The Minimum Equivalent DNF Problem and Shortest Implicants

Christopher Umans*
Computer Science Division
University of California, Berkeley
umans@cs.berkeley.edu

Abstract

We prove that the Minimum Equivalent DNF problem is Σ_2^p-complete, resolving a conjecture due to Stockmeyer. The proof involves as an intermediate step a variant of a related problem in logic minimization, namely, that of finding the shortest implicant of a Boolean function. We also obtain certain results concerning the complexity of the Shortest Implicant problem that may be of independent interest. When the input is a formula, the Shortest Implicant problem is Σ_2^p-complete, and Σ_2^p-hard to approximate to within an $n^{1/2-\epsilon}$ factor. When the input is a circuit, approximation is Σ_2^p-hard to within an $n^{1-\epsilon}$ factor. However, when the input is a DNF formula, the Shortest Implicant problem cannot be Σ_2^p-complete unless $\Sigma_2^p = NP[\log^2 n]^{NP}$.

1. Introduction

Two-level (DNF) logic minimization is a central practical problem in logic synthesis and also one of the more natural problems in the polynomial hierarchy. In his well-know polynomial hierarchy paper [12], Stockmeyer conjectured that the following logic minimization problem is Σ_2^p-complete:

MINIMUM EQUIVALENT DNF (MIN DNF). Given a DNF formula ϕ and an integer k, is there a DNF formula equivalent to ϕ that contains k or fewer occurrences of literals?

In this paper we prove Stockmeyer's conjecture.

Our proof involves a variant of a related problem, that of finding a shortest implicant of a Boolean function. The connection between MIN DNF and implicant problems is not surprising; since the introduction of the now-classic Quine-McCluskey algorithm [11, 8], most heuristic and exact methods for two-level logic minimization have been based on explicitly or implicitly generating the prime implicants of the input function (see [3, 13]). Our results show that the close connection between MIN DNF and finding implicants is the source of at least one aspect of its difficulty.

The shortest implicant problem itself is worth investigating, as a computationally difficult but conceptually simple optimization problem. In this paper, we examine the complexity of the following three variants of the basic problem:

SHORTEST IMPLICANT[DNF]. Given a DNF formula ϕ and an integer k, is there an implicant of ϕ that contains k or fewer literals?

SHORTEST IMPLICANT[FORMULA]. Given a $(\wedge, \vee, \neg)$-formula ϕ and an integer k, is there an implicant of ϕ that contains k or fewer literals?

SHORTEST IMPLICANT[CIRCUIT]. Given a $(\wedge, \vee, \neg)$-circuit ϕ and an integer k, is there an implicant of ϕ that contains k or fewer literals?

In addition to proving that MIN DNF is Σ_2^p-complete, we obtain the following results for the shortest implicant problem:

1. SHORTEST IMPLICANT[DNF] is in NP$[\log^2 n]^{\text{NP}}$ and therefore it is unlikely to be Σ_2^p-complete.
2. SHORTEST IMPLICANT[FORMULA] is Σ_2^p-complete and Σ_2^p-hard to approximate to within an $n^{1/2-\epsilon}$ factor.
3. SHORTEST IMPLICANT[CIRCUIT] is Σ_2^p-complete and Σ_2^p-hard to approximate to within an $n^{1-\epsilon}$ factor.

The number of papers on exact algorithms and heuristics for both DNF minimization and implicant finding is too great to survey here. We direct the reader to [3] and [4] for an overview. Comparatively little research has focused on the computational complexity of these problems. Since its introduction in [12], no progress was reported on the hardness of MIN DNF until the recent paper by Hemaspaandra and Wechsung [6], in which they prove that MIN DNF is

*Supported by NSF grant CCR-9626361 and an NSF Graduate Research Fellowship.

$P_{||}^{NP}$-hard. Their proof relies on a result of Wagner [14] that allows certain NP-hardness results to be boosted to $P_{||}^{NP}$-hardness results. Our approach is different, and we are able to show Σ_2^p-completeness directly with a two-step reduction.

To our knowledge, this paper is the first to consider the computational complexity of the shortest implicant problem, although variants of the problem are similar to certain "circuit subfunction relation" problems shown to be Σ_2^p-complete in [2]. Our result for SHORTEST IMPLICANT[DNF] should be contrasted with the proof in [1] that the Boolean Isomorphism problem cannot be Σ_2^p-complete unless the polynomial hierarchy collapses. Our conditional collapse is weaker; to show a full PH collapse if SHORTEST IMPLICANT[DNF] is Σ_2^p-complete would likely involve new general results on the power of limited non-determinism.

2. Preliminaries

A DNF formula (or just DNF) is a disjunction of conjunctions of literals. Each conjunction is a *term* of the formula. We will also be concerned with formulae and circuits over $(\wedge, \vee, \neg)$; their precise definitions may be found in any standard text on Boolean function complexity. Every formula and circuit computes a Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ in the obvious way. If ϕ is a formula or circuit, we will also use ϕ to denote the function computed by ϕ.

