On Testing Convexity of 2DNF

Josué A. Ruiz[1], Amir Masoumzadeh[1], Paliath Narendran[1], and Padmavathi Iyer[2]

[1] University at Albany–SUNY (USA),
e-mail: {jaruiz, amasoumzadeh, pnarendran}@albany.edu
[2] Drury University (USA)
e-mail: riyer@drury.edu

Abstract

Testing the convexity of boolean formulae has applications in problems such as the convertibility of access control policies. In this paper, we report on our ongoing work on developing a polynomial algorithm for testing the convexity of the set of minterms of a disjunctive normal form formula where every term has exactly two literals in it (2DNF).

1 Introduction and Preliminaries

Our goal in this paper is to develop an algorithm for testing the convexity of the set of minterms of a boolean formula in disjunctive normal form where every term has exactly two literals in it (2DNF). The motivation for this problem comes from the convertibility problem for rule-based access control policies [4, 5].

Let $X = \{x_1, \ldots, x_n\}$ be a set of Boolean variables. Expressions, terms (products or conjuncts), and minterms are defined as usual [3]. Given a set of Boolean variables X, we denote the set of all possible minterms as $\mathcal{M}_X$. For a Boolean expression Ψ, let $\mu(\Psi)$ be the set of its minterms. Note that each minterm of an expression can also be viewed as a (representation of a) satisfying assignment for that expression.

We define a partial order $\leq$ on bit strings (of the same length) as follows: first define the order on single bits as $0 \leq 0,\ 0 \leq 1$ and $1 \leq 1$. This is extended to bit strings X and Y as

- $X \leq Y$ iff $X[j] \leq Y[j]$ for all j and
- $X < Y$ iff $X \leq Y$ and $X \neq Y$.

Definition 1.1. *A set of minterms M is* convex *if and only if for every pair $m_1 \leq m_2 \in M$:*

$$\{\ m \mid m_1 \leq m \leq m_2\} \ \subseteq \ M.$$

Definition 1.2. *Let M be a set of minterms. Its* upward closure *$M^{\uparrow}$ is defined as*

$$M^{\uparrow} = \{\ u \mid \exists m \in M : m \leq u\}.$$

Definition 1.3. *A set of minterms M is* upward closed *if and only if $M = M^{\uparrow}$.*

Definition 1.4. *For a term t, the product of its positive literals is denoted by* $\Pi(t)$ *and the product of its negative literals is denoted by* $\mathcal{N}(t)$*.*

If there are no positive literals in a term t, then $\Pi(t) = 1$. Similarly, if there are no negative literals in a term t, then $\mathcal{N}(t) = 1$.

Definition 1.5. *Given two product terms t_1 and t_2, their* separator*, denoted as* $\mathrm{sep}(t_1, t_2)$*, is* $\Pi(t_2)\mathcal{N}(t_1)$*, i.e., the conjunction of the positive literals of t_2 and the negative literals of t_1.*

For instance, if $t_1 = x_1\overline{x_2}$ and $t_2 = x_3\overline{x_4}$ then $sep(t_1, t_2) = \overline{x_2}x_3$.

In an earlier paper [5], we showed that:

Lemma 1.1. *The set of minterms of a boolean expression Φ in DNF is convex if and only if every separator is an implicant of Φ.*

Another characterization of convexity is as follows.

Lemma 1.2. *The set of minterms of a boolean expression Φ in DNF is convex if and only if there exist* positive *DNF expressions Ψ_1 and Ψ_2 such that*

$$\Phi \equiv \Psi_1 \wedge \neg\Psi_2$$

Note that a boolean expression in DNF is said to be in *positive* DNF form if and only if no negated literals appear in it.

Our goal in this paper is to design an efficient algorithm for checking the convexity of an expression in DNF. By Lemma 1.2 this problem can be formulated as a matching problem as follows:

Instance: A formula Φ in DNF.

Question: Are there *positive* DNF formulae $\mathcal{A}$ and $\mathcal{B}$ such that $\Phi \equiv \mathcal{A} \wedge \neg\mathcal{B}$?

We consider in this paper a restricted version of this problem where every product term in the DNF formula *has exactly two literals*. There are three main cases:

(a) There exists an all-positive term and an all-negative term.

(b) Every term is *mixed*, i.e., every term has a negated literal and an unnegated literal. Thus we have neither all-positive terms nor all-negative terms. We refer to such formulae as "all-mixed 2DNF."

(c) There is an all-positive term, but no all-negative term. (The dual case where there is an all-negative term but no all-positive terms is similar.)

Case (a) is the most straightforward. The set of minterms $\mu(t)$ of an all-positive term t contains the highest minterm. Similarly if t is all-negative, then $\mu(t)$ contains the lowest minterm.

Hence, the set of minterms of such a formula is convex if and only if the formula is valid [4]. The validity of such formulae can be checked in linear time [1].

