Without Loss of Generality – Symmetric Reasoning for Resolution Systems

(Extended Abstract for SymCon'07 Invited Talk)

Stefan Szeider
Department of Computer Science
Durham University
Durham, UK
stefan.szeider@durham.ac.uk

Abstract

Krishnamurthy [1985] introduced symmetry rules that make the informal "without loss of generality" reasoning available for resolution-based systems. The homomorphism rules of Szeider [2005] are more powerful variants of Krishnamurthy's rules that can save in certain cases an exponential number of inference steps over symmetry rules. In this talk we will review the concepts of symmetry and homomorphism rules and discuss various questions and results that arise in that context.

1 Resolution

Resolution is a reasoning method that is particularly suited for automated reasoning as it rests on one single rule of inference:

> If from a set of clauses one can derive the clauses $C \cup \{x\}$ and $D \cup \{\neg x\}$ then the *resolution rule* (Res) allows to derive also the clause $C \cup D$.

A *clause* is a set of literals and represents a disjunction; a literal is a variable or a negated variable. The clauses in the given set are *axioms* and can be derived immediately. It is well known that a set of clauses is unsatisfiable if and only if one can derive the empty clause from it using the resolution rule. A derivation of the empty clause is called a *refutation*.

Resolution is fundamental for many automated reasoning systems. Some unsatisfiable sets of clauses are "hard" for resolution: an exponential number of resolution steps is required to derive the empty clause. A famous hard example for resolution are the "pigeon hole clause sets." The pigeon hole clause set PH_n encodes (the negation of) the fact that $n+1$ pigeons do not fit into n holes if each hole can hold at most one pigeon. Using variable $x_{i,j}$ to represent the proposition "pigeon i sits in hole j" we can define PH_n as the set of the following clauses: all clauses $\{x_{i,1}, \dots, x_{i,n}\}$ for $1 \le i \le n+1$ ("pigeon i sits in some hole"), and all the clauses $\{\neg x_{i,j}, \neg x_{i',j}\}$ for $1 \le i < i' \le n+1$, $1 \le j \le n$ ("pigeons i and i' cannot sit in the same hole"). Haken [1985] has shown that it requires $2^{\Omega(n)}$ resolution steps to obtain the empty clause from PH_n.

Such exponential lower bounds for resolution imply that satisfiability solvers that are based on the Davis-Putnam-Logmann-Loveland procedure (DPLL) [Davis *et al.*, 1962] need exponential time for these instances, independent of the branching heuristics used. As demonstrated by Mitchel [1998] exponential resolution lower bounds are also significant for the running time of constraint solvers.

2 Symmetries and Homomorphisms

If we take into account that PH_n is highly symmetric, we can make the following inductive argument: Assume that PH_n is satisfiable. *Without loss of generality*, assume that pigeon $n+1$ sits in hole n; thus we set variable $x_{n+1,n}$ to true. This leaves us with PH_{n-1}. Repeating this step several times we obtain PH_1 which is evidently unsatisfiable.

Krishnamurthy [1985] suggested certain *symmetry rules* that formalize this type of reasoning for resolution-based systems. He distinguished between two variants of the symmetry rule: a *global symmetry rule* that takes into account symmetries of the given set of axioms as a whole, and a more powerful *local symmetry rule* that takes into account what axioms were actually used for deriving a certain clause. We will describe the rules more detailed below.

Krishnamurthy's symmetry rules are special cases of *homomorphism rules*; the latter were introduced by Szeider [2005] using the concept of *CNF homomorphism* [Szeider, 2003].

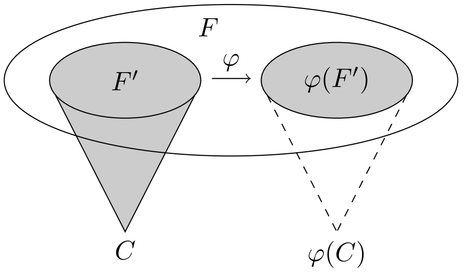

Figure 1: Illustration for the homomorphism rule. The homomorphism rule allows to obtain $\varphi(C)$ in one single step; $\varphi(C)$ could be derived from $\varphi(F')$ but this derivation (indicated by dashed lines) can be omitted.

Let F and G be sets of clauses. A *homomorphism from F to G* is a mapping φ from the literals of F to the literals of G such that the following conditions hold ($\varphi(C)$ denotes the set $\{\, \varphi(x) : x \in C \,\}$).

1. $\varphi(\neg x) = \neg\varphi(x)$ for all variables x of F ("φ preserves complements");
2. $\varphi(C) \in G$ for all clauses $C \in F$ ("φ preserves clauses").

