WDS'11 Proceedings of Contributed Papers, Part I, 101–105, 2011. *ISBN 978-80-7378-184-2 © MATFYZPRESS*

Minimization of Matched Formulas

Š. Gurský

Charles University, Faculty of Mathematics and Physics, Prague, Czech Republic.

Abstract. Decision problems of finding minimal representation of a given Boolean formula is known to be Σ_2 complete for formulas in conjunctive normal form and for two basic measures of minimality – number of occurences of literals or numbers of clauses. In this paper we study this problem restricted to a class of satisfiable Boolean CNF formulas called matched formulas. We present several variants of minimization and show that their complexity does not differ from general case. Then we show that minimization remains Σ_2 complete even restricted to matched formulas.

Introduction

Finding a minimal representation of a given object is a common task in many areas. In this paper the area of interest is Boolean minimization where we deal with Boolean functions represented in conjunctive normal form (CNF). The task is to find a shortest possible CNF representation for a given function with respect to a given measure.

This paper deals with decision problems connected with Boolean minimization and discusses their complexity. It is known that deciding whether there exists shorter CNF for function given as CNF is Σ_2 complete for general formulas. However, for certain classes of formulas the complexity of minimization is different.

In this paper we deal with class of satisfiable formulas where matching of variables to clauses exists. These formulas are called matched and can be recognized in polynomial time. Several subproblems connected to minimization are examined and their complexity is set. Then, building on a work by C. Umans, complexity of minimization of matched formulas is established.

Definitions

We assume unlimited supply of Boolean variables $x_1, x_2, \ldots$ that can have value either true or false. Literal is a variable or its negation. Clause is a disjunction of literals. Value of a clause is true, if any of its literals is true. Value of an empty clause is false. Conjunctive normal form (CNF) is a conjunction of clauses. CNF evaluates to true if all its clauses evaluate to true. Empty CNF evaluates to true.

We will also need dual objects to clauses and CNFs – terms and DNFs. Term is a conjunction of literals. An empty term's value is true. Disjunctive normal form (DNF) is a disjunction of terms. Empty DNF evaluates to false.

It is a well known fact, that any Boolean function can be represented by both CNF and DNF, often in more than one way. We will say, that two formulas φ and ψ are equivalent if they represent the same function.

Definition 1 (Incidence graph). For CNF φ, the *incidence graph* G is a bipartite graph, where vertices of one partity are clauses of φ and vertices of the other partity are its variables. A clause is connected to variables that it contains (that is clause C is connected to variable x if and only if it contains either literal x or literal $\overline{x}$).

Definition 2 (Matched CNF). CNF is called *matched* if its incidence graph has matching that covers all clauses. That is every clause can be assigned its own variable.

Matched DNFs can be defined in an analogous way.

Matched formulas were first defined in [*Franco and Van Gelder*, 2003] and the following simple observation was made.

Observation 1 ([*Franco and Van Gelder*, 2003]). Every matched CNF is satisfiable.

Proof. For clause C take its matched variable x. If C contains literal x, set x to true, othervise set it to false. This way all clauses can be satisfied. □

Minimization

General case

We will look at Boolean minimization as a decision problem. There are several flavors of the problem. In all of them we are given a formula and a limit and the question is whether there is an equivalent formula that has size smaller than the limit with respect to a given measure. Let us first look at nonrestricted cases of minimization problems.

The first natural limit is number of clauses in a CNF or number of terms in a DNF. Formal statements of problems follow.
Problem: MinCNFClauses
Input: CNF φ and integer k
Question: Is there a CNF ψ equivalent to φ that has at most k clauses?

Problem: MinDNFTerms
Input: DNF φ and integer k
Question: Is there a DNF ψ equivalent to φ that has at most k terms?

Lemma 1. Problems MinCNFClauses and MinDNFTerms are equivalent, that is one can be reduced to the other in polynomial time.

Proof. Given the instance φ and k of MinDNFTerms, we create CNF φ' that is negation of φ by switching ORs and ANDs and by negating each literal. φ' and k then represent an instance of MinCNFClauses and it is positive instance if and only if the φ and k were the positive instance of MinDNFTerms. That is because if φ has an equivalent DNF ψ that has at most k terms, then ψ' (negation of ψ) is CNF equivalent to φ' that has at most k clauses and vice versa. Reduction in the opposite direction is identical. □

Another measure of CNF or DNF length is number of occurences of literals.
Problem: MinCNFLit
Input: CNF φ and integer k
Question: Is there a CNF ψ equivalent to φ that has at most k occurences of literals?

Problem: MinDNFLit
Input: DNF φ and integer k
Question: Is there a DNF ψ equivalent to φ that has at most k occurences of literals?

Lemma 2. Problems MinCNFLit and MinDNFLit are equivalent in the same sense as in Lemma 1.