An *implicant* of a Boolean function f is a conjunction C that implies f. An implicant *covers* an assignment to f if the assignment makes the implicant true. An implicant C_1 *subsumes* another implicant C_2 if C_1 covers every assignment covered by C_2. In keeping with the logic synthesis literature, we write $C_1 \supseteq C_2$ for "C_1 subsumes C_2." However, notice that if this is true, then C_1 actually consists of a subset of the literals of C_2. The notion of implicants as sets of literals is useful, and we will occasionally manipulate them as sets. A *prime implicant* is an implicant that is not subsumed by any other implicant except itself.

An *partial assignment* is a mapping from the set of variables $\{x_1, x_2, \ldots x_n\}$ into $\{0, 1, *\}$. If ϕ is a formula, and A is a partial assignment, then we denote by ϕ_A the formula obtained by substituting the appropriate constants for each variable that A maps to 0 or 1. Two partial assignments are *consistent* if there is no x_i that is mapped to 0 by one and 1 by the other. We may combine partial assignments A and B if they are consistent, obtaining AB. If a partial assignment A does not map any variable to $*$, then it is simply an assignment, which we emphasize by writing $\phi(A)$ instead of ϕ_A.

For each conjunction C, there is an associated partial assignment that maps the variable x_i to 1 if the literal x_i appears in C, to 0 if the literal $\overline{x_i}$ appears in C, and to $*$ otherwise. By this interpretation, a conjunction C is an implicant of ϕ if and only if $\phi_A \equiv 1$, where A is the partial assignment associated with C. Two conjunctions are consistent if their associated partial assignments are consistent; a partial assignment A is consistent with a conjunction C if the partial assignment associated with C is consistent with A.

We use $|\phi|$ to denote the length of ϕ. For formulae, $|\phi|$ is interpreted as the number of occurrences of literals in ϕ; for circuits, $|\phi|$ is the number of gates in the circuit. We use the same notation for the length (number of literals) of conjunctions and implicants.

Frequently, in proofs, we will need to show that a particular conjunction C is *not* an implicant of a Boolean function ϕ. We do this by exhibiting an assignment A that is consistent with C but for which $\phi(A) = 0$. Such an assignment is a *witness* that C is not an implicant of ϕ.

Familiarity with the complexity class $\Sigma_2^p = NP^{NP}$ is assumed, although to understand the proofs, one needs only to know that $QSAT_2$ is a complete problem for Σ_2^p (see [9]). An instance of $QSAT_2$ is given by $\langle X_1, X_2, \phi \rangle$, where X_1 and X_2 are disjoint sets of variables (the *existential* and *universal* variables, respectively) and ϕ is a DNF formula involving only the variables in X_1 and X_2. The instance is a positive instance if the following statement is true: $\exists X_1 \forall X_2 \phi$. Notice that this is equivalent to saying that there exists an implicant of ϕ consisting of only those variables in X_1.

3. MIN DNF is Σ_2^p-complete

In its most general form, the DNF minimization problem takes an input DNF and outputs an equivalent DNF formula that satisfies some minimality criteria. For the MIN DNF problem as we have defined it, minimality means the fewest occurrences of literals among equivalent DNF formulas. We obtain a different problem if, for example, we ask for an equivalent DNF formula with the fewest terms among equivalent DNF formulas. Our starting point for showing that MIN DNF is Σ_2^p-complete is an "easier" DNF minimization problem based on an alternative form of minimality.

First, observe that the DNF minimization problem can be weakened significantly by requiring only that the output be an equivalent *prime* DNF, one in which every term is a prime implicant. To find such an equivalent DNF, we can simply pare down each term of the input one literal at a time (querying an NP oracle at each step to determine if the new term is an implicant), until every term is a prime implicant. While finding an equivalent prime DNF is still far from easy – it is trivially co-NP-hard – it can be solved with a polynomial number of queries to an NP oracle, so it escapes the hardness of Σ_2^p.

However notice that in the above procedure, the overall length of the output DNF may depend on the order literals are removed from each term. If we require that the equivalent DNF is *term-wise minimal*, we require that each term is a *shortest* prime implicant that subsumes the corresponding term of the input formula. Term-wise minimality reduces the search space in comparison to length minimality: in general, a equivalent DNF with minimum length may include terms that do not subsume any term in the input DNF. The essence of term-wise minimality is captured by the following problem, which we show to be Σ_2^p-complete:

SHORTEST IMPLICANT CORE. Given a DNF formula $\phi = t_1 \vee t_2 \vee \ldots \vee t_n$ and an integer k, does ϕ have an implicant $C \supseteq t_n$ that contains k or fewer occurrences of literals?

Theorem 1 SHORTEST IMPLICANT CORE *is Σ_2^p-complete under logspace, many-one reductions.*

Proof: Let $\langle X_1, X_2, \phi \rangle$ be an instance of QSAT$_2$, and let $m = |X_1|$. There are three components to the reduction. First, we introduce a set S of new variables which will become our final term t_n, and construct a formula ϕ' from ϕ so that certain "valid" subsets of S correspond to partial assignments to the variables in X_1. Second, we build a formula f with the property that all shortest implicants of f that subsume S are "valid." Finally, we combine f and ϕ and add S as the final term.