In Section 2, we discuss Case (b), i.e., where every product term is of the form $x_i\overline{x_j}$. We derive a graph-theoretic characterization of the implication graph of the *negation* of such formulae (which clearly will be in CNF). This leads to a linear algorithm for testing convexity. We also briefly discuss a quadratic-time algorithm for Case (c) in Section 3, and conclude the paper in Section 4.

2 Linear-Time Algorithm For All-Mixed Case

Let Φ be an all-mixed 2DNF formula and let $\neg\Phi$ be its complement in CNF. Let $IG(\neg\Phi)$ be the implication graph of $\neg\Phi$ [1]:

- For each variable x_i, we add two nodes named x_i and $\overline{x}_i$.
- For each clause $u \vee v$ of $\neg\Phi$, we add edges $\overline{u} \to v$ and $\overline{v} \to u$.

Note that every clause in $\neg\Phi$ is of the form $\overline{x_i} \ \vee \ x_j$. Hence we only keep the part of the graph with nodes with positive literals since there are no edges between nodes with positive literals and nodes with negative literals.

Lemma 2.1. *Let Φ be an all-mixed 2DNF formula and let x_1 and x_2 be two distinct variables. Then $x_1\overline{x_2}$ is an implicant of Φ if and only if there is a path from x_1 to x_2 in $IG(\neg\Phi)$.*

Lemma 2.2. *Let Φ be an all-mixed 2DNF formula. Then Φ is convex if and only if the following holds for all distinct variables x_1, x_2, x_3, x_4:*

> *if $x_1\overline{x_2}$ and $x_3\overline{x_4}$ are terms in Φ then there are paths in $IG(\neg\Phi)$ from x_1 to x_4 and from x_3 to x_2.*

Proof. Follows from Lemmas 1.1 and 2.1, since $x_1\overline{x_4}$ and $\overline{x_2}x_3$ are separators. □

Since the implication graph may be cyclic, we will also need to consider the *component graph* $CIG(\neg\Phi)$ of the implication graph. This is obtained by coalescing all nodes in a strongly-connected component (SCC) into one node. This component graph can be constructed in time linear in the size of the original digraph [2].

Lemma 2.3. *Let Φ be an all-mixed DNF formula. Then Φ is convex* only if *the following holds for all distinct variables x_1, x_2, x_3, x_4:*

> *If $x_1\overline{x_2} \ \vee \ x_2\overline{x_3} \ \vee \ x_3\overline{x_4}$ is a subexpression of Φ then x_2 and x_3 belong to the same strongly-connected component in $CIG(\neg\Phi)$.*

Proof. There must be a path in $IG(\neg\Phi)$ from x_3 to x_2 by Lemma 2.2. $\square$

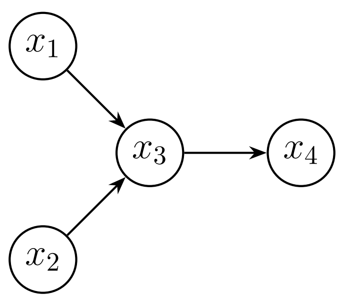
Corollary 2.3.1. *Let Φ be an all-mixed DNF formula. If Φ is convex, then $CIG(\neg\Phi)$ cannot contain a path of 3 edges.*

We now have to consider two separate cases:

1. All paths in $CIG(\neg\Phi)$ are of length 1.
2. There is a path of length 2 in $CIG(\neg\Phi)$.

(There is also the case where $CIG(\neg\Phi)$ has exactly one node, but that case is taken care of by the definition of radial dags given below.)

We call a dag G *radial* if there is a unique node v such that there is an edge from every source node to v, there is an edge to every sink node from v, and every node other than v is either a source node or a sink node. In other words, the set of nodes V can be partitioned into 3 disjoint subsets (V_1, $\{v\}$, V_2) such that V_1 is the set of source nodes, V_2 is the set of sink nodes, every node in V_1 is connected to v and v is connected to every node in V_2. (See Figure 1: Nodes x_1 and x_2 are source nodes, x_4 is the only sink node and x_3 is the "middle node.")



A *bipartite dag* is a dag where the set of nodes V can be partitioned into two disjoint subsets V_1 and V_2, such that every edge is from a node in V_1 to a node in V_2. In other words, every node in V_1 is a source node, and every node in V_2 is a sink node. A *complete bipartite dag* is a dag where the set of nodes V can be partitioned into two disjoint subsets V_1 and V_2, such that every node in V_1 has an edge to every node in V_2.

