}_2$ (Lemma 19), and also here we can compute the hardness efficiently (see [15] for the proof):

Lemma 25. *For $F \in$ 2–$\mathcal{CLS}$ the hardness is computable in polynomial time.*

See Theorem 29 for coNP-completeness of computing an upper bound.

6 The SLUR hierarchy

We now define the $\mathcal{SLUR}_k$ hierarchy, generalising $\mathcal{SLUR}$ (recall Subsection 3.1) in a natural way, by replacing r_1 with r_k. In Subsection 6.1 we show $\mathcal{SLUR}_k = \mathcal{UC}_k$, and as application obtain coNP-completeness of membership decision for $\mathcal{UC}_k$ for $k \geq 1$. In Section 6.2 we determine the relations to the previous hierarchies $\mathcal{SLUR}*(k)$ and CANON(k) as discussed in Subsection 3.2.

Definition 26. *Consider $k \in \mathbb{N}_0$. For clause-sets $F, F' \in \mathcal{CLS}$ the relation $\boldsymbol{F \xrightarrow{SLUR:k} F'}$ holds if there is $x \in \mathrm{lit}(F)$ such that $F' = \mathrm{r}_k(\langle x \to 1\rangle * F)$ and $F' \neq \{\bot\}$. The transitive-reflexive closure is denoted by $\boldsymbol{F \xrightarrow{SLUR:k}_* F'}$. The set of all fully reduced clause-sets reachable from F is denoted by $\mathbf{slur_k(F)} := \{F' \in \mathcal{CLS} \mid F \xrightarrow{SLUR:k}_* F' \wedge \neg\exists\, F'' \in \mathcal{CLS} : F' \xrightarrow{SLUR:k} F''\}$. Finally the class of all clause-sets which are either identified by r_k to be unsatisfiable, or where by k-SLUR-reduction always a satisfying assignment is found, is denoted by $\boldsymbol{\mathcal{SLUR}_k} := \{F \in \mathcal{CLS} : \mathrm{r}_k(F) \neq \{\bot\} \Rightarrow \mathrm{slur}_k(F) = \{\top\}\}$.*

We have $\mathcal{SLUR}_1 = \mathcal{SLUR}$. Obviously $\top \in \mathrm{slur}_k(F) \Leftrightarrow F \in \mathcal{SAT}$ for $F \in \mathcal{CLS}$ and all k. And by definition we get:

Lemma 27. *We have for $F \in \mathcal{CLS}$, $k \in \mathbb{N}_0$ and a partial assignment φ with $\mathrm{r}_k(\varphi * F) \neq \{\bot\}$ that $F \xrightarrow{SLUR:k}_* \mathrm{r}_k(\varphi * F)$ holds.*

6.1 SLUR = UC

For $F \in \mathcal{UC}_k$ there is the following polynomial-time SAT decision: F is unsatisfiable iff $\mathrm{r}_k(F) = \{\bot\}$. And a satisfying assignment can be found for satisfiable F via self-reduction, that is, probing variables, where unsatisfiability again is checked for by means of r_k. For $k = 1$ this means exactly that the nondeterministic "SLUR"-algorithm will not fail. And that implies that $F \in \mathcal{SLUR}$ holds,

where $\mathcal{SLUR}$ is the class of clause-sets where that algorithm never fails. So $\mathcal{UC}_1 \subseteq \mathcal{SLUR}$. Now it turns out, that actually this property characterises $\mathcal{UC}_1$, that is, $\mathcal{UC}_1 = \mathcal{SLUR}$ holds, which makes available the results on $\mathcal{SLUR}$. We now show that this equality between $\mathcal{UC}$ and $\mathcal{SLUR}$ holds in full generality for the $\mathcal{UC}_k$ and $\mathcal{SLUR}_k$ hierarchies.