We obtain ϕ' from ϕ as follows: introduce $2m$ new variables $a_1, a_2, \ldots a_m$ and $b_1, b_2, \ldots b_m$ and replace occurrences of the literal x_i with a_i and occurrences of $\overline{x_i}$ with b_i.

Claim 1 *The instance of QSAT$_2$ is a positive instance if and only if there exists an implicant of ϕ' consisting of exactly one literal from every (a_i, b_i) pair.*

Proof: ($\Rightarrow$) If the instance of QSAT$_2$ is a positive instance, then there exists a partial assignment A to the variables in X_1 that produces a tautology, that is, $\phi_A \equiv 1$. Let A' be the partial assignment mapping a_i to 1 if the literal x_i appears in A and b_i to 1 if the literal $\overline{x_i}$ appears in A. The formula $\phi'_{A'}$ includes every term in ϕ_A, plus additional terms. Therefore, $\phi_A \equiv 1$ implies $\phi'_{A'} \equiv 1$, so the conjunction corresponding to partial assignment A' is an implicant of ϕ', and it consists of exactly one literal from every (a_i, b_i) pair.

($\Leftarrow$) We show the contrapositive. Assume the instance of QSAT$_2$ is a negative instance, and let C be a conjunction consisting of exactly one literal from every (a_i, b_i) pair. We will show that C cannot be an implicant of ϕ'. We begin by constructing A', a partial assignment to the variables in X_1 defined as follows: for each i, if a_i appears in C, we include $a_i\overline{b_i}$ in A', and if b_i appears in C, we include $\overline{a_i}b_i$ in A'. Notice that there exists a partial assignment A to exactly the variables in X_1 such that $\phi'_{A'}$ is equivalent to ϕ_A. Since the instance of QSAT$_2$ is false, there must be an assignment B to the variables in X_2 such that $\phi_A(B) = 0$. Since $\phi'_{A'} \equiv \phi_A$ it follows that $\phi'_{A'}(B) = 0$, and the assignment given by $A'B$ is a complete assignment to ϕ' that makes ϕ' false, and is also consistent with C. Assignment $A'B$ is thus a witness that C is not an implicant of ϕ'. ∎

Now we construct formula f, for which all shortest implicants whose literals are subsets of the set $\{a_1, b_1, a_2, b_2, \ldots a_m, b_m\}$ consist of exactly one of every (a_i, b_i) pair. In the following, $z_1, z_2, \ldots z_{m-1}$ are new variables:

$$
\begin{array}{rcllcl}
f & = & a_1 z_1 & & \vee & b_1 z_1 \\
 & \vee & \overline{z_1} a_2 z_2 & & \vee & \overline{z_1} b_2 z_2 \\
 & \vee & \overline{z_2} a_3 z_3 & & \vee & \overline{z_2} b_3 z_3 \\
 & & \vdots & & & \vdots \\
 & \vee & \overline{z_{n-2}} a_{n-1} z_{n-1} & & \vee & \overline{z_{n-2}} b_{n-1} z_{n-1} \\
 & \vee & \overline{z_{n-1}} a_n & & \vee & \overline{z_{n-1}} b_n
\end{array}
$$

Claim 2 *Let C be a conjunction containing only literals from $\{a_1, b_1, a_2, b_2, \ldots a_m, b_m\}$. Conjunction C is an implicant of f if and only if C includes* at least *one literal from every (a_i, b_i) pair.*

Proof: ($\Rightarrow$) Suppose C excludes both a_i and b_i for some i. We can then construct an assignment A consistent with C, and for which $f(A) = 0$, as follows: A maps every variable not in C to 0, except z_j for all $j \geq i$, which it maps to 1. Assignment A is a witness that C is not an implicant of f.

($\Leftarrow$) Any conjunction that includes exactly one of each (a_i, b_i) pair is an implicant since it results from the consensus[1] of exactly one term of f from each row above. Conjunction C is just such a conjunction, possibly with additional literals. ∎

Finally, let $\phi'' = (f \wedge \phi') \vee a_1 b_1 a_2 b_2 \ldots a_m b_m$. When the parenthesized expression is multiplied out into DNF, the length remains polynomial in the length of the original input formula ϕ.

Claim 3 *Formula ϕ'' has an implicant $C \supseteq a_1 b_1 a_2 b_2 \ldots a_m b_m$ of length at most m if and only if the instance of QSAT$_2$ is a positive instance.*

Proof: ($\Rightarrow$) Suppose ϕ'' has an implicant $C \supseteq a_1 b_1 a_2 b_2 \ldots a_m b_m$ of length at most m. It must consist of at least one literal from every (a_i, b_i) pair, for if it does not, then by the proof of Claim 2 above, there is an assignment witnessing the fact that C is not an implicant of f *for which some a_i or b_i is mapped to 0.* For this assignment f is false and the final term of ϕ'' is false, so ϕ'' is false; hence C cannot be an implicant of ϕ'', contrary to our supposition.