For example, let $F = \{\{x\}, \{\neg y\}, \{\neg x, y\}\}$ and $G = \{\{z\}, \{\neg z\}\}$. The mapping φ with $\varphi(\neg x) = \varphi(y) = z$, and $\varphi(x) = \varphi(\neg y) = \neg z$, is a homomorphism from F to G. Note that $\varphi(\{\neg x, y\}) = \{z\}$, thus a homomorphism can "shrink" clauses.

If φ is a homomorphism from F to G and $F' \subseteq F$, then we write $\varphi(F') = \{\, \varphi(C) : C \in F' \,\}$. A *symmetry* φ is an injective homomorphisms; *i.e.*, a homomorphism where $\varphi(x) \neq \varphi(y)$ whenever $x \neq y$. A homomorphism is *nontrivial* if it differs from the identity mapping.

It is easy to see that if F is unsatisfiable and there exists a homomorphism from F to G, the G is unsatisfiable as well [Szeider, 2003].

Now we can formulate the following rule (see Figure 1 for an illustration).

> Give a set of clauses F. Assume that we can derive a clause C from a subset $F' \subseteq F$. If φ is a homomorphism from F' to F, then the *local homomorphism rule* (LHR) allows to derive the clause $\varphi(C)$.

The soundness of this rule can be seen as follows: we consider a sequence $C_1, \dots, C_n$ of clauses, with $C_n = C$, that corresponds to a derivation of C from F'. Applying φ we obtain the sequence $\varphi(C_1), \dots, \varphi(C_n)$, $\varphi(C_n) = \varphi(C)$. It is not difficult to see that the latter sequence contains a valid derivation of $\varphi(C)$ from the clauses in $\varphi(F')$. However, $\varphi(F') \subseteq F$, hence there exits a derivation of $\varphi(C)$ from F.

The argument outlined above suggests to consider refutations that use the homomorphism rule as a *succinct representation* of resolution refutations.

Restricting the local homomorphism rule in various ways leads to the following less general rules:

- The *global homomorphism rule* (GHR) arises from the local one by requesting that φ is an *endomorphism* of F, that is, φ is a homomorphism from F to F.
- Krishnamurty's *local symmetry rule* (LSR) arises from the local homomorphism rule by requesting that φ is a *monomorphism* from F' to F, *i.e.*, $\varphi(x) \neq \varphi(y)$ whenever $x \neq y$.
- Krishnamurty's *global symmetry rule* (GSR) arises from the local homomorphism rule by requesting that φ is an *automorphism* of F, *i.e.*, φ is both a monomorphism as well as an endomorphism of F.

Thus one can consider the following five resolution-based systems: Res, Res+GSR, Res+LSR, Res+GHR, and Res+LHR (for example, the system Res+GSR uses the resolution rule together with the global symmetry rule).

3 Comparison of the systems

We say that an unsatisfiable set F of clauses is *easy* for a system A if one can derive the empty clause from F using a polynomial number of inference steps of system A. If F requires an exponential number of steps, we say that F is *hard* for A. How are the above resolution-based systems related to each other? Is system A *strictly stronger* than system B in the sense that every instance that is easy for B is also easy for A, but some instances are easy for A and hard for B? Or are two systems A and B *incomparable* in the sense that there are instances that are easy for A and hard for B and instances where the converse prevails?

For the five systems under consideration the following is known (the first two statements have been established in [Urquhart, 1999] and [Arai and Urquhart, 2000], respectively; statements 3-6 have been established in [Szeider, 2005]).

1. Res+GSR is strictly stronger than Res.
2. Res+LSR is strictly stronger than Res+GSR.
3. Res+GHR is strictly stronger than Res+GSR.
4. Res+LHR is strictly stronger than Res+GHR.
5. Res+LHR is strictly stronger than Res+LSR.
6. Res+LSR and Res+GHR are incomparable.

The diagram in Figure 2 illustrates the relationships between the systems.

In the following let us briefly review the proof ideas used for establishing the above comparison results. Recall from above that the pigeon hole clause sets are hard instances for Res. For Res+GSR however, PH_n has as a short refutation as the following induction argument, given by Urquhart [1999], shows.

Evidently PH_1 has a short proof. Now consider PH_n for $n > 1$. Using the resolution rule we can derive from $\{\neg x_{n+1,n}, \neg x_{1,n}\} \in \mathrm{PH}_n$ and $\{x_{1,1}, \dots, x_{1,n-1}, x_{1,n}\} \in \mathrm{PH}_n$ the clause $\{x_{1,1}, \dots, x_{1,n-1}, \neg x_{n+1,n}\}$. Similarly we can derive all the clauses $\{x_{i,1}, \dots, x_{i,n-1}, \neg x_{n+1,n}\}$ for

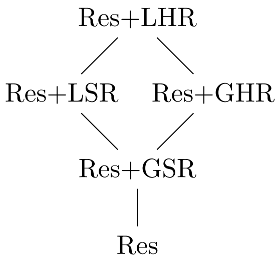

Figure 2: The relative strength of the considered reasoning systems.

