Boolean Minimization

Ondřej Čepek

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Boolean seminar, Liblice, September 26, 2023

Knowledge representation languages

There are many ways in which Boolean function can be represented:

- Lists of vectors (truth table, set of true vectors (models))
- Boolean formulas (general formula, CNF, DNF)
- Decision diagrams (BDD, FBDD, OBDD)
- Negation normal forms (NNF, DNNF, d-NNF)

Knowledge representation preprocessing

- 1) preprocess input - done **offline** and only **once**, so it is OK if this step is (very) time consuming
- 2) use the result to answer queries (preferably by running a dedicated poly-time algorithm) - done **online** or **repeatedly**, should be efficient
- There are two main types of such preprocessing:

  - **Knowledge Compression**

  - **Knowledge Compilation**

Knowledge Compression = Boolean Minimization

Input: representation F of Boolean function f in language L
Output: representation G of f **also in L** but **smaller** than F
Examples:
- given circuit F design a logically equivalent circuit G with fewer gates
- given CNF F construct a logically equivalent CNF G with fewer clauses

Knowledge Compilation

Input: representation F of a Boolean function f in language L
Output: representation G of f **in some language L'** which is **better suited** for the intended use (expected queries)
Examples:
- given CNF F construct a logically equivalent OBDD G
- given NNF F construct a logically equivalent DNF G

CNF minimization - decision problem

Input: CNF F and $k \in \mathbb{N}$
Question: $\exists$ CNF $G : |G| \leq k$ and $G \equiv F$
How to measure CNF size: # of clauses (C) or total # of literals (L)

Tractable classes of CNFs

Class of CNFs is tractable if

- SAT is poly-time solvable in the class
- the class is closed under conditioning

Equivalence can be tested in poly-time for tractable classes, so for these classes

- SAT is in $P = \Sigma_0^p$
- minimization is in $NP = \Sigma_1^p$ and frequently it is also NP-hard

For many tractable classes (e.g. Horn CNFs) the gap between the complexity of SAT and BM is one level in the polynomial hierarchy.

General CNFs

Equivalence testing is co-NP-hard and CNF minimization is clearly in Σ_2^p $(\exists G : |G| \leq k \text{ and } \forall x : G(x) = F(x))$. In fact

- SAT is $NP - complete = \Sigma_1^p - complete$
- minimization is in $\Sigma_2^p - complete$ [Umans 2001] both for (C) and (L) measures

For general CNFs the gap between the complexity of SAT and BM is also one level in the polynomial hierarchy.

Some outliers

For the class of matched CNFs

- SAT is in P (in fact trivial)
- minimization is in $\Sigma_2^p - complete$ [Čepek, Gurský, Kučera 2014]

The complexity gap for matched CNFs is two levels in the polynomial hierarchy.

CNF minimization - optimization problem

Input: CNF F of a Boolean function f
Output: CNF G of f which is **smaller** than F
The decision problem is intractable $\Rightarrow$ look for approximation algorithms or heuristics. In the latter case, nontrivial lower bounds for the size of CNFs of the given function are needed to judge the quality of output.

Inapproximability results

- BM is inapproximable within a factor of $2^{log^{1-\epsilon}(n)}$ under a standard complexity-theoretic assumption [Bhattacharya, DasGupta, Mubayi, Turán 2010]
- BM is inapproximable within a factor of $2^{O(log^{1-o(1)} n)}$ if $P \neq NP$ [Boros and Gruber 2012]
- both results hold already for a subclass of Horn CNFs
- when approximation algorithms provably fail, heuristics is all that remains (and lower bounds are needed)

Heuristic minimization

A minimization algorithm based on heuristic decomposition technique [Boros, Čepek, Kučera 2013] is based on exclusive sets of implicates introduced in [Boros, Čepek, Kogan, Kučera 2010]

Exclusive sets of implicates

Let f be a Boolean function and X a set of implicates of f. Then X is an **exclusive** set of implicates if for every pair of resolvable implicates C, D

$$Resolvent(C, D) \in X \Rightarrow (C \in X \text{ and } D \in X)$$

- Both an intersection and a union of exclusive sets is again an exclusive set.
- If $F \equiv G$ are two CNFs representing f and X is an exclusive set of implicates of f then $(F \cap X) \equiv (G \cap X)$ and the corresponding subfunction is called an X-component of f.
- Exclusive components can be minimized independently of the rest of the input CNF which justifies a decomposition approach.

