Computing maximal autarkies

Oliver Kullmann[1] João Marques-Silva[2]

[1] Computer Science Department
Swansea University, UK

[2] INESC-ID
Lisbon, Portugal

SAT 2015

September 25, 2015

Finding redundancies with few SAT calls

- “Autarkies” are some form of redundancies in CNFs, with good theoretical properties.
- We show how to determine all these redundancies.
- We capture them via “maximal autarkies”.
- And this with few simple SAT calls.

In our paper we develop a framework to represent all known algorithms.
In this talk we focus on the intuitions for the new algorithm.

Outline

Partial assignments and clause-sets

For

- $\varphi \in \mathcal{PASS}$ partial assignment
- $F \in \mathcal{CLS}$ clause-set

by

$$\varphi * F \in \mathcal{CLS}$$

the application of φ to F is denoted (removing satisfied clauses and falsified literals).

- The empty clause-set $\top := \emptyset \in \mathcal{CLS}$.
- φ satisfies F iff $\varphi * F = \top$.

Autarkies I

A partial assignment φ with

$$\forall\, C \in F : \varphi * \{C\} \in \{\top, \{C\}\}$$

is called an **autarky** for F (Monien and Speckenmeyer [7], Klee and Ladner [1], Kleine Büning and Kullmann [2]). That is:

Every clause of F touched by φ is satisfied by φ.

The extreme cases for autarkies:

1. The empty partial assignment $\langle\rangle$ is an autarky for every $F \in \mathcal{CLS}$, since $\langle\rangle * \{C\} = \{C\}$ for every clause C.
2. A satisfying assignment for F is also an autarky for F (here now $\varphi * \{C\} = \top$ for all $C \in F$).

Autarkies II

The otherwise simplest case of an autarky:

For a pure literal x for F, that is, $\overline{x} \notin \bigcup F$,
the assignment $\langle x \to 1 \rangle$ is an autarky for F.

The basic application of autarkies is

autarky reduction:

If φ is an autarky for F, then $\varphi * F$ is sat-equivalent to F.

Autarky reduction is confluent,
and the unique result is $\mathrm{N_a}(F) \subseteq F$,
the **lean kernel** of F.

The autarky-resolution duality

Consider $F \in \mathcal{CLS}$:

A clause $C \in F$ can be satisfied by some autarky for F
iff C can not be used in any resolution refutation of F.
((Kullmann [3])
I.e.: $\mathrm{N_a}(F)$ is the set of all clauses
usable in some resolution refutation of F.

That's obvious for

- satisfying assignments
- and pure literals.

Thus from a resolution refutation we obtain
variables not usable in any autarky.

Maximal autarkies I

A **maximal autarky** for $F \in \mathcal{CLS}$

- satisfies precisely all clauses of F not usable in any resolution refutation of F,
- and doesn't touch the other clauses.

And maximal autarkies do not leave variables of F unassigned, if they can be assigned. (Otherwise they are **quasi-maximal**.)

There is always a maximal autarky.
Okushi [8], Kullmann [3].

For a maximal autarky φ holds:

$$\varphi * F = \mathrm{N_a}(F).$$

Maximal autarkies II

Examples:

1. If F is **lean**, i.e., does not have non-trivial autarkies, then the unique maximal autarky for F is the empty assignment.
2. If F is satisfiable, then every satisfying partial assignment is quasi-maximal.

Applications

Clauses satisfied by some autarky

- cannot be included in MUSes (minimally unsatisfiable sub-clause-sets)
- and thus cannot be included in MCSes (minimal corrections sets, whose removal leads to a satisfiable clause-set).

The largest autark sub-clause-set (all clauses never usable in a resolution refutation) is contained in every maximal satisfiable sub-clause-set.

The general framework

For input $F \in \mathcal{CLS}$ (output maximal autarky):

1. Compute the SAT translation $t(F)$ (once).
2. Add **steering information.**
3. Solve that SAT problem.
4. If SAT, apply the autarky (to translation and steering).
5. If UNSAT, and the oracle provides information **on the refutation**, simplify translation and steering accordingly.
6. Repeat until nothing left.

Basic property of $t(F)$: "transparency" regarding SAT **and** UNSAT.

Grabbing autarkies via SAT solvers

Formulating autarky search as SAT problem:

$$F \mapsto t(F).$$

Exercise: find a translation yourself.
You need to represent the **three** states of a variable.

First systematic study of such translation: Marques-Silva, Ignatiev, Morgado, Manquinho, and Lynce [6].
Using the translation for computing maximal autarkies:

1. Find a non-trivial autarky.
2. *Non-triviality condition*: at least one variable is assigned.
3. Apply the autarky-reduction.
4. Repeat until lean.

Problem: Up to $n(F)$ calls.

Grabbing more? I

To find a maximal autarky in this way

with $\log_2(n(F))$ calls of the SAT oracle,

cardinality constraints are needed. And even one call is enough, when using *partial MaxSAT* (Liffiton and Sakallah [5]).