$i = 2, \ldots, n$. Thus we have the clauses of PH_{n-1} except that some contain additionally the literal $\neg x_{n+1,n}$. By induction hypothesis there is a short Res+GSR-derivation of the empty clause from PH_{n-1}. This gives rise to a short Res+GSR-derivation of the unit clause $\{\neg x_{n+1,n}\}$ from PH_n. Evidently there is an automorphisms φ of PH_n such that $\varphi(x_{n+1,n}) = x_{n+1,n-1}$. Hence the global symmetry rule allows to derive the clause $\varphi(\{\neg x_{n+1,n}\}) = \{\neg x_{n+1,n-1}\}$. Similarly, we can use the global symmetry rule to derive all the clauses $\{\neg x_{n+1,n-2}\}, \ldots, \{\neg x_{n+1,1}\}$. Resolving all these unit clauses with the clause $\{x_{n+1,1}, \ldots,\ x_{n+1,n}\} \in \mathrm{PH}_n$ yields the empty clause.

Thus PH_n is hard for Res and easy for Res+GSR.

For the further comparison results one uses the following strategy. One "disguises" the clauses of PH_n in a certain way, making the symmetry/homomorphism rule under consideration inapplicable. The disguise is formed in such a way that the original clauses of PH_n can be obtained from the disguised clauses by means of a few resolution steps. This property is used to show that the disguised PH_n remains hard for Res.

The simplest disguising technique is to replace a clauses C by the clause $C \cup \{x\}$ and to add the unit clause $\{\neg x\}$; the original clauses C can be res-established by resolving $C \cup \{x\}$ with $\{\neg x\}$. Applying this technique one can disguise PH_n in such a way that every clause has a unique size (except for unit clauses). Thus the global symmetry rule is inapplicable.

The disguise can be designed in such a way that various symmetry/homomorphism rules cannot be applied any more. If a certain symmetry/homomorphism rule X is not applicable for the disguised PH_n, then we have an instance that is hard for the system Res+X. If we can design the disguise in such a way that rule X can still be applied but another more restricted rule Y cannot, we have shown that system Res+X is strictly stronger than system Res+Y.

There exists a disguise that makes the local homomorphism rule, the strongest of the rules considered, inapplicable [Szeider, 2005]. Hence there are hard instances for the system Res+LHR.

4 Algorithmic Aspects

In our above considerations we have asked for the *existence* of short refutations; we have not worried about the question of how a short refutation can actually be found. This "proof complexity" perspective has the advantage that hardness results apply to all possible algorithms that search for refutations: Even if we had an ideal heuristics that always tells us the best next move in our derivation, the running time of such an algorithm would still be exponential if the instance under consideration has no short refutation.

An algorithm that searches for proofs of one of the four systems considered in Section 2, one needs to find symmetries/homomorphisms in order to apply the respective rule.

The following decision problems arise in the context of applying the various symmetry/homomorphism rules.

P1 Has a given set F of clauses a nontrivial automorphism?

P2 Given a set F of clauses and a subset F' of F; is there a nontrivial monomorphism from F' to F?

P3 Has a given set F of clauses an endomorphism that is not an automorphism ?

P4 Given a set F of clauses and a subset F' of F; is there a nontrivial homomorphism from F' to F?

Problems P1, P2, P3, and P4 are associated with applications of the rules GSR, LSR, GHR, and LHR, respectively. All problems belong to NP as one can easily verify whether a given homomorphism has the required property. Except for problem P1, all other problems are known to be NP-hard (NP-hardness of P2 is shown in [Boy de la Tour and Demri, 1995]; NP-hardness of problems P3 and P4 is shown in [Szeider, 2005]). As observed by Boy de la Tour and Demri [1995], problem P1 is polynomial-time equivalent to the graph automorphism problem GA, which asks whether a given graph has a nontrivial automorphism. GA is a natural problem in NP that appears to be not solvable in polynomial time. However, it is believed that GA is not NP-complete since otherwise the Polynomial Hierarchy would collapse to its second level [Schöning, 1988]. Thus the weakest of the rules, GSR, is apparently computationally less costly than the other rules. Moreover, one can compute first the automorphism group at a preprocessing stage (cf. the discussion in [Boy de la Tour and Demri, 1995]). Some experimental results on the algorithmic use of a restricted version of GSR have been reported by Benhamou and Saïs [1994].