Theorem 28. *For all $k \in \mathbb{N}_0$ holds $\mathcal{SLUR}_k = \mathcal{UC}_k$.*

Proof. Consider $F \in \mathcal{CLS}$. We have to show $F \in \mathcal{SLUR}_k \Leftrightarrow \mathrm{hd}(F) \leq k$. For $F \in \mathcal{USAT}$ this follows from the definitions, and thus we assume $F \in \mathcal{SAT}$. First consider $F \in \mathcal{SLUR}_k$. Consider a partial assignment φ such that $\varphi * F \in \mathcal{USAT}$. We have to show $\mathrm{r}_k(\varphi * F) = \{\bot\}$, and so assume $\mathrm{r}_k(\varphi * F) \neq \{\bot\}$. It follows $F \xrightarrow{\mathrm{SLUR}}_* \mathrm{r}_k(\varphi * F)$ by Lemma 27. In general we have that $F \in \mathcal{SLUR}_k$ together with $F \in \mathcal{SAT}$ and $F \xrightarrow{\mathrm{SLUR}:k}_* F'$ implies $F' \in \mathcal{SAT}$. Whence $\mathrm{r}_k(\varphi * F) \in \mathcal{SAT}$, contradicting $\varphi * F \in \mathcal{USAT}$. Now assume $\mathrm{hd}(F) \leq k$, and we show $F \in \mathcal{SLUR}_k$. For $F \xrightarrow{\mathrm{SLUR}:k}_* F'$ we have $F' \in \mathcal{SAT}$ by Lemma 19, Part 1, and thus $\mathrm{slur}_k(F) = \{\top\}$. □

Theorem 29. *For fixed $k \in \mathbb{N}$ the decision whether $\mathrm{hd}(F) \leq k$ (i.e., whether $F \in \mathcal{UC}_k$, or, by Theorem 28, whether $F \in \mathcal{SLUR}_k$) is coNP-complete.*

Proof. The decision whether $F \notin \mathcal{SLUR}_k$ is in NP by definition of $\mathcal{SLUR}_k$ (or use Lemma 11). By Theorem 3 in [7] we have that $\mathcal{SLUR}$ is coNP-complete, which by Lemma 17 can be lifted to higher k. □

6.2 Comparison to the previous hierarchies

The alternative hierarchies $\mathcal{SLUR}*(k)$ and $\mathrm{CANON}(k)$ (recall Subsection 3.2) extend r_1 in various ways (maintaining linear-time computation for the (non-deterministic) transitions). We give now short proofs that these alternative hierarchies are subsumed by our hierarchy, while already the second level of our hierarchy is (naturally) not contained in any levels of these two hierarchies (naturally, since the time-exponent for deciding whether a (non-deterministic) transition can be done w.r.t. hierarchy $\mathcal{SLUR}_k$ depends on k). First we simplify and generalise the main result of [2], that $\mathrm{CANON}(1) \subseteq \mathcal{SLUR}$.

Theorem 30. *For $k \in \mathbb{N}_0$ we have:* $\mathrm{CANON}(k) \subseteq \mathcal{UC}_k$ *and* $\mathcal{UC}_1 \not\subseteq \mathrm{CANON}(k)$ *(and thus* $\mathrm{CANON}(k) \subset \mathcal{UC}_k$ *for $k \geq 1$).*

Proof. By Theorem 14 and the fact, that the Horton-Strahler number of a tree is at most the height, we see $\mathrm{CANON}(k) \subseteq \mathcal{UC}_k$. There are formulas in $\mathcal{HO} \cap \mathcal{USAT}$ with arbitrary resolution-height complexity and so $\mathcal{HO} \not\subseteq \mathrm{CANON}(k)$. By $\mathcal{HO} \subset \mathcal{UC}_1$ we get $\mathcal{UC}_1 \not\subseteq \mathrm{CANON}(k)$. □

Also the other hierarchy $\mathcal{SLUR}*(k)$ is strictly contained in our hierarchy:

