On Satisfiability of Partially Defined Double and Bidual Horn Functions*

(extended abstract)

Thomas Eiter† Toshihide Ibaraki‡ Kazuhisa Makino‡

1 Introduction

A partially defined Boolean function (pdBF) is a natural generalization of the familiar concept of Boolean function, by allowing that the function value on some vectors is unknown [5]. Such functions have many applications in computer science, including representation of incomplete information about cause–effect relationships [5], switching circuit design, and machine learning.

Formally, a pdBF is a pair (T, F) of sets T and F of true and false vectors $v \in \{0,1\}^n$, respectively, where $T \cap F = \emptyset$. Clearly, each pdBF can be completed to a total Boolean function (which has $T \cup F = \{0,1\}^n$) by arbitrarily assigning function values to the vectors in $\{0,1\}^n \setminus (T \cup F)$.

In general, however, one is interested to know whether the pdBF can be completed to a total Boolean function from a particular class of Boolean functions; this is known as the *extension problem*, and corresponds in a sense to the satisfiability problem of Boolean formulas.

The extension problem and variants thereof have been investigated for a number of classes of Boolean functions [5, 3, 2]. Among these classes are Horn functions; a function is *Horn*, if it can be represented by a DNF in which each term contains at most one negative literal. Denote for any Boolean function f by $T(f)$ and $F(f)$ the sets of its true and false vectors of f, i.e., $T(f) = \{v \mid f(v) = 1\}$ and $F(f) = \{w \mid f(w) = 0\}$, respectively; moreover, denote for any set $S \subseteq \{0,1\}^n$, denote by $Cl_\wedge(S)$ resp. $Cl_\vee(S)$ the closure of S under intersection resp. union, (i.e., conjunction resp. disjunction of vectors). It is well-known that the Horn functions are those functions f whose set $F(f)$ of false vectors is closed under intersection, i.e., satisfying $F(f) = Cl_\wedge(F(f))$.

As shown in [13, 3], a Horn extension of a pdBF (T, F) can be found in polynomial time (if any exists). In fact, a Horn extension for (T, F) exists precisely if T is disjoint from $Cl_\wedge(F)$. However, this characterization shows that the Horn extension problem is, in a sense, unbalanced in (T, F). From a conceptual point, we would like to have a more balanced role of T and F in the condition for a Horn extension. Thus, we ask for suitable constraints on Horn functions in order to reach this goal.

A natural and suggestive possibility is to require symmetry between T and F, i.e., the role of T and F is interchangeable in the condition for an extension. This gives rise double Horn functions. A function f is *double Horn*, if $F(f) = Cl_\wedge(F(f))$ and $T(f) = Cl_\wedge(T(f))$; i.e., both f and $\overline{f}$ are Horn. For example,

$$f \;=\; \overline{x}_1 \vee x_2 x_3 \overline{x}_4 \vee x_2 x_3 x_5 x_6 \overline{x}_7$$

is double Horn, because

$$\overline{f} \;=\; x_1(\overline{x}_2 \vee \overline{x}_3 \vee x_4)(\overline{x}_2 \vee \overline{x}_3 \vee \overline{x}_5 \vee \overline{x}_6 \vee x_7) \;=\; x_1\overline{x}_2 \vee x_1\overline{x}_3 \vee x_1 x_4 \overline{x}_5 \vee x_1 x_4 \overline{x}_6 \vee x_1 x_4 x_7.$$

---

*The authors gratefully acknowledge the partial support of the Scientific Grant in Aid by the Ministry of Education, Science and Culture of Japan. The third author is supported by Research Fellowships of the Japan Society for the Promotion of Science for Young Scientists. The major part of this research was conducted while the first author visited Kyoto University in 1995, by the support of the Scientific Grant in Aid by the Ministry of Education, Science and Culture of Japan (Grant 06044112).

†Christian Doppler Laboratory for Expert Systems, Institut für Informationssysteme, Technische Universität Wien, 1040 Vienna, Austria. (Email: eiter@dbai.tuwien.ac.at)

‡Department of Applied Mathematics and Physics, Graduate School of Engineering, Kyoto University, Kyoto 606, Japan. (Email: ibaraki@kuamp.kyoto-u.ac.jp)

Another possibility is to require that T and F behave dually, since 0 and 1 are dual values, and T and F have dual roles. This gives rise to bidual Horn functions. A function f is *bidual Horn*, if $F(f) = Cl_\wedge(F(f))$ and, dually, $T(f) = Cl_\vee(T(f))$; i.e., both f and and its dual f^d are Horn. For example,

$$f \;=\; \overline{x}_1 x_2 x_3 \vee x_1 \overline{x}_3 x_4 \vee x_2 x_3 x_4$$

is bidual Horn, because

$$f^d \;=\; (\overline{x}_1 \vee x_2 \vee x_3)(x_1 \vee \overline{x}_3 \vee x_4)(x_2 \vee x_3 \vee x_4) \;=\; \overline{x}_1 x_4 \vee x_1 x_2 \vee x_2 \overline{x}_3 \vee x_2 x_4 \vee x_1 x_3 \vee x_3 x_4.$$

In particular, every positive function is bidual Horn. Notice that the classes of double and bidual Horn functions are incomparable. Clearly, $f = \overline{x}_1 \vee \overline{x}_2$ is double Horn but not bidual Horn; on the other hand, $g = x_1 \vee x_2$ is bidual Horn but not double Horn.

A third possible restriction on Horn functions emerging from the above line of thinking is the condition that both f and $f^* = f(\overline{x})$ are Horn. Those functions, which are precisely the *submodular* functions, are considered in [9], where graph-theoretic characterizations and efficient recognition algorithms are presented. The characterizations imply that also the extension problem for these functions is polynomial.

This extended abstract contains some results of a comprehensive investigation into double Horn and bidual Horn functions [8]. Its main contributions can be shortly summarized as follows.