[1] The *consensus* of two conjunctions xA and $\overline{x}B$, where x is a variable and A and B are conjunctions of literals, is AB, unless A and B are inconsistent, in which case the consensus is said to not exist. If xA and $\overline{x}B$ are implicants of a function ϕ, then their consensus, if it exists, is also an implicant of ϕ.

We now know that C contains at least one literal from every (a_i, b_i) pair, and since $|C| \leq m$, it must be that C contains exactly one literal from every (a_i, b_i) pair. Now suppose the instance of QSAT$_2$ is a negative instance. By the proof of Claim 1 above, there is an assignment witnessing the fact that C is not an implicant of ϕ' for which some a_i or b_i is mapped to 0. For this assignment, ϕ' is false and the final term of ϕ'' is false, so ϕ'' is false, and hence C cannot be an implicant of ϕ'', a contradiction. Therefore, the instance of QSAT$_2$ is a positive instance.

($\Leftarrow$) By Claim 1, if the instance of QSAT$_2$ is a positive instance, then there exists an implicant of ϕ' consisting of exactly one literal from every (a_i, b_i) pair. By Claim 2, a conjunction of this form is also an implicant of f, so it is an implicant of ϕ'', and it has length m as required. ∎

The instance of SHORTEST IMPLICANT CORE given by $\langle \phi'', m \rangle$ is therefore a positive instance if and only if the instance of QSAT$_2$ is a positive instance. This completes the reduction. ∎

Theorem 2 MIN DNF *is Σ_2^p-complete under logspace, many-one reductions.*

Proof: Let $\langle \phi = t_1 \vee t_2 \vee \ldots \vee t_n, k \rangle$ be an instance of SHORTEST IMPLICANT CORE. The general idea of the reduction is to force any minimum length DNF that is equivalent to ϕ to include term t_i for all $i < n$. The problem of finding a minimum equivalent DNF then becomes exactly the problem of finding the shortest implicant core of ϕ. We first need to modify ϕ to ensure that no term t_i for $i < n$ covers an assignment covered by t_n. Otherwise, if t_n is redundant (i.e. it can be removed from ϕ without changing the function computed by ϕ), the DNF $t_1 \vee t_2 \vee \ldots \vee t_{n-1}$ by itself will constitute an equivalent DNF, and we will obtain no useful information about the shortest implicant that subsumes t_n. So, let $\phi' = \phi \wedge \neg t_n$. When ϕ' is multiplied out into DNF, its length remains polynomial in the length of ϕ. Let $s_1, s_2, \ldots s_m$ be the terms of ϕ' when it is multiplied out into DNF, and let $m' = \max(m, k+1)$. We define the following formula:

$$\phi'' = t_n w_1 w_2 \ldots w_{m'} \vee \bigvee_{i=1}^{m} s'_i,$$
$$\text{where } s'_i = s_i w_1 w_2 \ldots w_{i-1}\ w_{i+1} \ldots w_{m'}$$

Let $k' = k + m' + \sum_{i=1}^{m} |s'_i|$. Our intended instance of MIN DNF is $\langle \phi'', k' \rangle$. We begin with the easy half of the reduction:

Claim 1 *If C is an implicant of ϕ such that $C \supseteq t_n$, then*

$$\phi'' \equiv C w_1 w_2 \ldots w_{m'} \vee \bigvee_{i=1}^{m} s'_i.$$

Proof: As noted in the introduction, an equivalent DNF may be obtained by paring down each term in the original DNF, as long as each pared-down term remains an implicant. Therefore it is sufficient to show that $C w_1 w_2 \ldots w_{m'}$ is an implicant of ϕ''. Notice that $\phi \equiv \phi' \vee t_n \equiv \phi''_B$, where B is the partial assignment that maps $w_1, w_2 \ldots w_{m'}$ to 1 and all other variables to $*$. Since C is an implicant of ϕ, it is also an implicant of ϕ''_B, and therefore $C w_1 w_2 \ldots w_{m'}$ is an implicant of ϕ''. ∎

If the implicant C in the previous claim has length $\leq k$, then the equivalent DNF given in the claim has length $\leq k'$; hence the instance of MIN DNF is a positive instance if the instance of SHORTEST IMPLICANT CORE is a positive instance.

We now need to show that the existence of an equivalent DNF of length at most k' implies that the instance of SHORTEST IMPLICANT CORE is a positive instance. We begin by showing that for all i, the term s'_i is required (in a loose sense) in any equivalent DNF.