Proof. Proof of this lemma is identical to the proof of Lemma 1. □

From these two lemmas we see, that we can choose either CNF or DNF variant of any problem, find its complexity and then apply the result to the other normal form. The complexity of minimization of DNFs was determined by Christopher Umans in 2000.

Theorem 1 ([*Umans*, 1998, 1999, 2000]). The problems MinDNFTerms and MinDNFLit are both Σ_2 complete.

From lemmas 1, 2 and Theorem 1 we get the following corollary.

Corollary 1. Problems MinCNFClauses and MinCNFLit are also Σ_2 complete.

Matched formulas

In general case the minimization is Σ_2 complete. However, there are classes of formulas where minimization is easier. For example minimization of Horn formulas (those with at most one positive literal in each clause) is known to be NP complete [*Hammer and Kogan*, 1993]. For monotone formulas (those that have no negation) the minimization is trivial. In both of these classes of CNFs, the famous Satisfiability problem can be solved in polynomial time, so one can examine minimization in other classes of formulas with this property and expect minimization to be easier than in general case.

Let us concentrate on matched formulas. We will look at several smaller problems that are connected with minimization and determine the complexity of these problems. These problems deal with finding unnecessary parts of the formula — parts that can be removed from the formula without changing the function it represents.
Problem: Dependency on Variable
Input: Matched CNF φ and variable x.
Question: Does the function represented by φ depend on x?

Theorem 2. The problem Dependency on Variable is NP complete.

Proof. The problem is in NP. The certificate would be pair of assignments v_1 and v_2 that differ only in x, but $v_1(\varphi) \neq v_2(\varphi)$.

To show NP hardness, consider an instance of Satisfiability $\varphi = C_1 \wedge C_2 \wedge \ldots \wedge C_n$. From this we construct $\psi = (C_1 \vee w_1) \wedge (C_2 \vee w_2) \wedge \cdots \wedge (C_n \vee w_n) \wedge (w_1 \vee w_2 \vee \cdots \vee w_n \vee x)$, an instance of Dependency on Variable. ψ is clearly matched (Clauses $(C_i \vee w_1)$ can be matched with w_i and the last clause can be matched with x). Suppose that φ is satisfiable with assignment v. Construct assignment v' by extending v such that w_i is false for every i and assignments v_1 and v_2 that both extend v' but v_1 sets x to true and v_2 sets x to false. These two assignments differ only in x but v_1 satisfies ψ and v_2 does not.

Suppose φ is not satisfiable. Then for assignment that sets every w_i to false ψ also evaluates to false and does not depend on value of x. For assignment that sets any of w_i to true the last clause of ψ is satisfied and ψ does not depend on x. φ is satisfied if and only if ψ depends on x. □

It should be noted that x occurs in φ from previous proof only once. Let us call DoV1 a restricted version of the previous problem, where we limit number of occurences of x to one. DoV1 is then also NP complete.
Problem: Removal of Variable
Input: Matched CNF φ and variable x.
Question: Can x be removed from φ without changing the function it represents?

Theorem 3. The problem Removal of Variable is coNP complete.

Proof. The problem is clearly in coNP. For negative answer the certificate would be an assignment on which the input formula φ and φ with x removed would have different value. coNP hardness can be shown by reduction from DoV1. Given instance φ and x of DoV1, where x occurs only once in φ, the corresponding instance of Removal of Variable is also φ and x. If φ depends on x, it cannot be removed, since the result would not contain x. If φ does not depend on x then removing x is same as setting x to false (it occurs only once in φ) and for all assignments this is same as setting x to true (φ does not depend on x), so x can be removed. Therefore x can be removed if and only if φ does not depend on it. □

Problem: Removal of Clause
Input: Matched CNF φ and its clause C.
Question: Can C be removed from φ without changing the function?

Theorem 4. The problem Removal of Clause is coNP complete.

Proof. This proof is almost identical to the proof of Theorem 3. Problem belongs to coNP since certificate for negative answer is an assignment where φ has value opposite to φ with C removed. To show coNP hardness we again start from instance of DoV1 and ask whether we can remove clause containing x. If φ does depend on x the clause cannot be removed. If φ does not depend on x then setting x to true is same as setting x to false and removal of clause containing x is equivalent to setting x to true. Therefore φ does not depend on x if and only if clause containing x can be removed. □

The next problem is not exactly minimization subproblem, however, it is an interesting one and we can use the previous problem to assess its complexity.
Problem: Is Implicate
Input: Matched CNF φ and some clause C.
Question: Does every assignment that satisfies φ also satisfy C?

Theorem 5. The problem Is Implicate is coNP complete.

Proof. The problem is in coNP. The certificate for negative answer is an assignment which sets φ to true but C to false. The coNP hardness can be seen from the previous problem. Instance φ and C of problem Removal of Clause is a positive instance if and only if $\varphi \setminus C$ and C are positive instance of problem Is Implicate. □

And now we look at real minimization problems. For general formulas we had problems MinCNFClause, MinCNFLit and their respective DNF variants. For matched formulas we limit CNFs (DNFs) in problem description to matched formulas. The problem statements for DNF follow. For CNF the problems are analogous.
Problem: MinDNFTermMatch
Input: Matched DNF φ and integer k
Question: Is there DNF ψ that is equivalent to φ and has at most k clauses?