- We introduce the classes of double Horn functions, $\mathcal{C}_{DH}$, and bidual Horn functions, $\mathcal{C}_{BH}$. We investigate their properties and describe relationships to other classes of Boolean functions.
- We investigate important computational problems for double and bidual Horn functions. In particular, we consider recognition from a formula, computation of an extension for a pdBF, and variants of the latter problem. For each problem, we present polynomial time algorithms (several are linear time) or prove intractability. We also present algorithms for enumerating all extensions with polynomial delay.
- We study the closures of double and bidual Horn functions under renamings (i.e., a change in the polarity of some variables). It appears that the closure of double Horn functions, $\mathcal{C}^R_{DH}$, is properly contained in the closure of bidual Horn functions, $\mathcal{C}^R_{BH}$. Moreover, the class $\mathcal{C}^R_{DH}$ corresponds 1-1 to a class $\mathcal{C}_{R\text{-}1}$ of read-once functions, i.e., functions definable by an expression in which each variable occurs at most once (cf., e.g., [11, 2]), and we obtain unexpected relationships to other well-known classes of Boolean functions.
- We present polynomial time algorithms for the main computational problems on $\mathcal{C}^R_{DH}$ and $\mathcal{C}^R_{BH}$, or prove intractability. Some of the algorithms are obtained by extending algorithms for $\mathcal{C}_{DH}$ resp. $\mathcal{C}_{BH}$, while others require new techniques.

In this extended abstract, we concentrate on the extension problem:

(Extension) : Given a pdBf (T, F), does there exist an extension f of (T, F) such that $f \in \mathcal{C}$?

Variants of this problem concern inquiring the uniqueness of an extension, and generation of all extensions (and representations thereof), possibly subject to certain selection criteria or in a specified order.

For space reasons, we do not provide proofs of the results. They are given in [8], where many additional results appear.

We use letters a, b, c and u, v, w to denote vectors in $\{0,1\}^n$; furthermore, $\mathbf{0} = (0, 0, \ldots, 0)$ and $\mathbf{1} = (1, 1, \ldots, 1)$. For a set S of vectors, let $\bigwedge S$ (resp., $\bigvee S$) denote the intersection (resp., union) of all vectors in S. For each $a = (a_1, a_2, \ldots, a_n)$ we define $ON(a) = \{i \mid a_i = 1\}$ and $OFF(a) = \{i \mid a_i = 0\}$, and denote $\overline{a} = (\overline{a}_1, \ldots, \overline{a}_n)$, where $\overline{a}_i = 1 - a_i$, $i = 1, \ldots, n$.

For any Boolan function f, we denote by $\overline{f}$, f^d, and f^* its *negation* (or *complement*), *dual*, and *contra-dual*, respectively, which are defined by $T(\overline{f}) = F(f)$, $F(f^d) = \{a \mid \overline{a} \in T(f)\}$, and $T(f^*) = \{a \mid \overline{a} \in T(f)\}$.

We use Boolean variables $x_1, \ldots, x_n$. A *term* t is a conjunction $\bigwedge_{i \in P(t)} x_i \wedge \bigwedge_{j \in N(t)} \overline{x}_j$ of literals (conjunction symbols my be omitted) such that $P(t) \cap N(t) = \emptyset$; by $V(t)$ we denote the variable indices in t. The length of any formula φ, denoted by $|\varphi|$, is the number of symbols in φ. A term t is positive if $N(t) = \emptyset$, and Horn if $|N(t)| \leq 1$. A DNF $\varphi = \bigvee_i t_i$ is called positive if all t_i are positive, and Horn if all t_i are Horn.

2 Double Horn functions

The following theorem tells us that double Horn functions have a Horn DNF representation which can be built from some ordering of the variables.

Let $V = \{1, 2, \ldots, n\}$. For an ordering $L = i_1, i_2, \ldots, i_n$ on V, define a set of Horn terms Γ_L by

$$\Gamma_L = \{\overline{x}_{i_1},\ x_{i_1}\overline{x}_{i_2}, \ldots\ldots\ldots,\ x_{i_1}x_{i_2}\ldots x_{i_{n-2}}\overline{x}_{i_{n-1}},\ x_{i_1}x_{i_2}\ldots x_{i_{n-1}}\overline{x}_{i_n},\ x_{i_1}x_{i_2}\ldots x_{i_n}\}.$$

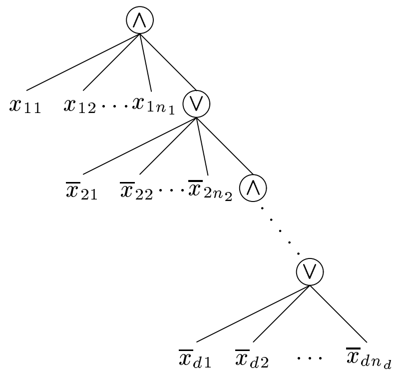
Theorem 2.1 *Let f be a function of n variables. Then f is double Horn iff f can be represented by a DNF $\varphi = \bigvee_{t \in S} t$ for some ordering L on V and $S \subseteq \Gamma_L$.*

From this theorem, we obtain the following result, which gives an alternative syntactical characterization of double Horn functions.

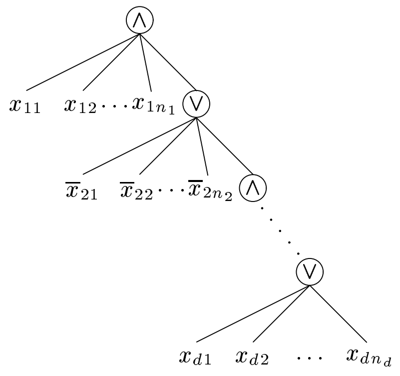
Theorem 2.2 *A function f is double Horn iff it can be represented by a read-once formula of the following type:*

$$\psi = \begin{cases} x_{11}x_{12}\ldots x_{1n_1}(\overline{x}_{21} \vee \overline{x}_{22} \vee \ldots \vee \overline{x}_{2n_2} \vee (x_{31}x_{32}\ldots x_{3n_3}(\ldots(\overline{x}_{d1} \vee \overline{x}_{d2} \vee \ldots \vee \overline{x}_{dn_d})))) & \text{if } d \text{ is even} \\ x_{11}x_{12}\ldots x_{1n_1}(\overline{x}_{21} \vee \overline{x}_{22} \vee \ldots \vee \overline{x}_{2n_2} \vee (x_{31}x_{32}\ldots x_{3n_3}(\ldots(x_{d1}x_{d2}\ldots x_{dn_d})))) & \text{if } d \text{ is odd,} \end{cases} \tag{2.1}$$

where $d \geq 0$, $n_1 \geq 0$, $n_i \geq 1$ for $i = 2, 3, \ldots, d$, and the $x_{11}, x_{12}, \ldots, x_{dn_d}$ are all different (see figure below).