Essential prime implicates

A prime implicate of f is **essential** if it appears in every prime CNF of f.
Trivial lower bound on the (C) measure: $\#$ of essential prime implicates.
This concept was generalized in [Boros, Čepek, Kogan, Kučera 2010]

Essential sets of implicates

Let f be a Boolean function and X a set of implicates of f. Then X is an **essential** set of implicates if for every pair of resolvable implicates C, D

$$Resolvent(C, D) \in X \Rightarrow (C \in X \text{ or } D \in X)$$

- F is a CNF of $f \Leftrightarrow (F \cap X) \neq \emptyset$ for every nonempty essential set X.
- Pairwise disjoint essential sets $\Rightarrow$ lower bound on the (C) measure
- $ess(f)$ = max number of pairwise disjoint essential sets of f
- $cnf(f)$ = min number of clauses in a CNF representation of f
- Weak duality lower bound: $ess(f) \leq cnf(f)$
- This bound is not tight [Čepek, Kučera, Savický 2012] and in fact the gap can be very large [Hellerstein, Kletenik 2013]

Coverable classes of CNFs

A class Z of CNFs is called **coverable** if $ess(f) = cnf(f)$ for every function f representable by a CNF from Z [Čepek, Kučera, Savický 2012]

- Strong duality lower bound for coverable classes.
- All classes of CNFs for which a polynomial time minimization algorithm for the (C) measure exists (that I know of) are coverable.

Conjecture

Class Z of CNFs is coverable $\Rightarrow$ Z admits poly-time exact minimization for the (C) measure.

Horn CNF minimization - hardness results for the (C) measure

- *NP*-hard for general Horn CNFs (no degree bound) [Ausiello, D'Atri, Sacca 1986] unsatisfactory proof
- *NP*-hard for cubic Horn CNFs [Boros, Čepek 1994] proof has an error
- *NP*-hard for cubic Horn CNFs [Boros, Čepek, Kučera 2013]
- *NP*-hard for hydra CNFs (subclass of Horn 3-CNFs) [Kučera 2017]

Horn CNF minimization - hardness results for the (L) measure

- *NP*-hard for general Horn CNFs (no degree bound) [Maier 1980]
- reproved later [Ausiello, D'Atri, Sacca 1986] and [Hammer, Kogan 1993]
- *NP*-hard for Horn CNFs of degree 7 [Čepek 1995]
- *NP*-hard for Horn CNFs of degree 5 [Čepek,Kučera 2008]
- *NP*-hard for cubic Horn CNFs [Boros, Čepek, Kučera 2013]
- *NP*-hard for hydra CNFs (subclass of Horn 3-CNFs) [Kučera 2017]

Horn CNF minimization - classes with poly-time algorithms

- acyclic and quasi-acyclic Horn CNFs [Hammer, Kogan 1993]
- CQ-Horn CNFs (superclass of the previous two classes) [Boros, Čepek, Kogan, Kučera 2009]

Horn CNF minimization - poly-time approximation algorithms

- $O(n)$ – *factor* approximation algorithm for Horn CNFs for (C) [Hammer, Kogan 1993]
- $O(log(n))$-factor approximation algorithms for key Horn CNFs [Bercsi, Boros, Čepek, Kučera, Makino 2021]
- key Horn CNFs are a superclass of hydra CNFs
- two different algorithms, one for (C) and one for (L)

Representations of pure Horn functions

- pure Horn CNF

$$(\overline{a} \vee b) \wedge (\overline{b} \vee a) \wedge (\overline{a} \vee \overline{c} \vee d) \wedge (\overline{a} \vee \overline{c} \vee e)$$

- directed hypergraph

$$(\{a\}, b), (\{b\}, a), (\{a, c\}, d), (\{a, c\}, e)$$

- implicational (closure) system:

$$a \longrightarrow b, b \longrightarrow a, ac \longrightarrow d, ac \longrightarrow e$$

Adjacency list representations

- directed hypergraph: $\{a\} : b, \{b\} : a, \{a, c\} : d, e$
- implicational (closure) system: $a \longrightarrow b, b \longrightarrow a, ac \longrightarrow de$

Measures for the size of adjacency lists

The paper [Ausiello, D'Atri, Sacca 1986] introduced 5 different measures

1. # of source sets (# lists, # of rules)
2. # of hyperarcs (equivalent to (C) for CNFs)
3. # of hyperarcs + # of source sets
4. source area = sum of sizes of source sets
5. source area + # of hyperarcs

All measures except of the first one are NP-hard to minimize.
The number of source sets can be minimized in polynomial time and this fact was rediscovered many times in different contexts.

Poly-time algorithms minimizing the number of source sets

- database context (minimum covers in relational DB) [Maier 1980]
- closure systems context [Guigues, Duquenne 1986]
- directed hypergraph context [Ausiello, D'Atri, Sacca 1986]

Common properties of these algorithms

All three algorithms perform the following three steps:

1. Right saturate all rules
2. Left saturate all rules
3. Remove redundant rules

The output is a UNIQUE adjacency list representation (called a GD-base in closure systems context)

Proofs of correctness

All proofs are very technical, messy, and hard to read, all rely on the uniqueness of the output.

Why is # of source sets (bodies) easy to minimize?

- *bess*(f) = max number of pairwise BODY-disjoint essential sets of f
- *body*(f) = min number of bodies in a CNF representation of f
- Strong duality: *bess*(f) = *body*(f) holds for all pure Horn functions [Boros, Čepek, Makino 2017]

Algorithmic consequences of strong duality

- Left saturation step can be skipped
- Output guaranteed to be minimal
- Uniqueness of the output is lost
- Asymptotic complexity stays the same
- Running time of course decreases
- The minimization algorithm is conceptually much simpler

THANK YOU FOR YOUR ATTENTION