Claim 2 *Let f be a DNF equivalent to ϕ''. Then for all i, (1) f includes a term s''_i such that $s'_i \supseteq s''_i$, (2) the collective length of these terms is at least $\sum_{i=1}^{m} |s'_i|$, and (3) none of these terms covers any assignment covered by $t_n w_1 w_2 \ldots w_{m'}$.*

Proof: Let A_i be an assignment consistent with s'_i and with w_i mapped to 0, and let s''_i be a term of f that covers A_i. If s''_i excludes any w_j for $j \neq i$, then s''_i covers an assignment in which w_j and w_i are mapped to 0, but f covers no such assignments. If s''_i excludes any literal from s_i, then s''_i covers an assignment inconsistent with s'_i, but with w_i mapped to 0. This is a contradiction, since the only assignments with w_i mapped to 0 that make ϕ'' true are consistent with s'_i. So $s'_i \supseteq s''_i$, which implies that $|s'_i| \leq |s''_i|$. Since s''_i contains the literal w_j for all $j \neq i$, it cannot cover A_j for $j \neq i$. Since the A_i are all distinct, the s''_i are distinct, so their collective length is at least $\sum_{i=1}^{m} |s''_i| \geq \sum_{i=1}^{m} |s'_i|$. Finally, no s''_i covers any assignment covered by $t_n w_1 w_2 \ldots w_{m'}$, since $s'_i \supseteq s''_i$ and our initial modification to ϕ ensures that s'_i and $t_n w_1 w_2 \ldots w_{m'}$ are disjoint. ∎

In addition to the s''_i terms, an equivalent DNF must include terms that collectively cover all the assignments covered by $t_n w_1 w_2 \ldots w_{m'}$. Assuming that the equivalent DNF has length at most k', these additional terms must have total length of at most $k' - \sum_{i=1}^{m} |s'_i| \leq k + m'$. Any implicant of ϕ'' that covers an assignment covered by $t_n w_1 w_2 \ldots w_{m'}$ must include the literals $w_1 w_2 \ldots w_{m'}$; otherwise it would cover an assignment consistent with t_n but with some w_i set to false, and ϕ'' covers no such assignments (recall our initial modification to ϕ). Therefore, each of the additional terms has length at least m', and since $k + m' < 2m'$, there must be only one such additional term. This term, C, has length at most $k + m'$, and $C \supseteq t_n w_1 w_2 \dots w_{m'}$. As noted, $\phi''_B \equiv \phi$, where B is the partial assignment that maps $w_1, w_2, \cdots, w_{m'}$ to 1 and all other variables to $*$. Hence, letting $C' = C - w_1 w_2 \dots w_{m'}$, we get an implicant of ϕ, with $C' \supseteq t_n$ and $|C'| \leq k$. Thus the instance of SHORTEST IMPLICANT CORE is a positive instance if there exists an equivalent DNF of length at most k'. This completes the reduction from SHORTEST IMPLICANT CORE to MIN DNF. ■

Note that this proof also shows that the following "easier" DNF minimization problem, based on term-wise minimality, is Σ_2^p-complete: given a DNF formula $\phi = t_1 \vee t_2 \vee \dots t_n$, and an integer k, is there a DNF formula $\phi' = t'_1 \vee t'_2 \vee \dots \vee t'_n$ equivalent to ϕ that contains k or fewer occurrences of literals, *and with the property that* $\forall i \;\; t'_i \supseteq t_i$?

4. Complexity of SHORTEST IMPLICANT[DNF]

The SHORTEST IMPLICANT CORE problem is a restricted version of a more natural problem which simply asks for the shortest implicant of a DNF formula. One might reasonably expect that finding shortest implicants of a DNF formula is also Σ_2^p-complete. However, in contrast to MIN DNF, we find that SHORTEST IMPLICANT[DNF] is not Σ_2^p-complete unless Σ_2^p collapses to the class NP$[\log^2 n]^{\text{NP}}$.

The class NP$[\log^2 n]$ is a subclass of NP with $\log^2 n$-limited nondeterminism: the usual polynomial-time-bounded nondeterministic Turing machine is permitted only $O(\log^2 n)$ non-deterministic steps, after which the remainder of the computation must be deterministic. This class (also called β_2) is explored in [7], [5] and [10]. The relevant class for this section is NP$[\log^2 n]$ with an NP oracle, which lies between P$^{\text{NP}}$ and Σ_2^p.

Theorem 3 SHORTEST IMPLICANT[DNF]$\in$ NP$[\log^2 n]^{\text{NP}}$.

Proof: Let $\phi = t_1 \vee t_2 \vee \dots \vee t_n$ be the input DNF formula on m variables. We show that *all* shortest implicants of ϕ are "close" to a term of ϕ in the following sense: they can be derived from a term of ϕ with at most $\log n$ insertions and at most $\log n$ deletions of literals.

Let C be a shortest implicant of ϕ, and let $t_1, t_2, \dots, t_{n'}$ be only those terms of ϕ that are consistent with C. Now, for all i, define $t'_i = t_i \wedge C$; this ensures that $C \supseteq t'_i$, for all i. We first argue that there must exist an i such that $|t'_i| \leq |C| + \log n'$. Suppose there were not. Then each t'_i has length greater than $|C| + \log n'$, and the maximum number of assignments covered by the terms $t'_1, t'_2, \dots t'_{n'}$ is therefore strictly less than $n' 2^{m-|C|-\log n'} = 2^{m-|C|}$. This is a contradiction, since it implies that the terms consistent with C do not cover all of the assignments covered by C. So there must be an i such that $|t'_i| \leq |C| + \log n'$.