Theorem 31. *For all $k \in \mathbb{N}_0$ we have $\mathcal{SLUR}*(k) \subset \mathcal{SLUR}_{k+1}$ and $\mathcal{SLUR}_2 \not\subseteq \mathcal{SLUR}*(k)$.*

Proof. The inclusion follows most easily by using Lemma 18 together with the simple fact that $\mathrm{slur}*(k)(F) = \{F\}$ for $F \neq \top$ implies $\mathrm{r}_{k+1}(F) = \{\bot\}$. The non-inclusion follows from $\mathrm{CANON}(2) \not\subseteq \mathcal{SLUR}*(k)$ (Lemma 13 in [2]), while by Theorem 30 we have $\mathrm{CANON}(2) \subseteq \mathcal{SLUR}_2$. □

In [15] we show that $\mathcal{SLUR}*(k)$ and $\mathcal{SLUR}_k$ are incomparable in general.

7 Conclusion and outlook

We brought together two streams of research, one started by [10] in 1994, introducing $\mathcal{UC}$ for knowledge compilation, one started by [22] in 1995, introducing $\mathcal{SLUR}$ for polytime SAT decision. Two natural generalisations, $\mathcal{UC}_k$ and $\mathcal{SLUR}_k$ have been provided, and the (actually surprising) identity $\mathcal{SLUR}_k = \mathcal{UC}_k$ provides both sides of the equation with additional tools. Various basic lemmas have been shown, providing a framework for elegant and powerful proofs. Regarding computational problems, we solved the most basic questions. The next steps for us, which have already been partially accomplished, consist in the following investigations:

1. Complementary to "unit-refutation completeness" there is the notion of "propagation completeness", as investigated in [9,5]. This will be captured and generalised by a corresponding measure $\mathrm{phd} : \mathcal{CLS} \to \mathbb{N}_0$ of **propagation-hardness**.
2. The real power of SAT representations comes with **new variables**. Expressive power and limitations of the "good representations" have to be studied. Relevant here is [3], which shows that for example the satisfiable pigeonhole formulas PHP_m^m do not have polysize representations of bounded hardness.
3. Applications of representations of bounded hardness to cryptographic problems has to be experimentally evaluated. We consider especially attacking AES/DES, as preliminary discussed in [14,13].
4. The theory started here has to be generalised via the use of oracles as in [19,20] (this is one way of overcoming the principal barriers shown in [3], by employing oracles which can handle pigeonhole formulas).

References

1. Carlos Ansótegui, María Luisa Bonet, Jordi Levy, and Felip Manyà. Measuring the hardness of SAT instances. In Dieter Fox and Carla Gomes, editors, *Proceedings of the 23th AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 222–228, 2008.
2. Tomáš Balyo, Štefan Gurský, Petr Kučera, and Václav Vlček. On hierarchies over the SLUR class. In *Twelfth International Symposium on Artificial Intelligence and Mathematics (ISAIM 2012)*, January 2012. Available at `http://www.cs.uic.edu/bin/view/Isaim2012/AcceptedPapers`.
3. Christian Bessiere, George Katsirelos, Nina Narodytska, and Toby Walsh. Circuit complexity and decompositions of global constraints. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 412–418, 2009.
4. Armin Biere, Marijn J.H. Heule, Hans van Maaren, and Toby Walsh, editors. *Handbook of Satisfiability*, volume 185 of *Frontiers in Artificial Intelligence and Applications*. IOS Press, February 2009.
5. Lucas Bordeaux and Joao Marques-Silva. Knowledge compilation with empowerment. In Mária Bieliková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán, editors, *SOFSEM 2012: Theory and Practice of Computer Science*, volume 7147 of *Lecture Notes in Computer Science*, pages 612–624. Springer, 2012.
6. Ondřej Čepek and Petr Kučera. Known and new classes of generalized Horn formulae with polynomial recognition and SAT testing. *Discrete Applied Mathematics*, 149:14–52, 2005.