(i) d is even.



(ii) d is odd.

In particular, this representation theorem implies that double Horn functions have a unique prime DNF, and that this prime DNF is short.

Corollary 2.1 *Every $f \in \mathcal{C}_{DH}$ has a unique prime DNF, which has the form $\varphi = \bigvee_{i=1}^{m} t_1 t_2 \cdots t_i \overline{x}_{\ell_i}$ $(*)$, where t_i and x_{ℓ_i}, $i = 1, 2, \ldots, m$, are pairwise disjoint positive terms (in this case, literals x_{ℓ_i} are also regarded as terms), and some of $t_1, t_2, \ldots, t_m$ and x_{ℓ_m} may be empty. In particular, $(*)$ implies $\varphi = \bot$ if $m = 0$. Conversely, every such formula φ represents an $f \in \mathcal{C}_{DH}$.*

We mention here some results about recognition of a double Horn function from a formula φ. If φ is arbitrary, the problem is co-NP-complete and co-NP-hard even for DNF. However, using graph techniques, it can be shown that recognition from a Horn formula φ is possible in linear time. Moreover, polynomial time recognition is possible if the formula φ is from a class that satisfies certain conditions. Among these classes are, e.g., 2DNF formulas, 2CNF formulas, and positive Boolean formulas.

The double Horn extension problem is polynomially solvable. In fact, even a linear time algorithm, based on Theorem 2.2, is possible. Basically, the method is to use lists with cross-references and counters to avoid that the same bit of the input is looked up more than once. Together with counter maintenance that takes constant time each step, this will give us linear running time.

We use counters $\#T_1, \#T_2, \ldots, \#T_n$ so that $\#T_i$ tells how many vectors in T have value 1 at component i, i.e., $|\{v \in T \mid v_i = 1\}|$. In order to find out the counters with certain value quickly, we prepare buckets $BT[0], BT[1] \ldots, BT[m]$ so that i is in bucket $BT[\#T_i]$, where $m = |T|$. Moreover, we use a counter $\#T$ that tells the number of vectors in T (i.e., $|T|$). Furthermore, we keep lists LT_i of all the vectors $v \in T$ such that $v_i = 0$, and we establish at each component i of v with $v_i = 0$ a pointer to the entry of v in list LT_i. For F, we use completely analogous data structures $\#F_i$, $\#F$, LF_i and $BF[k]$. Fig. 1 shows the data structure for T in a simple example.

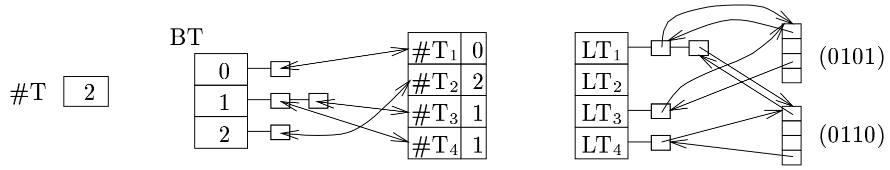


Figure 1: Data structure for set $T = \{(0101), (0110)\}$ used in DH-EXTENSION+.

The algorithm, described in detail below, first scans the input and builds the data structures. After that, it processes, exploiting Theorem 2.2, the sets of vectors and tries to construct a read-once formula (2.1) compatible with the input pdBF (T, F).

Algorithm DH-EXTENSION+
Input: A pdBf (T, F), where $T, F \subseteq \{0,1\}^n$.
Output: A read-once formula ψ (2.1) for f if a double Horn extension f of (T, F) exists; otherwise, "No".

Step 0. Scan the input: build the data structures and set up counters and buckets.

Step 1. Call DH-AUX+ with the data structures, and output the obtained result. Halt.

Procedure DH-AUX+
Input: The data structures as stated above representing (T, F), $T, F \subseteq \{0,1\}^n$.
Return: A read-once formula ψ (2.1) for f if a double Horn extension f of (T, F) exists; otherwise, "No".

Step 1. **if** $\#T = 0$ **then** return $\psi := \bot$ and exit; **else if** $\#F = 0$ **then** return $\psi := \top$ and exit; **else** set $I^+ = J^+ := \emptyset$;

Step 2. **if** $BT[\#T] = \emptyset$ **then** go to Step 3;
  else for each i in $BT[\#T]$ **do:** add i to I^+;
   while there is a vector w in LF_i **do:** $\#F := \#F - 1$;
    for each component j with $w_j = 0$ **do:** remove w from LF_j;
    for each component j with $w_j = 1$ **do:** $\#F_j := \#F_j - 1$;
   Call DH-AUX+ with the current data structures;
    if it returns ψ' $(\neq \top)$ **then** return $\psi := \psi' \wedge \bigwedge_{i \in I^+} x_i$;
    else if it returns $\top$ **then** return $\psi := \bigwedge_{i \in I^+} x_i$;
    else return "No";
   Exit;

Step 3. **if** $BF[\#F] = \emptyset$ **then** return "No" and exit;
  else for each i in $BF[\#F]$ **do** add i to J^+;
   while there is a vector v in LT_i **do:** $\#T := \#T - 1$;
    for each component j with $v_j = 0$ **do:** remove v from LT_j;
    for each component j with $v_j = 1$ **do:** $\#T_j := \#T_j - 1$;
   Call DH-AUX+ with the current data structures;
    if it returns ψ' $(\neq \bot)$ **then** return $\psi := \psi' \vee \bigvee_{i \in J^+} \overline{x}_i$;
    else if it returns $\bot$ **then** return $\psi := \bigvee_{i \in J^+} \overline{x}_i$;
    else return "No";
   Exit; □

Theorem 2.3 *Given a pdBf (T, F), where $T, F \subseteq \{0,1\}^n$, algorithm DH-EXTENSION+ correctly finds a read-once formula ψ of* (2.1) *for a double Horn extension or outputs "No" in $O(n(|T|+|F|))$ time.*

Corollary 2.2 *Given a pdBf (T, F), where $T, F \subseteq \{0,1\}^n$, the (unique) prime DNF of a double Horn extension can be computed in $O(n(|T|+|F|+n))$ time.*

A byproduct of the linear time extension algorithm is a fast algorithm for a problem on combining intersection closures of vectors. The following condition characterizes the existence of a double Horn extension.