Recall that t'_i was obtained from t_i by adding literals. We cannot have added more than $\log n'$ literals, unless $|t_i| < |C|$, which contradicts C being a shortest implicant of ϕ. Since $n' \leq n$, we have shown that C results from t_i with at most $\log n$ insertions and at most $\log n$ deletions of literals.

The NP$[\log^2 n]^{\text{NP}}$ algorithm is straightforward. In the nondeterministic phase, guess the following: (1) an index $1 \leq i \leq n$, (2) a subset I of the set of the $2m$ literals to insert, such that $|I| \leq \log n$, and (3) a subset D of the literals to delete, such that $|D| \leq \log n$. This requires $O(\log^2 n)$ nondeterministic steps. The deterministic phase simply constructs the conjunction $C = (t_i \cup I) - D$, rejects trivially if C contains both a variable and its complement or has length greater than k, and otherwise accepts if and only if a query to the NP oracle indicates C is an implicant of ϕ. ■

Corollary 4 SHORTEST IMPLICANT[DNF] *is not* Σ_2^p-*complete unless* $\Sigma_2^p =$NP$[\log^2 n]^{\text{NP}}$.

5. Shortest implicants of formulae and circuits

Although SHORTEST IMPLICANT[DNF] is not likely to be Σ_2^p-complete, if we allow the input to be a de Morgan formula, the resulting problem is Σ_2^p-complete, and Σ_2^p-hard to approximate to within an $n^{1/2-\epsilon}$ factor. The same method can be used to show that SHORTEST IMPLICANT[CIRCUIT] is Σ_2^p-hard to approximate to within an $n^{1-\epsilon}$ factor.

We first give a simple reduction from QSAT$_2$ and then describe a general way to introduce a gap between positive and negative instances which leads to the inapproximability results.

Theorem 5 SHORTEST IMPLICANT[FORMULA] *and* SHORTEST IMPLICANT[CIRCUIT] *are* Σ_2^p-*complete under logspace, many-one reductions.*

Proof: We give the reduction for SHORTEST IMPLICANT[FORMULA], which is a special case of SHORTEST IMPLICANT[CIRCUIT].

Let $\langle X_1, X_2, \phi \rangle$ be an instance of QSAT$_2$, and let $X_1 = \{x_1, x_2, \dots x_m\}$. As noted in the introduction, the instance is true if and only if there exists an implicant of ϕ consisting of only variables in X_1. We modify ϕ so that *all* implicants must include the variables in X_1 and m additional variables. In the resulting formula, an implicant consisting of *exactly* these $2m$ variables exists if and only if there is an implicant of ϕ consisting of only variables in X_1.

We introduce new variables $w_1, w_2 \dots w_m$ and define ϕ' as follows:

$$\phi' \quad = \quad \phi \wedge \bigwedge_{i=1}^{m} (x_i \oplus w_i)$$

$$= \quad \phi \wedge \bigwedge_{i=1}^{m} (x_i\overline{w_i} \vee \overline{x_i}w_i)$$

We claim that ϕ' has an implicant of length at most $2m$ if and only if the instance of QSAT_2 is true.

($\Leftarrow$) If the instance is true, then there is an implicant C of ϕ consisting of exactly the variables in X_1. Let A be the corresponding partial assignment to ϕ, and define a partial assignment A' to ϕ' as follows: A' coincides with A and additionally maps w_i to 1 if x_i is mapped to 0 by A, and maps w_i to 0 if x_i is mapped to 1 by A. By construction $\phi'_{A'} \equiv \phi_A$, and since C is an implicant of ϕ, $\phi_A \equiv 1$. Therefore the conjunction corresponding to A' is an implicant of ϕ'. This implicant has length $2m$ as required.

($\Rightarrow$) Let C' be an implicant of ϕ', and let A' be the corresponding partial assignment. For all i, A' must either map x_i to 1 and w_i to 0, or x_i to 0 and w_i to 1; otherwise there is an assignment consistent with A' in which $(x_i \oplus w_i)$ is false for some i, contradicting C' being an implicant of ϕ'. Since $|C'| \leq 2m$, A' must map *exactly* the variables in X_1 and the w_i to 0 or 1, and the other variables to $*$. Let A be A' restricted to the variables in ϕ. We have that $\phi_A \equiv \phi'_{A'}$, and $\phi'_{A'} \equiv 1$, so there exists an implicant of ϕ corresponding to A that includes exactly the variables in X_1. Hence the instance of QSAT_2 is true. ■

Because the shortest implicant problem is an optimization problem, it is natural to ask how well it can be approximated. We introduce a simple method for obtaining "gap"-type reductions for this problem that yields inapproximability to within an $n^{1/2-\epsilon}$ factor for the formula version and an $n^{1-\epsilon}$ factor for the circuit version.