Lemma 2.4. *Let Φ be an all-mixed DNF formula such that $CIG(\neg\Phi)$ has no edges at all. Then Φ is convex if and only if $CIG(\neg\Phi)$ has only one node.*

Lemma 2.5. *Let Φ be an all-mixed DNF formula such that all paths in $CIG(\neg\Phi)$ are of length 1. Then Φ is convex if and only if $CIG(\neg\Phi)$ a complete bipartite dag.*

Lemma 2.6. *Let Φ be an all-mixed DNF formula such that there is a path of length 2 in $CIG(\neg\Phi)$. Then Φ is convex if and only if $CIG(\neg\Phi)$ is a radial dag.*

Theorem 1. *An all-mixed DNF formula Φ is convex if and only if $CIG(\neg\Phi)$ is either a complete bipartite dag or a radial dag.*

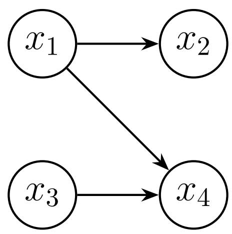
Proof. Follows from Corollary 2.3.1 and lemmas 2.5 and 2.6. □

The above theorem provides a linear-time algorithm for the all-mixed case.

Example 2.1. *Consider the DNF expression* $\Phi = x_1\overline{x_2} \vee \overline{x_1}x_2$. $CIG(\neg\Phi)$ *has exactly one node* $\{1, 2\}$. *Thus this expression is convex.*

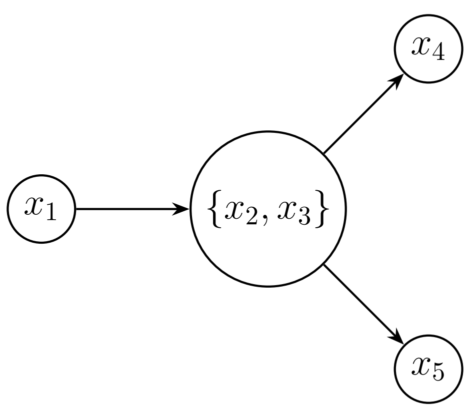
Example 2.2. *Consider the DNF expression* $\Phi = x_1\overline{x_2} \vee \overline{x_1}x_2 \vee x_3\overline{x_4} \vee \overline{x_3}x_4$. $CIG(\neg\Phi)$ *has two nodes but no edges. This formula is not convex because* 1100 *is not a minterm of* Φ, *whereas* 1110 *and* 1000 *are.*

Example 2.3. *Consider the DNF expression* $\Phi = x_1\overline{x_2} \vee x_1\overline{x_4} \vee x_3\overline{x_4}$. *Then the* $CIG(\neg\Phi)$ *is:*



Since the graph is not a complete bipartite dag, *according to Lemma 2.5,* Φ *is not convex. Note that* $\overline{x_2}x_3$, *which is a separator, is not an implicant.*

Example 2.4. *Consider the DNF expression* $\Phi = x_1\overline{x_2} \vee x_2\overline{x_3} \vee \overline{x_2}x_3 \vee x_2\overline{x_4} \vee x_3\overline{x_5}$. *Then* $CIG(\neg\Phi)$ *is as follows:*



Note that this dag is radial. *The given expression is convex.*

3 Quadratic-Time Algorithm For Case with An All-Positive Term

In this case it can be shown that the formula is convex if and only if it is upward-closed.

Lemma 3.1. *A formula Φ in DNF is upward-closed if and only if it is convex and contains an all-positive term.*

Lemma 3.2. *A formula Φ in DNF is upward-closed if and only if for every term t in Φ, $\Pi(t)$ is an implicant of Φ.*

Proof. This follows from Lemma 1.1: if Φ contains an all-positive term, then from every mixed term t we can get $\Pi(t)$ as a separator. □

A quadratic algorithm can be derived fairly easily since all we need to do is to check for every mixed term $u\overline{v}$ whether u is an implicant of the formula.

4 Conclusion and Future Work

In this paper, we investigated the problem of testing the convexity of DNF formula where every term has exactly two literals, splitting the problem into three main cases. We showed that the problem is easiest when both an all-positive and an all-negative term exist, since in that case convexity reduces to validity. We provided a linear-time algorithm to solve the second case, where every term is mixed, based on a characterization of the implication graph of the negation of the formula. We also showed that the third case, where the formula contains either an all-positive term or an all-negative term (but not both), can be solved in polynomial time. As part of future work we plan to derive another graph-theoretic characterization of the last case.

References

[1] Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A Linear-Time Algorithm for Testing the Truth of Certain Quantified Boolean Formulas. *Inf. Process. Letters*, 8(3):121–123, 1979.

[2] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, 2nd edition, 2001.

[3] Yves Crama and Peter L. Hammer. *Boolean Functions - Theory, Algorithms, and Applications*, volume 142 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2011.

[4] Amirreza Masoumzadeh, Paliath Narendran, and Padmavathi Iyer. Towards a Theory for Semantics and Expressiveness Analysis of Rule-Based Access Control Models. In *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*, SACMAT '21, pages 33–43. Association for Computing Machinery, 2021.

[5] Josué A. Ruiz, Amirreza Masoumzadeh, Paliath Narendran, and Padmavathi Iyer. Converting Rule-Based Access Control Policies: From Complemented Conditions to Deny Rules. In *Proceedings of the 29th ACM Symposium on Access Control Models and Technologies (to appear)*, 2024. Preprint Available at https://www.cs.albany.edu/~amir/papers/ruiz2024sacmat.pdf.