Too expensive!
We just want to use SAT oracles.
And very simple SAT calls (relative to F).

Grabbing more? II

We reuse $v \in \mathrm{var}(F)$ in $t(F)$ for indicating

whether the autarky envisaged by $t(F)$ uses v.

So the non-triviality condition is

$$\bigvee_{v \in \mathrm{var}(F)} v.$$

In order to get larger autarkies, we add a bunch of such **steering clauses**

$$\bigvee_{v \in V} v$$

now for $V \subseteq \mathrm{var}(F)$, to $t(F)$.

What if we grabbed too much?

The extended translation $t(F)$ is UNSAT iff for some of the $V \subseteq \mathrm{var}(F)$ we have

$$V \subseteq \mathrm{N_a}(F),$$

that is, no autarky uses any variable from V.

The main lemma now is:

The autarky-resolution duality extends to this extension of $t(F)$!

So if the solver returns UNSAT, and we know the **variables** involved in the refutation, then we know they are NOT in a maximal autarky.

Note on SAT oracle

Besides SAT decision, the SAT oracle in case of UNSAT

needs to return the set of variables involved in the refutation.

This is indeed much easier than returning the full refutation.

Outline of new algorithm

For input $F \in \mathcal{CLS}$, the algorithm computes $t(F)$ and a set S of steering clauses.

1. Run a SAT solver.
2. If SAT, then from all steering clauses we can remove at least one element — better, we apply the autarky.
3. If UNSAT, then we can eliminate at least one steering clause — better, we remove all parts of the translation which must use one of the variables involved in the refutation.

What does it achieve?

The optimum is achieved by

- partitioning $\text{var}(F)$ into $\sqrt{n(F)}$ many disjoint parts,
- and getting one steering clause from each part.

This achieves

at most $2\sqrt{n(F)}$ many SAT calls.

Summary and outlook

I The main open problem is: Can we do better?

II And what is a framework for proving lower bounds??

III In general, we are seeking for a refined framework for handling oracle calls, which can take into account

details of translations

as well as

fine-grained measurements of oracle calls,

and that for various types of oracles.

𝔈𝔫𝔡

(references on the remaining slides).

For my papers see
http://cs.swan.ac.uk/~csoliver/papers.html.

Bibliography I

[1] Victor Klee and Richard Ladner. Qualitative matrices: Strong sign-solvability and weak satisfiability. In Harvey J. Greenberg and John S. Maybee, editors, *Computer-Assisted Analysis and Model Simplification*, pages 293–320, 1981. doi:10.1016/B978-0-12-299680-1.50022-7. Proceedings of the First Symposium on Computer-Assisted Analysis and Model Simplification, University of Colorado, Boulder, Colorado, March 28, 1980.

[2] Hans Kleine Büning and Oliver Kullmann. Minimal unsatisfiability and autarkies. In 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*, chapter 11, pages 339–401. IOS Press, February 2009. ISBN 978-1-58603-929-5. doi:10.3233/978-1-58603-929-5-339.

Bibliography II

[3] Oliver Kullmann. Investigations on autark assignments. *Discrete Applied Mathematics*, 107:99–137, 2000. doi:10.1016/S0166-218X(00)00262-6.

[4] Oliver Kullmann, Inês Lynce, and João Marques-Silva. Categorisation of clauses in conjunctive normal forms: Minimally unsatisfiable sub-clause-sets and the lean kernel. In Armin Biere and Carla P. Gomes, editors, *Theory and Applications of Satisfiability Testing - SAT 2006*, volume 4121 of *Lecture Notes in Computer Science*, pages 22–35. Springer, 2006. ISBN 3-540-37206-7. doi:10.1007/11814948_4.

Bibliography III

[5] Mark Liffiton and Karem Sakallah. Searching for autarkies to trim unsatisfiable clause sets. In Hans Kleine Büning and Xishun Zhao, editors, *Theory and Applications of Satisfiability Testing - SAT 2008*, volume 4996 of *Lecture Notes in Computer Science*, pages 182–195. Springer, 2008. ISBN 978-3-540-79718-0. doi:10.1007/978-3-540-79719-7_18.

[6] Joao Marques-Silva, Alexey Ignatiev, Antonio Morgado, Vasco Manquinho, and Ines Lynce. Efficient autarkies. In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *21st European Conference on Artificial Intelligence (ECAI 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 603–608. IOS Press, 2014. doi:10.3233/978-1-61499-419-0-603.

[7] B. Monien and Ewald Speckenmeyer. Solving satisfiability in less than 2^n steps. *Discrete Applied Mathematics*, 10(3):287–295, March 1985. doi:10.1016/0166-218X(85)90050-2.

Bibliography IV

[8] Fumiaki Okushi. Parallel cooperative propositional theorem proving. *Annals of Mathematics and Artificial Intelligence*, 26(1-4): 59–85, 1999. doi:10.1023/A:1018946526109.