An examination of the proof of Theorem 5 reveals that *all* prime implicants of ϕ' contain at least one variable from X_2 if and only if the instance is a negative instance. In the following proofs, we use substitutions of formulae or circuits for variables in X_2 to "weight" these variables. The result is that all prime implicants (and hence, all shortest implicants) of a negative instance are long, while a positive instance has an implicant that avoids the weighted variables and remains short. The substituted functions *and their negations* must have long shortest implicants for the reductions to work.

We first recall the following facts about the parity function:

Observation 1 *The parity function on k variables has shortest implicants of length k, and its negation has shortest implicants of length k. Parity on k variables can be computed by de Morgan formulae of size $O(k^2)$ and $(\wedge, \vee, \neg)$-circuits of size $O(k)$.*

In the next two theorems, n is the length of the input formula or circuit as we have defined it (number of occurrences of literals or number of gates). However since ϵ is arbitrary, the results hold when n is the more precise measure of the input length that counts bits in the input representation, which differs from our definition of length by no more than a logarithmic factor.

Theorem 6 *Approximating* SHORTEST IMPLICANT[FORMULA] *to within an $n^{1/2-\epsilon}$ factor is Σ_2^p-hard, for any $\epsilon > 0$.*

Proof: Recall from the proof of Theorem 5 that we transformed an instance of QSAT_2 given by $\langle X_1, X_2, \phi\rangle$ into the following formula:

$$\phi' = \phi \wedge \bigwedge_{i=1}^{m} (x_i\overline{w_i} \vee \overline{x_i}w_i)\ ,$$

where $x_1, x_2, \ldots x_m$ are the variables in X_1. Let $y_1, y_2, \ldots y_p$ be the variables in X_2, and let $N = |\phi'|$. For each $1 \leq i \leq p$, let f_i be a parity formula on N^c new variables. The exponent c is a constant that will depend on ϵ. We obtain ϕ'', the "weighted" version of ϕ', by substituting f_i for the literal y_i, and $\neg f_i$ for the literal $\overline{y_i}$, for all i.

We need to show that the largest answer an $n^{1/2-\epsilon}$-approximate algorithm might give on a positive instance is strictly smaller than the smallest answer it might give on a negative instance. This is shown in the next two claims:

Claim 1 *The largest answer an $n^{1/2-\epsilon}$-approximate algorithm might give on input ϕ'' if the $QSAT_2$ instance is a positive instance is $2m \cdot O(N^{(2c+1)(1/2-\epsilon)})$.*

Proof: From the proof of Theorem 5, we know that a positive instance of QSAT_2 implies that ϕ' has a shortest implicant C of length $2m$ consisting of exactly the variables $x_1, x_2, \ldots x_m, w_1, w_2, \ldots w_m$. We claim that this is also an implicant of ϕ''. Suppose it were not. Then there is an assignment A witnessing the fact that C is not an implicant of ϕ''. We can then construct a witness A' that C is not an implicant of ϕ' as follows: A' maps each x_i and w_i to the same value as A does, and maps each y_i to the value induced by A at the root of the corresponding substituted parity formula f_i. This is a contradiction, so C must be an implicant of ϕ''.

Each substituted parity formula has length $O(N^{2c})$, so $|\phi''| \leq N \cdot O(N^{2c})$. Since ϕ'' has an implicant of length at most $2m$, the largest answer an $n^{1/2-\epsilon}$-approximate algorithm may give on input ϕ'' is $2m|\phi''|^{1/2-\epsilon} \leq 2m \cdot O(N^{(2c+1)(1/2-\epsilon)})$, as required. ■

Claim 2 *The smallest answer an $n^{1/2-\epsilon}$-approximate algorithm might give on input ϕ'' if the $QSAT_2$ instance is a negative instance is $2m + N^c$.*

Proof: Suppose that there exists an implicant C of ϕ'' with length less than $2m + N^c$. If C excludes x_i or w_i for any

i, then there is an assignment consistent with C that makes ϕ'' false, as in the proof of Theorem 5, so C must include at least these $2m$ variables. Since $|C| < 2m + N^c$, C cannot include an implicant of f_i or $\neg f_i$, for all i. Then for all i, we can choose partial assignments P_i and N_i to ϕ'' such that: (1) P_i is consistent with C and makes f_i take on value 1, (2) N_i is consistent with C and makes f_i take on value 0, and (3) P_i and N_i map all variables other than those in the formula f_i to $*$.

For any assignment A to ϕ' we define a corresponding assignment $\psi(A)$ to ϕ'' as follows: $\psi(A)$ coincides with A on variables $x_1, x_2 \ldots x_m, w_1, w_2 \ldots w_m$, and if A maps y_i to 0, $\psi(A)$ includes N_i; otherwise $\psi(A)$ includes P_i. It is easy to see that $\phi'(A) = \phi''(\psi(A))$.