Problem: MinDNFLitMatch
Input: Matched DNF φ and integer k
Question: Is there DNF ψ that is equivalent to φ and has at most k occurences of literals?

Using the same technique as in the proof of Lemma 1 we can show that CNF variants, problems MinCNFClauseMatch and MinCNFLitMatch, have the same complexity as their DNF equivalents. Therefore, we can use version that we want, to establish the complexity for both CNF and DNF (note that we cannot say anything about equivalence of problems that deal with different measures of formula length).

Theorem 6. Both problems MinDNFTermMatch and MinDNFLitMatch are Σ_2 complete.

Sketch of proof. The proofs of both these statements follow very closely the original proofs of C. Umans for general case with certain ammendments. Booth proofs are quite long and can be found in full in [*Gurský*, 2010]. The original proof for MinDNFLit can be found in [*Umans*, 1998], the proof for MinDNFTerm can be found in [*Umans*, 1999], both in [*Umans*, 2000] and cleaned up versions in [*Gurský*, 2010]. We will only show how to get a proof of complexity of minimization of matched formulas from the original version.

In both proofs the instance of certain Σ_2 complete problem is reduced to instance of problem at hand. We present here only the formula in the resulting instance and a way it can be made matched formula without breaking the proof of complexity.

Let us first look at MinDNFLit and therefore at MinDNFLitMatch. In his proof Umans reduced an instance of another Σ_2 complete problem Shortest Implicant Core (proof of Σ_2 completness provided therein) to instance of MinDNFLit. The formula in the resulting instance of MinDNFLit is in the form $\varphi'' = t_l w_1 w_2 w_3 \dots w_{m'} \vee \bigvee_{i=1}^{m} s'_i$ where s'_i is in the form $s_i w_1 w_2 \dots w_{i-1} w_{i+1} \dots w_{m'}$ with w_i being variable for all i and t_l and s_i are terms for all i. Important fact is that in this formula we can find matching that matches each of s'_i terms with variable w_{i+1} (the last one with w_1) and the first term can be matched with any of the variables in t_l (it is not empty). Therefore this formula is matched and problem MinDNFLitMatch is Σ_2 complete.

In proof of MinDNFTerm, C. Umans reduces again problem Shortest Implicant Core (albeit a somewhat different version) to an instance of MinDNFTerm. The resulting instance in this case is a formula that consists of two parts. The first part are terms in form $s'_i = s_i z_1 z_2 \dots z_{i-1} z_{i+1} \dots z_m$ with s_i being term and z_i being variable for each i. Each s'_i can be then matched with z_{i+1}. The second part of formula has terms in form $u_{i,j} = (p_i)\bar{x}_j z_1 z_2 \dots z_m$, where all parts are variables. However, we can change p_i from being a variable to being a term of parity function on new set of variables $a_1, a_2 \dots$ such that there is enough of a-variables to provide matching for all terms. The parity is used since it cannot be shortened in any way. The formula length grows only polynomially (each term gets polynomial amount of new variables) and the resulting formula is again matched. So MinDNFLitMatch is also Σ_2 complete. □

Corollary 2. Problems MinCNFClauseMatch and MinCNFLitMatch are also Σ_2 complete.

Proof. Same as in lemmas 1 and 2. □

Conclusion

We saw, that the class of matched formulas has the same complexity of minimization and attached problems as a general formulas. This result is partially surprising, since the matched formulas have a simple structure and the "standard" hard problem (SATISFIABILITY for CNF) is trivial for them.

Future work can be done on discovering distinguishing property between classes of formulas with Σ_2 complete minimization and classes whose minimization is in lower complexity level.

References

Franco J., Van Gelder A., A perspective on certain polynomial-time solvable classes of satisfiability, in *Discrete Appl. Math.*, 125, 177-214, 2003

Gurský, Š., Časová zložitosť minimalizácie Booleovských funkcií, diploma thesis, Charles University, Faculty of Mathematics and Physics, Prague, 2010, available online at http://artax.karlin.mff.cuni.cz/ gurss5am/diplomka.pdf (accessed on 2011-06-11).

Hammer P., Kogan A., Optimal compression of propositional Horn knowledge bases: complexity and approximation, in: *Artif. Intell.*, 64, 131-145, 1993

Umans C., The minimum equivalent DNF problem and shortest implicants, in: Foundations of Computer Science, 1998. Proceedings.39th Annual Symposium on, 556 -563, 1998

Umans, C., Hardness of Approximating Σ_2^p Minimization Problems, in: *FOCS '99: Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 465-474, 1999

Umans C., Approximability and completeness in the polynomial hierarchy, disertation thesis, University of California, Berkeley, 2000