Theorem 2.4 *Let (T, F) be a pdBf. Then (T, F) has an extension $f \in \mathcal{C}_{DH}$ iff $Cl_\wedge(T) \cap Cl_\wedge(F) = \emptyset$.*

Corollary 2.3 *Given a pdBf (T, F), deciding $Cl_\wedge(T) \cap Cl_\wedge(F) = \emptyset$ can be done in $O(n(|T|+|F|))$ time.*

2.0.1 Computing all double Horn extensions

We describe an algorithm that outputs the read-once formulas for all double Horn extensions of (T, F) with *polynomial delay* [12], i.e., the time between consecutive outputs is bounded by a polynomial in the input size, and the first (resp., last) output occurs also in polynomial time after (resp., before) the start (resp., halt) of the algorithm.

The algorithm, ALL-DH-EXTENSIONS, is a slight variant of algorithm DH-EXTENSION. It builds a read-once formula γ for an extension from left to right, exploiting Theorem 2.2. Each extension output has a maximum common prefix with the last output extension. To avoid multiple outputs of the same extension, an implicit ordering of the literals is used at each level in the formula.

The formula γ is represented by the list of its literals, which is preceded for technical convenience by a special literal x_0 that represents $\top$. For example, $\overline{x}_1 \vee x_3\overline{x}_2$ is represented by $x_0, \overline{x}_1, x_3, \overline{x}_2$.

Before describing the main algorithm, we first explain the algorithm REST-DH-EXTENSION, which checks whether a given (T, F) has a double Horn extension f, projected to a given $I \subseteq \{1, 2, \ldots, n\}$, such that

> **Property (i)** f is not allowed to use positive (resp., negative) literals in the first decomposition (i.e., literals $x_{11}, x_{12}, \ldots x_{1n_1}$ (resp., $\overline{x}_{11}, \overline{x}_{12}, \ldots \overline{x}_{1n_1}$) of (2.1)) other than those from a given $P = \{x_j \mid j \in I_P\}$ (resp., $N = \{\overline{x}_j \mid j \in I_N\}$),

where I_P (resp., I_N) satisfies $I_P \subseteq \bigcap_{v \in T} ON(v) \cap I$ (resp., $I_N \subseteq \bigcap_{w \in F} ON(w) \cap I$). For example, suppose that $T = \{110, 111\}$, $F = \{101, 001\}$, $I = \{1, 2, 3\}$ and $P = \{x_2\}$ are given. Then an extension $f = \overline{x}_3 \vee x_2$ has the desired property, but $g = x_1x_2$ does not.

Algorithm REST-DH-EXTENSION
Input: A pdBf (T, F), where $T, F \subseteq \{0,1\}^n$, $I \subseteq \{1, 2, \ldots, n\}$ and $P = \{x_j \mid j \in I_P\}$ (resp., $N = \{\overline{x}_j \mid j \in I_N\}$), where I_P (resp., I_N) satisfies $I_P \subseteq \bigcap_{v \in T} ON(v) \cap I$ (resp., $I_N \subseteq \bigcap_{w \in F} ON(w) \cap I$),
Output: "Yes", if $(T[I], F[I])$ has a double Horn extension that satisfies Property (i), where $T[I]$ and $F[I]$ denote the projection of T and F to I, respectively; otherwise, "No".

> **Step 1.** Check if $(T[I], F[I])$ has a double Horn extension. If not, output "No" and halt.
>
> **Step 2.** For each $j \in I_P$ (resp., $j \in I_N$), check if there is a $w \in F$ (resp., $v \in T$) such that $ON(w) \cap I_P = I_P \setminus \{j\}$ (resp., $ON(v) \cap I_N = I_N \setminus \{j\}$). If this condition fails for some j, then output "Yes" and halt.
>
> **Step 3.** Let $F_P := F \setminus \{w \in F \mid OFF(w) \cap I_P \neq \emptyset\}$ (resp., $T_N := T \setminus \{v \in T \mid OFF(v) \cap I_N \neq \emptyset\}$). If $F_P = \emptyset$ or $\bigcap_{w \in F_P} ON(w) \cap (I \setminus I_P) \neq \emptyset$ (resp., $T_N = \emptyset$ or $\bigcap_{v \in T_N} ON(v) \cap (I \setminus I_N) \neq \emptyset$), then output "Yes"; otherwise, output "No". Halt. □

Lemma 2.1 *Algorithm REST-DH-EXTENSION correctly outputs the answer in $O(n(|T|+|F|))$ time.*

The main algorithm for finding all double Horn extensions is now described below.

Algorithm ALL-DH-EXTENSIONS
Input: A pdBf (T, F), where $T, F \subseteq \{0,1\}^n$.
Output: Read-once formulas $\psi_1, \psi_2, \ldots, \psi_m$ of form (2.1) for all double Horn extensions of (T, F).