7. Ondřej Čepek, Petr Kučera, and Václav Vlček. Properties of SLUR formulae. In Mária Bieliková, Gerhard Friedrich, Georg Gottlob, Stefan Katzenbeisser, and György Turán, editors, *SOFSEM 2012: Theory and Practice of Computer Science*, volume 7147 of *LNCS Lecture Notes in Computer Science*, pages 177–189. Springer, 2012.
8. Yves Crama and Peter L. Hammer. *Boolean Functions: Theory, Algorithms, and Applications*, volume 142 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, 2011. ISBN 978-0-521-84751-3.
9. Adnan Darwiche and Knot Pipatsrisawat. On the power of clause-learning SAT solvers as resolution engines. *Artificial Intelligence*, 175(2):512–525, 2011.
10. Alvaro del Val. Tractable databases: How to make propositional unit resolution complete through compilation. In *Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 551–561, 1994.
11. John Franco. Relative size of certain polynomial time solvable subclasses of satisfiability. In Dingzhu Du, Jun Gu, and Panos M. Pardalos, editors, *Satisfiability Problem: Theory and Applications (DIMACS Workshop March 11-13, 1996)*, volume 35 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 211–223. American Mathematical Society, 1997. ISBN 0-8218-0479-0.
12. John Franco and Allen Van Gelder. A perspective on certain polynomial-time solvable classes of satisfiability. *Discrete Applied Mathematics*, 125:177–214, 2003.
13. Matthew Gwynne and Oliver Kullmann. Towards a better understanding of hardness. In *The Seventeenth International Conference on Principles and Practice of Constraint Programming (CP 2011): Doctoral Program Proceedings*, pages 37–42, September 2011. Proceedings available at `http://people.cs.kuleuven.be/~guido.tack/dp2011/DP_at_CP2011.pdf`.
14. Matthew Gwynne and Oliver Kullmann. Towards a better understanding of SAT translations. In Ulrich Berger and Denis Therien, editors, *Logic and Computational Complexity (LCC'11), as part of LICS 2011*, June 2011. 10 pages, available at `http://www.cs.swansea.ac.uk/lcc2011/`.
15. Matthew Gwynne and Oliver Kullmann. Generalising unit-refutation completeness and SLUR via nested input resolution. Technical Report arXiv:1204.6529v4 [cs.LO], arXiv, October 2012.
16. Lawrence J. Henschen and Lawrence Wos. Unit refutations and Horn sets. *Journal of the Association for Computing Machinery*, 21(4):590–605, October 1974.
17. Hans Kleine Büning. On generalized Horn formulas and k-resolution. *Theoretical Computer Science*, 116:405–413, 1993.
18. Hans Kleine Büning and Oliver Kullmann. Minimal unsatisfiability and autarkies. In Biere et al. [4], chapter 11, pages 339–401.
19. Oliver Kullmann. Investigating a general hierarchy of polynomially decidable classes of CNF's based on short tree-like resolution proofs. Technical Report TR99-041, Electronic Colloquium on Computational Complexity (ECCC), October 1999.
20. Oliver Kullmann. Upper and lower bounds on the complexity of generalised resolution and generalised constraint satisfaction problems. *Annals of Mathematics and Artificial Intelligence*, 40(3-4):303–352, March 2004.
21. Oliver Kullmann. Present and future of practical SAT solving. In Nadia Creignou, Phokion Kolaitis, and Heribert Vollmer, editors, *Complexity of Constraints: An Overview of Current Research Themes*, volume 5250 of *Lecture Notes in Computer Science (LNCS)*, pages 283–319. Springer, 2008. ISBN-10 3-540-92799-9.
22. John S. Schlipf, Fred S. Annexstein, John V. Franco, and R.P. Swaminathan. On finding solutions for extended Horn formulas. *Information Processing Letters*, 54:133–137, 1995.
23. Hans van Maaren. A short note on some tractable cases of the satisfiability problem. *Information and Computation*, 158(2):125–130, May 2000.