In summary, the worst-case complexities of the problems associated with applications of the symmetry/homomorphism rules, except for the weakest rule GSR, are not very encouraging, and a direct use of the rules in automated deduction seems difficult. However, it is conceivable that in certain situations one has additional information on the given instance that allows an efficient computation of homomorphisms and symmetries. This aspect seems to be of particular relevance if

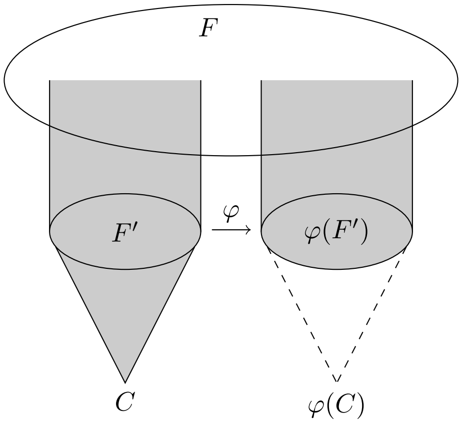

Figure 3: Illustration for the dynamic homomorphism rule.

the instance encodes a problem from a different domain, such as constraint satisfaction.

5 Dynamic Rules

The homomorphisms/symmetries of the above rules act on the clauses of the input set F, the axioms. However, one could apply first some inference steps in order to derive some clauses, and then apply the homomorphism/symmetry rules using homomorphisms/symmetries that act on the derived clauses. These considerations lead to the following definition (see Figure 3 for an illustration).

> Given a set F of clauses. Assume that we can derive from F the sets F' and F'', and assume that, in turn, we can derive from F' the clause C. If φ is a homomorphism from F' to F'', then the *dynamic homomorphism rule* (DHR) allows to derive the clause $\varphi(C)$.

The dynamic homomorphism rule is discussed in [Pitassi, 2003] and [Szeider, 2005].

The soundness of this rule can be established similarly as the soundness of the local homomorphism rule. In this more general setting, the above approach for finding hard instances does not work anymore: one can always derive the original PH_n from its disguise; once PH_n is derived, one can obtain the empty clause via a polynomial number of inference steps using resolution and symmetry rules that act on the derived clauses.

Currently we do not know any hard instances for the system Res+DHR.

References

[Arai and Urquhart, 2000] Noriko H. Arai and Alasdair Urquhart. Local symmetries in propositional logic. In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods (Proc. TABLEAUX 2000)*, volume 1847 of *Lecture Notes in Computer Science*, pages 40–51. Springer Verlag, 2000.

[Benhamou and Sais, 1994] Belaid Benhamou and Lakhdar Sais. Tractability through symmetries in propositional calculus. *Journal of Automated Reasoning*, 12:89–102, 1994.

[Boy de la Tour and Demri, 1995] Thierry Boy de la Tour and Stéphane Demri. On the complexity of extending ground resolution with symmetry rules. In *IJCAI-95, Vol. 1, 2 (Montreal, PQ, 1995)*, pages 289–295. Morgan Kaufmann, San Francisco, CA, 1995.

[Davis *et al.*, 1962] Martin Davis, George Logemann, and Donald Loveland. A machine program for theorem-proving. *Comm. ACM*, 5:394–397, 1962.

[Haken, 1985] Armin Haken. The intractability of resolution. *Theoret. Comput. Sci.*, 39:297–308, 1985.

[Krishnamurthy, 1985] Balakrishnan Krishnamurthy. Short proofs for tricky formulas. *Acta Informatica*, 22:253–275, 1985.

[Mitchell, 1998] David G. Mitchell. Hard problems for CSP algorithms. In *15th National Conference on Artificial Intelligence and 10th Innovative Applications of Artificial Intelligence Conference (AAAI'98/IAAI'98)*, pages 398–405. The MIT Press, 1998.

[Pitassi, 2003] Toniann Pitassi. Theory open problems session. Technical report, Department of Computer Science, University of Toronto, June 2003. `http://www.cs.toronto.edu/~toni/TheoryProblems/open.ps`.

[Schöning, 1988] Uwe Schöning. Graph isomorphism is in the low hierarchy. *J. of Computer and System Sciences*, 37:312–323, 1988.

[Szeider, 2003] Stefan Szeider. Homomorphisms of conjunctive normal forms. *Discr. Appl. Math.*, 130(2):351–365, 2003.

[Szeider, 2005] Stefan Szeider. The complexity of resolution with generalized symmetry rules. *Theory Comput. Syst.*, 38(2):171–188, 2005.

[Urquhart, 1999] Alasdair Urquhart. The symmetry rule in propositional logic. *Discr. Appl. Math.*, 96/97:177–193, 1999.