Step 1. If $T = \emptyset$, then output $\perp$. Otherwise call DH-EXTENSION+ for (T, F); if its answer is "No" then halt. Let $L_0 := x_0$, $I := \{1, 2, \ldots, n\}$, $P = \{x_j \mid j \in \bigcap_{v \in T} ON(v)\}$ and $N = \{\overline{x}_j \mid j \in \bigcap_{w \in F} ON(w)\}$. Call ALL-DH-AUX (T, F) with $L = L_0$, I, P and N, and consecutively output the obtained result. Halt. □

Procedure ALL-DH-AUX
Input: A pdBf (T, F), formula γ of form (2.1) given by list $L = L_0, L_1, \ldots, L_i$ of literals (in this order), set I of available variable indices, and sets of literals P and N allowed for decomposition (where $I_P = (\{j \,|x_j \in P\}) \subseteq I$ and $I_N = (\{j \,|\overline{x}_j \in N\}) \subseteq I$ are assumed).
Return: Read-once formulas $\gamma \vee \psi$ or $\gamma \wedge \psi$, where ψ are the read-once formulas of (2.1) for all double Horn extensions of $(T[I], F[I])$ (i.e., all read-once formulas for double Horn extensions of (T, F), which have γ as prefix) such that the literal L_{i+1} after γ is from $P \cup N$.

Step 1. (Check if the current formula γ represents an extension, and output it if it is the case.) If L_i is positive and $F = \emptyset$, or L_i is negative and $T = \emptyset$, then return γ.

Step 2. (Expand γ by a positive literal.) For each $x_j \in P$, let $T' := T$, $F' := \{v \in F \mid v_j = 1\}$, $I' := I \setminus \{j\}$ and $P := P \setminus \{x_j\}$. Call REST-DH-EXTENSION with (T', F'), I' and P; if it answers "Yes", then expand γ by x_j, and call recursively ALL-DH-AUX for $(T', F'), L = L_0, L_1, \ldots, L_{i+1}, I', P$ and N', where $L_{i+1} := x_j$ and $N' := \{\overline{x}_i \mid i \in I' \cap \bigcap_{w \in F'} ON(w)\}$.

Step 3. (Expand γ by a negative literal.) For each $\overline{x}_j \in N$, let $T' := \{v \in T \mid v_j = 1\}$, $F' := F$, $I' := I \setminus \{j\}$ and $N := N \setminus \{\overline{x}_j\}$. Call REST-DH-EXTENSION with (T', F'), I', N; if it answers "Yes", then extend γ by $\overline{x}_j$, and call recursively ALL-DH-AUX for $(T', F'), L = L_0, L_1 \ldots, L_{i+1}, I', P'$ and N, where $L_{i+1} := \overline{x}_j$ and $P' := \{x_i \mid i \in I' \cap \bigcap_{v \in T'} ON(v)\}$. □

The main algorithm initializes the variables and takes care of the case that $\perp$ is an extension. Step 1 of ALL-DH-AUX outputs the extension that is currently encountered. In particular, if $L = x_0$ we have by convention $\gamma = \top$. Steps 2 and 3 try to expand the formula at hand, in all possible ways. By recursively restricting the possibilities, it is achieved that each extension is output only once.

Theorem 2.5 *Algorithm ALL-DH-EXTENSIONS correctly outputs read-once formulas $\varphi_1, \varphi_2, \ldots, \varphi_m$ for all double Horn extensions of (T, F) with $O(n^3(|T| + |F|))$ delay, where $\varphi_i \not\equiv \varphi_j$ for $i \neq j$.*

Corollary 2.4 *There is a polynomial delay algorithm for enumerating (unique) prime DNFs for all double Horn extensions of a given pdBf (T, F).*

Furthermore, we obtain as a corollary to Theorem 2.5 that deciding whether a given pdBf (T, F) implicitly defines only one double Horn function is tractable.

Corollary 2.5 *Given a pdBf (T, F), deciding whether it has a unique double Horn extension is possible in polynomial time.* □

It is not clear whether enumerating all double Horn extensions can be done with linear time delay. The following example is an instance which has a unique double Horn extension, but checking this in linear time seems not straightforward. Take $T = \{\mathbf{1}\}$ and $F = \{v \mid |OFF(v)| = 1\}$. Then (T, F) has the unique doulbe Horn extension $x_1x_2\ldots x_n$. Although this is true even if arbitrary vectors except $\mathbf{1}$ are added to F, checking the fact appears to be more difficult.

Since computing some double Horn extension is fast, and the formulas φ for all extensions have small $|\varphi|$, a natural question that arises is how complex is computing a shortest one, measured in the size of the returned formula φ. It can be shown that this problem is NP-hard.

3 Bidual Horn functions

The following lemma is a useful characterization of bidual Horn functions. In fact, due to this criterion, it is possible to check whether a given Horn formula φ represents a bidual function in polynomial time.

For any pair of terms t_i and t_j, let us denote by $t^+_{i,j}$ the positive term such that $P(t^+_{i,j}) = P(t_i) \cup P(t_j)$, and by $t^\pm_{i,j}$ the positive term such that $P(t^\pm_{i,j}) = V(t_i) \cup V(t_j)$. For example, if $t_1 = x_1\overline{x}_2x_3$ and $t_2 = \overline{x}_1x_4$, then $t^+_{1,2} = x_1x_3x_4$ and $t^\pm_{1,2} = x_1x_2x_3x_4$.

Theorem 3.1 *Let φ be a Horn DNF. Then φ represents a bidual Horn function iff $t^+_{i,j} \leq \varphi$ (equivalently, $t^\pm_{i,j} \leq \varphi$) holds for all pairs of Horn terms t_i and t_j in φ such that $|N(t_i) \cup N(t_j)| = 2$.*

Exploiting this theorem, we obtain a polynomial time algorithm for checking the biduality of a Horn DNF φ. For every pair of terms t_i and t_j in φ with a distinct negative literal. Each test $t^+_{i,j} \leq \varphi$ can be done in $O(|\varphi|)$ time [6, 4]. Totally, it requires $O(\binom{m}{2}|\varphi|) = O(m^2|\varphi|)$ time.

However, recognition of a bidual Horn function from a non-Horn formula is co-NP-complete in general, and co-NP-hard even if φ is in DNF. Notice that for Horn functions and double Horn functions, the similar necessary conditions $T \cap Cl_\wedge(F) = \emptyset$ and $Cl_\wedge(T) \cap Cl_\wedge(F) = \emptyset$, respectively, for the existence of an extension are also sufficient. However, for bidual Horn extensions, the necessary condition $Cl_\vee(T) \cap Cl_\wedge(F) = \emptyset$ is not sufficient, which is shown by the following example.

