Good representations via full clause-learning

Oliver Kullmann

Swansea University, United Kingdom
http://cs.swan.ac.uk/~csoliver

Rome, September 27, 2012, Limits of Theorem Proving

Good representations of boolean functions

- Contrary to what I believed in my Cambridge talk this year, there are no good representations of PHP^m_m.
- Doesn't matter — interesting theory, relevant applications (and relativisation can overcome the PHP-barrier).

Collaboration with my student Matthew Gwynne.

Project home page

Representing boolean functions

- A

  clause-set F

  is a **(CNF-)representation** of

  a boolean function f

  if $\mathrm{var}(f) \subseteq \mathrm{var}(F)$ and the satisfying assignments of F projected to $\mathrm{var}(f)$ are (precisely) the satisfying assignments of f.
- Important special case: $\mathrm{var}(f) = \mathrm{var}(F)$, i.e., *without using new variables* (so F is equivalent to f).

Using QCNF we can say that F is a representation of f iff

$$f = \exists_{v \in \mathrm{var}(F) \setminus \mathrm{var}(f)} F.$$

Paradigmatic example for new variables: extension variables.

Less powerful: Circuits (= effective CNF-representations)

In the SAT context, a natural restriction on (CNF-)representations is to demand that evaluation of a *total* assignment for f is "effective" for F.

- Evaluation must be possible by unit-clause propagation.
- **Lemma** Such representations are basically "the same" as circuits.

We will actually consider here only further restricted forms of representations.

Example: PHP_m^m

PHP_m^m as boolean function:

- analogous to "all-different" constraint
- useful as building block for CNF-representations (requiring some bijection).

We have an effective representation of PHP_m^m (as boolean function), namely the usual clause-set (don't need injectivity-clauses here), which even does not use new variables.

Stronger condition: Partial evaluation

Consider a CNF-representation F of f:

Now we want for all **partial** assignments φ for f
such that $\varphi * f$ is unsatisfiable,
that UCP for $\varphi * F$ yields the empty clause.

That's needed for SAT solving!

We call that a 1-**soft representation**.

(More generally, k-soft means that for every clause C with $f \models C$ there is a tree resolution derivation $F \vdash C$ using space at most $k + 1$.)

Partialising boolean function

To boolean function f we associate boolean function $\hat{f}$, allowing to consider partial assignments:

- variables $v \in \mathrm{var}(f)$ are doubled: $v_0, v_1 \in \mathrm{var}(\hat{f})$.
- $\hat{f}$ true iff the corresponding partial assignment to f makes f unsatisfiable.
- If $v_0 = v_1 = 1$, then $\hat{f} = 1$.

$\hat{f}$ is monotone.

Partialising PHP_m^m

$\widehat{\mathrm{PHP}_m^m}$ determines whether a bipartite graph with at most m edges on each side admits a perfect matching (setting the missing edges to false).

- So we have a short circuit for $\widehat{\mathrm{PHP}_m^m}$ (perfect matching decision).
- However that's not useful for SAT (only for constraint solving):

  SAT needs partial evaluation.

Relative 1-softness = monotone circuits

From [BKNW09] follows:

Monotone circuits for $\hat{f}$ correspond effectively
to CNF-representations of f of relative softness 1.

Easiest from a monotone circuit for $\hat{f}$ to a CNF-representation F of relative softness 1: a monotone circuit which evaluates to 1 doesn't bother about input equal to 0 (which here means missing assignments).

Thus there is no poly-size CNF-representation
of relative softness 1 for PHP^m_m.

The UC hierarchy

$\mathcal{UC}_k$ is the set of clause-sets F such that for every $F \models C$ there is a tree-resolution derivation $F \vdash C$ using clause-space at most $k+1$.

Many equivalent characterisations, for example using generalised UCP.

A(n) (absolute) k-**soft representation** F for f is a representation $F \in \mathcal{UC}_k$ for f.

Hierarchy conjecture

For the relative condition everything collapses to $k = 1$, since there are no restrictions on the new variables (so they can absorb higher k).

Conjecture We have a true representation hierarchy for the absolute condition.

Partial result: can show this when not using new variables (here the relative and absolute condition coincide).

What is needed now is a hierarchy inside monotone circuits!

SLUR and clause-learning

In [GK12] we show

$$\mathcal{UC}_k = \mathcal{SLUR}_k$$

where $\mathcal{SLUR}$ is the class of clause-sets where "single look-ahead unit-resolution" is guaranteed to succeed, and $\mathcal{SLUR}_k$ generalises this via generalised UCP.
This means:

$\mathcal{UC}_1$ is the class of clause-sets closed under **clause learning**,
where the algorithm not just uses UCP,
but also look-ahead with UCP (to set a variable).

Note that here satisfiable clause-sets are most interesting. Finally, we can also use standard clause-learning, arriving at the hierarchy $\mathcal{PC}_k$ ("propagation-completeness").

End

(references on the remaining slides).

Bibliography I

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.

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.

Lucas Bordeaux, Mikoláš Janota, Joao Marques-Silva, and Pierre Marquis.
On unit-refutation complete formulae with existentially quantified variables.
In *Knowledge Representation 2012 (KR 2012)*, 2012.

Bibliography II

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.

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.

Bibliography III

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.

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.

Bibliography IV

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.

Matthew Gwynne and Oliver Kullmann.
Generalising unit-refutation completeness and SLUR via nested input resolution.
Technical Report arXiv:1204.6529v3 [cs.LO], arXiv, July 2012.

Bibliography V

Stasys Jukna.
Boolean Function Complexity: Advances and Frontiers, volume 27 of *Algorithms and Combinatorics*.
Springer, 2012.
ISBN 978-3-642-24507-7.

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.

Bibliography VI

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.

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.