Let A be an assignment to ϕ' that coincides with C on variables $x_1, x_2 \ldots x_m, w_1, w_2 \ldots w_m$, and for which $\phi'(A) = 0$. Such an assignment must exist because the instance of QSAT$_2$ is a negative instance. Then $\phi''(\psi(A)) = 0$, and notice that $\psi(A)$ is consistent with C, by our choice of P_i and N_i above. Assignment $\psi(A)$ is a witness that C is not an implicant of ϕ'', a contradiction. So there can be no implicant of ϕ'' shorter that $2m + N^c$ if the QSAT$_2$ instance is a negative instance. ■

To complete the proof, we note that $2m \le N$, and so the following inequality holds for sufficiently large N and $c \ge (3/4)\epsilon^{-1}$:

$$2m \cdot O(N^{(2c+1)(1/2-\epsilon)}) < 2m + N^c.$$

Therefore, given any $n^{1/2-\epsilon}$-approximate algorithm for SHORTEST IMPLICANT[FORMULA], we can construct an instance for which that algorithm distinguishes between positive and negative instances. ■

Using the same method, we obtain an even stronger inapproximability result for the version of the problem in which the input is a circuit.

Theorem 7 *Approximating* SHORTEST IMPLICANT[CIRCUIT] *to within an $n^{1-\epsilon}$ factor is Σ_2^p-hard, for any $\epsilon > 0$.*

Proof: Analogous to the previous proof. We start with the same formula ϕ' but obtain *circuit* ϕ'' by substituting parity *circuits* on N^c variables for each y_i. The size of ϕ'' is at most $N \cdot O(N)$. The largest answer an $n^{1-\epsilon}$-approximate algorithm might give on a positive instance is $2m \cdot O(N^{(c+1)(1-\epsilon)})$; the smallest answer possible on a negative instance is $2m + N^c$. For sufficiently large N and $c \ge 2\epsilon^{-1}$ the following inequality holds:

$$2m \cdot O(N^{(c+1)(1-\epsilon)}) < 2m + N^c.$$

This completes the reduction. ■

6. Conclusions and Open Problems

By simply negating formulas, our results imply Σ_2^p-completeness for MIN CNF. Similarly, the results for SHORTEST IMPLICANT[DNF] carry over to the Shortest Impli*cate* problem for a CNF formula (an *implicate* of a function f is a disjunction implied by f), and both the completeness and inapproximability results for SHORTEST IMPLICANT[FORMULA] and SHORTEST IMPLICANT[CIRCUIT] carry over to the problem of finding shortest implicates of formulae and circuits, respectively.

Several related open problems remain. It would be interesting to precisely classify the complexity of SHORTEST IMPLICANT[DNF] (i.e. show completeness for some complexity class). The appropriate class may be NP$[\log^2 n]^{\text{NP}}$, or it may be a class analogous to the LOGNP class defined in [10]. Any approximability or inapproximability results for MIN DNF might yield insight into the best ways to attack this real-world problem. Similar results for some of the "simpler" optimization problems involving shortest implicants might be a useful starting point. Of course it is yet to be shown that the Minimum Equivalent Expression and Minimum Equivalent Circuit problems are Σ_2^p-complete. Our result for MIN DNF uses the structure of a DNF formula heavily, and it seems that it is not readily extendible to these other logic minimization problems.

A significant amount of effort is spent designing and implementing practical methods for prime implicant generation and logic minimization, and the problems are well-suited to complexity analysis. Complexity theory should have a lot to offer to guide the design of algorithms for these important problems.

We would like to thank Christos Papadimitriou for suggesting this problem, and many useful discussions.

References

[1] M. Agrawal and T. Thierauf. The boolean isomorphism problem. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS '96)*, pages 422–30, 1996.

[2] B. Borchert and D. Ranjan. The circuit subfunction relations are Σ_2^p-complete. Technical Report MPI-I-93-121, MPI Saarbrücken, 1993.

[3] O. Coudert. Two-level logic minimization: an overview. *Integration, the VLSI journal*, 17(2):97–140, 1994.

[4] S. Devadas, A. Ghosh, and K. Keutzer. *Logic Synthesis*. McGraw-Hill, Inc., 1994.

[5] J. Diaz and J. Toran. Classes of bounded nondeterminism. *Mathematical Systems Theory*, 23(1):21–32, 1990.

[6] E. Hemaspaandra and G. Wechsung. The minimization problem for boolean formulas. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS '97)*, pages 575–84, 1997.

[7] C. M. R. Kintaal and P. C. Fischer. Refining nondeterminism in relativized polynomial-time bounded computations. *SIAM Journal on Computing*, 9(1):46–53, 1980.

[8] E. J. McCluskey. Minimization of boolean functions. *Bell Systems Technical Journal*, 35(5):1417–44, 1956.

[9] C. H. Papadimitriou. *Computational Complexity*. Addison Wesley, 1994.

[10] C. H. Papadimitriou and M. Yannakakis. On limited nondeterminism and the complexity of the V–C dimension. *Journal of Computer and System Sciences*, 53(2):161–70, 1996.

[11] W. V. Quine. The problem of simplifying truth functions. *American Mathematics Monthly*, 59(8):521–531, 1952.

[12] L. J. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[13] T. Strzemecki. Polynomial-time algorithms for generation of prime implicants. *Journal of Complexity*, 8(1):37–63, 1992.

[14] K. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51(1–2):53–80, 1987.