Example 3.1 Let (T, F) be a pdBf defined by $T = \{a^{(1)} = (10100), a^{(2)} = (01010)\}$ and $F = \{b^{(1)} = (11111), b^{(2)} = (11100)\}$. It is easily checked that $Cl_\vee(T) \cap Cl_\wedge(F) = \emptyset$ holds. However, it can be verified that (T, F) has no bidual Horn extension.

In spite of all these complications with bidual Horn functions, we can fortunately show that a bidual Horn extension can be found in polynomial time. For a pdBf (T, F), let us define $F^+(v) = \{w \in F \mid w \geq v\}$, and $I(v) = ON(\bigwedge F^+(v)) \setminus ON(v)$. In particular, if no $w \in F^+(v)$ exists, we have $I(v) = OFF(v)$. Then it is not difficult to verify that for every $v \in T$ such that $v \neq \mathbf{1}$, any Horn extension f of (T, F) must have an Horn implicant t of f such that $P(t) = ON(v)$ and $N(t) \subseteq I(v)$.

We denote for every $v \in T$ by $R(v)$ the set of all Horn terms t_v such that (i) $P(t_v) = ON(v)$, and (ii) $\emptyset \neq N(t_v) \subseteq I(v)$ if $v \neq \mathbf{1}$, and $N(t_v) = \emptyset$ if $v = \mathbf{1}$. Such a term $t_v \in R(v)$ is called *canonical* for v (with respect to (T, F)). For a pdBf (T, F), $\varphi = \bigwedge_{v \in T} t_v$, where $t_v \in R(v)$ is called *canonical* Horn DNF of (T, F). Since any canonical Horn DNF of (T, F) represents an extension of (T, F), a Boolean function represented by a canonical Horn DNF is called *canonical* Horn extension of (T, F).

The following lemma is the key to our algorithm for finding a bidual Horn extension.

Lemma 3.1 *A pdBf (T, F) has an extension $f \in \mathcal{C}_{BH}$ iff there exists a canonical term $t_v \in R(v)$ for every $v \in T$, such that any term $t^+_{v,w}$ defined from a pair of terms t_v and t_w with $|N(t_v) \cup N(t_w)| = 2$ satisfies $T(t^+_{v,w}) \cap F = \emptyset$. Furthermore, given such a choice t_v for $v \in T$, the DNF $\varphi = \bigvee_{v \in T} t_v \vee \bigvee_{\substack{v,w \in T: \\ |N(t_v) \cup N(t_w)| = 2}} t^+_{v,w}$ represents a bidual Horn extension of (T, F).*

The lemma implies an $O(n|T|^2|F|)$ time algorithm that finds a bidual Horn extension using graph techniques, which is described in [8]. However, the algorithm treats T and F unbalanced, which indicates that it might not be optimal.

Exploiting duality, we can find a faster algorithm. First, we introduce some additional concepts. A term t is called *co-Horn* if $|P(t)| \leq 1$. A DNF $\varphi = \bigvee_i t_i$ is called *co-Horn* if every term t_i is co-Horn; a Boolean function is co-Horn if it can be represented by some co-Horn DNF. Notice that f is co-Horn if and only if $f^* = f(\bar{x})$ is Horn. In the context of pdBfs (T,F), we thus define the canonical concept for co-Horn functions in a dual way. For any vector $v \in T$, let $F^-(v) = \{w \in F : w \leq v\}$, let co-$I(v) = OFF(\bigvee F^-(v)) \setminus OFF(v)$, and let co-$R(v)$ be the set of canonical co-Horn terms of v, which are all terms t such that $N(t) = OFF(v)$ and $P(t) = \{\ell\}$ with $\ell \in$co-$I(v)$ for $v \neq \mathbf{0}$ and the negative term $\bigwedge_{i=1}^n \overline{x}_i$ for $v = \mathbf{0}$.

Proposition 3.1 *Let (T,F) be a pdBF. Let $v \in T$, $t_v \in R(v)$ and $w \in F$, $t_w \in$ co-$R(w)$ for the pdBF (F,T). Then (i) $T(t_v) \cap T(t_w) \neq \emptyset$ implies $v \leq w$, and (ii) $v \leq w$, $T(t_v) \cap T(t_w) = \emptyset$ implies $N(t_v) = P(t_w)$.*

The next lemma describes the existence of a bidual Horn extension in terms of Horn and co-Horn functions.

Lemma 3.2 *A pdBf (T,F) has a bidual Horn extension f iff there exist a Horn function g_1 and a co-Horn function g_2 such that $T(g_1) \supseteq T$, $T(g_2) \supseteq F$ and $T(g_1) \cap T(g_2) = \emptyset$.*

Define a bipartite graph $G_{(T,F)}$ on a pair of vertex sets $T' = T \setminus \{\mathbf{1}\}$ and $F' = F$ such that an edge is between $v \in T'$ and $w \in F'$ if $v \leq w$, and attach each vertex $v \in T'$ (resp., $w \in F'$) the set $I(v)$ for (T,F) (resp., co-$I(w)$ for the pdBF (F,T)) as label $L(v)$ (resp., $L(w)$). For a connected component C of $G_{(T,F)}$, let $L(C)$ be the intersection of all labels in C, i.e., $L(C) = \bigcap_{v \in C} L(v)$.

Lemma 3.3 *A pdBf (T,F) has a bidual Horn extension iff $L(C) \neq \emptyset$ for every connected component C of $G_{(T,F)}$.*

Algorithm BH-EXTENSION
Input: A pdBf (T,F), where $T, F \subseteq \{0,1\}^n$.
Output: "Yes", if there is a bidual Horn extension of (T,F); otherwise, "No".

> **Step 1.** Construct the bipartite graph $G_{(T,F)}$.
>
> **Step 2.** Compute all connected components C_i $(i = 1, 2, \ldots, k)$ of $G_{(T,F)}$.
>
> **Step 3.** If $L(C_i) \neq \emptyset$ holds for all i, output "Yes"; otherwise, "No". Halt. □

Theorem 3.2 *Given a pdBf (T,F), the existence of a bidual Horn extension of (T,F) can be checked using BH-EXTENSION in $O(n|T||F|)$ time.* □

Furthermore, the next lemma says that, if Step 3 is changed to the following Step 3′, then we can compute a bidual Horn extension of (T,F).

> **Step 3′.** If $L(C_i) \neq \emptyset$ holds for all i, then output $\varphi = \bigvee_{v \in T} t_v \vee \bigvee_{\substack{v,w \in T: \\ |N(t_v) \cup N(t_w)| = 2}} t^+_{v,w}$ $(**)$, where $t_v = \bigwedge_{j \in ON(v)} x_j \bar{x}_l$ for a fixed $l \in L(C)$ where $v \in C$, and $t_{\mathbf{1}} = x_1 \cdots x_n$; otherwise, output "No".

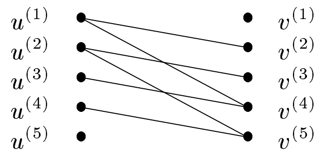
Lemma 3.4 *For a pdBf (T,F), let $\varphi_1 = \bigvee_{t_j \in S} t_j$ and φ_2 be respectively Horn and co-Horn DNFs satisfying that $T(\varphi_1) \supseteq T$, $T(\varphi_2) \supseteq F$ and $T(\varphi_1) \cap T(\varphi_2) = \emptyset$. Then the DNF $\varphi = \varphi_1 \vee \bigvee_{\substack{t_i, t_j \in S: \\ |N(t_i) \cup N(t_j)| = 2}} t^+_{i,j}$ represents a bidual Horn extension of (T,F).*

Note that $O(n|T|^2)$ time is needed, in order to output a bidual Horn DNF of (**) representing an extension of (T,F). Thus we have the following corollary.

Corollary 3.1 *Given a pdBf (T,F), an extension $f \in \mathcal{C}_{BH}$ of (T,F) can be computed in $O(n|T|(|T|+|F|))$ time.* □

Example 3.2 Suppose the input to BH-EXTENSION is the pdBF (T,F), $T=\{u^{(1)},u^{(2)},u^{(3)},u^{(4)},u^{(5)}\}$, $F=\{v^{(1)},v^{(2)},v^{(3)},v^{(4)},v^{(5)}\}$, where $u^{(1)}=(0010)$, $u^{(2)}=(0100)$, $u^{(3)}=(0011)$, $u^{(4)}=(0101)$, $u^{(5)}=(0110)$, $v^{(1)}=(1001)$, $v^{(2)}=(1010)$, $v^{(3)}=(1100)$, $v^{(4)}=(1011)$, $v^{(5)}=(1101)$.

Step 1. The graph $G=G_{(T,F)}$ is as follows:



Step 2. The graph G has four connected components, viz. $C_1=\{v^{(1)}\}$, $C_2=\{v^{(2)},u^{(1)},v^{(4)},u^{(3)}\}$, $C_3=\{v^{(3)},u^{(2)},v^{(5)},u^{(4)}\}$, and $C_4=\{u^{(5)}\}$.
Step 3. The labels $L(w)$ of the vertices w are as follows.

$$\begin{array}{ll}
L(u^{(1)})=I(u^{(1)})=\{1,3\}\setminus\{3\}=\{1\} & L(v^{(1)})=\text{co-}I(v^{(1)})=\{1,2,3,4\}\setminus\{2,3\}=\{1,4\}\\
L(u^{(2)})=I(u^{(2)})=\{1,2\}\setminus\{2\}=\{1\} & L(v^{(2)})=\text{co-}I(v^{(2)})=\{1,2,4\}\setminus\{2,4\}=\{1\}\\
L(u^{(3)})=I(u^{(3)})=\{1,3,4\}\setminus\{3,4\}=\{1\} & L(v^{(3)})=\text{co-}I(v^{(3)})=\{1,3,4\}\setminus\{3,4\}=\{1\}\\
L(u^{(4)})=I(u^{(4)})=\{1,2,4\}\setminus\{2,4\}=\{1\} & L(v^{(4)})=\text{co-}I(v^{(4)})=\{1,2\}\setminus\{2\}=\{1\}\\
L(u^{(5)})=I(u^{(5)})=\{1,2,3,4\}\setminus\{2,3\}=\{1,4\} & L(v^{(5)})=\text{co-}I(v^{(5)})=\{1,3\}\setminus\{3\}=\{1\}
\end{array}$$

Thus, the connected components of G have labels $L(C_1)=\{1,4\}$, $L(C_2)=\{1\}$, $L(C_3)=\{1\}$, $L(C_4)=\{1,4\}$. As a consequence, BH-EXTENSION outputs "Yes" (i.e., (T,F) has a bidual Horn extension).
Step 3′. Upon choosing $l=1$ from $L(C_2)$, $L(C_3)$, and $L(C_4)$, we obtain the bidual Horn extension

$$\varphi_1 \;=\; x_3\overline{x}_1 \vee x_2\overline{x}_1 \vee x_3x_4\overline{x}_1 \vee x_2x_4\overline{x}_1 \vee x_2x_3\overline{x}_1 \;=\; x_3\overline{x}_1 \vee x_2\overline{x}_1$$

for (T,F); choosing $l=1$ from $L(C_2)$, $L(C_3)$ and $l=4$ from $L(C_4)$, we would obtain the bidual Horn extension

$$\begin{aligned}\varphi_2 &= x_3\overline{x}_1 \vee x_2\overline{x}_1 \vee x_3x_4\overline{x}_1 \vee x_2x_4\overline{x}_1 \vee x_2x_3\overline{x}_4 \vee x_2x_3 \vee x_2x_3 \vee x_2x_3x_4 \vee x_2x_3x_4\\ &= x_3\overline{x}_1 \vee x_2\overline{x}_1 \vee x_2x_3.\end{aligned}$$

3.1 Computing all bidual Horn extensions

Different from the double Horn extensions, not every bidual Horn extension of a pdBf (T,F) can be represented by a short Horn DNF. This is easy to see, since every positive function is a bidual extension of (T,F) if $T=F=\emptyset$, and there are clearly positive functions, the size of whose unique prime DNF is exponential in the number of variables, e.g., $\varphi=\bigvee_{S\subseteq\{1,2,\ldots,n\}:|S|=\lfloor n/2\rfloor}\left(\bigwedge_{j\in S}x_j\right)$. Therefore, there is no polynomial delay algorithm which generates all bidual extensions. Furthermore, note that finding an extension $f\in\mathcal{C}_{BH}$ with a shortest DNF is co-NP-hard.

It appears that even deciding, given an extension, whether an additional extension exists (rather than outputting one) is not easy. In fact, this is at least as hard as the mutual duality problem (i.e., given positive DNFs φ and ψ, decide whether $\psi=\varphi^d$). The latter problem is polynomially equivalent to a number of other problems, cf. [1, 7, 12, 10], but no polynomial time algorithm is known to date.

Theorem 3.3 *Let a pdBf (T,F) and a Horn DNF φ representing a bidual Horn extension of (T,F) be given. If checking whether there exists an extension $g\in\mathcal{C}_{BH}$ with $g\neq f_\varphi$ can be done in polynomial time, then the mutual duality problem can be solved in polynomial time.*

Proof. We show this by a reduction of the Sperner saturation problem, which is known to be polynomially equivalent to the mutual duality problem [7]: given a Sperner family $\mathcal{S}=\{S_1,S_2,\ldots,S_m\}$ of subsets $S_i\subseteq V=\{1,2,\ldots,n\}$ (i.e., $S_i\not\subseteq S_j$ and $S_j\not\subseteq S_i$ for all $i\neq j$), decide whether it is not saturated (i.e., there does not exist a set $X\subseteq V$ such that $S_i\not\subseteq X\not\subseteq S_j$ holds for all i,j).

Given $\mathcal{S}$, we construct the instance of our problem as follows:

$$\begin{aligned} T &= \{x^{S_i} \mid i = 1, 2, \ldots, m\} \cup \{x^{S_i \cup \{l\}} \mid i = 1, 2, \ldots, m,\ \ l \in V \setminus S_i\} \\ F &= \{x^{S_i \setminus \{l\}} \mid i = 1, 2, \ldots, m,\ \ l \in S_i\} \\ \varphi &= \bigvee_{i=1}^{m} (\bigwedge_{l \in S_i} x_l), \end{aligned}$$

where x^S is the characteristic vector of S. It is easy to see that φ represents a bidual Horn extension of (T, F) since φ is positive. (It can be shown that, if we apply algorithm BH-EXTENSION to this instance, then it will always output the above DNF φ, independent of the choices made in the construction.) It holds that a bidual Horn extension $g \not\equiv f_\varphi$ of (T, F) exists iff $\mathcal{S}$ is not saturated. □

Corollary 3.2 *Given a pdBf (T, F), deciding whether it has a unique extension $f \in \mathcal{C}_{BH}$ is at least as hard as the mutual duality problem.* □

Corollary 3.3 *Let a pdBf (T, F) and a Horn DNF φ representing a bidual Horn extension of (T, F) be given. If checking whether there is a bidual Horn extension g such that $g \neq f_\varphi$ can be done in polynomial time, then there is a polynomial total time algorithm for the problem of dualizing a positive function, i.e, given a positive DNF, computing the prime DNF of its dual function.*

Proof. Follows from Theorem 3.3 and the result in [1] that there exists a polynomial total time algorithm for dualizing a positive function iff the mutual duality problem is polynomial. □

References

[1] J. C. Bioch and T. Ibaraki, Complexity of identification and dualization of positive Boolean functions, *Information and Computation*, 123 (1995) 50-63.

[2] E. Boros, V. Gurvich, P. L. Hammer, T. Ibaraki, and A. Kogan, Decomposability of partially defined Boolean functions, *Discrete Applied Mathematics*, 62 (1995) 51-75.

[3] E. Boros, T. Ibaraki and K. Makino, Error-free and best-fit extensions of partially defined Boolean functions, RUTCOR Research Report RRR 14-95, Rutgers University, 1995.

[4] V. Chandru, C. Coullard, P. Hammer, M. Montañez, and X. Sun, On renamable Horn and generalized Horn functions, *Annals of Mathematics and Artificial Intelligence*, 1 (1990) 33-48.

[5] Y. Crama, P. L. Hammer and T. Ibaraki, Cause-effect relationships and partially defined boolean functions, *Annals of Operations Research*, 16 (1988) 299-326.

[6] D. W. Dowling and J.H. Gallier, Linear-time algorithms for testing the satisfying of propositional Horn formulae, *Journal of Logic Programming*, 3 (1984) 267-284.

[7] T. Eiter and G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, *SIAM J. on Computing*, 24 (1995), 1278-1304.

[8] T. Eiter, T. Ibaraki and K. Makino, On Two-Face Horn Functions, Technical Report CD-TR 96/95, Information Systems Department, Technical University of Vienna, 1996. Forthcoming.

[9] O. Ekin, P.L. Hammer and U. N. Peled, Horn functions and submodular Boolean functions, RUTCOR Research Report RRR 01-95, Rutgers University, 1995.

[10] M. Fredman and L. Khachiyan, On the complexity of dualization of monotone disjunctive normal forms, Technical Report LCSR-TR-225, Department of Computer Science, Rutgers University, 1994.

[11] H. Hunt III and R. Stearns, The complexity of very simple Boolean formulas with applications, *SIAM J. Computing*, 19 (1990) 44-70.

[12] D. S. Johnson, M. Yannakakis, and C. H. Papadimitriou, On generating all maximal independent sets, *Information Processing Letters*, 27 (1988) 119-123.

[13] K. Makino, K. Hatanaka and T. Ibaraki, Horn extensions of a partially defined Boolean function, RUTCOR Research Report RRR 27-95, Rutgers University, 1995.