Information and Computation **144**, 155–190 (1998)
Article No. IC982713

# Double Horn Functions*



*Institut für Informatik, Universität Gießen, Arndtstraße 2, D-35392 Gießen, Germany*

Toshihide Ibaraki

*Department of Applied Mathematics and Physics, Graduate School of Engineering, Kyoto University, Kyoto 606, Japan*

and

Kazuhisa Makino

*Department of Systems and Human Science, Graduate School of Engineering Science, Osaka University, Toyonaka, Osaka, Japan*

E-mail: eiter@informatik.uni-giesen.de, ibaraki@kuamp.kyoto-u.ac.jp, makino@sys.es.osaka-u.ac.jp

In this paper, we define double Horn functions, which are the Boolean functions $f$ such that both $f$ and its complement (i.e., negation) $\bar{f}$ are Horn, and investigate their semantical and computational properties. Double Horn functions embody a balanced treatment of positive and negative information in the course of the extension problem of partially defined Boolean functions (pdBfs), where a pdBf is a pair $(T, F)$ of disjoint sets $T, F \subseteq \{0, 1\}^n$ of true and false vectors, respectively, and an extension of $(T, F)$ is a Boolean function $f$ that is compatible with $T$ and $F$. We derive syntactic and semantic characterizations of double Horn functions, and determine the number of such functions. The characterizations are then exploited to give polynomial time algorithms (i) that recognize double Horn functions from Horn DNFs (disjunctive normal forms), and (ii) that compute the prime DNF from an arbitrary formula, as well as its complement and its dual. Furthermore, we consider the problem of determining a double Horn extension of a given pdBf. We describe a polynomial time algorithm for this problem and moreover an algorithm that enumerates all double Horn extensions of a pdBf with polynomial delay. However, finding a shortest double Horn extension (in terms of the size of a formula $\varphi$ representing it) is shown to be intractable.


* 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).



0890-5401/98 $25.00

## 1. INTRODUCTION

A Boolean function (or a function in short) is *Horn* if it can be represented by a DNF (disjunctive normal form) in which each term contains at most one negative literal. Horn functions are at the heart of knowledge based systems, logical databases, and logic in computer science (see, e.g., [1, 4, 11, 30]). Many problems related to Boolean functions (resp., formulas) can be solved efficiently for Horn functions (resp., formulas), while they are intractable for arbitrary functions (resp., formulas); a well-known example is the classical satisfiability problem. This is a motivation for recent increasing activities on Horn functions, e.g., computing a minimum representation of a Horn function [16–18], learning and identification of Horn functions [1, 7, 29], constructing Horn approximations of non-Horn functions [35], and constructing Horn extensions [3, 29].

Let us denote the set of true (resp., false) vectors of a function $f$ as $T(f)$ (resp., $F(f)$). There is an elegant semantical characterization of a Horn function that $f$ is Horn if and only if its $F(f)$ is closed under intersection [20, 31]. This characterization, however, reveals an asymmetry between the roles of $T(f)$ and $F(f)$ for a Horn function $f$. From a conceptual point, therefore, we may want to have a more balanced role of $T(f)$ and $F(f)$ by imposing suitable additional constraints on Horn functions. A natural and suggestive possibility at hand is to require complete symmetry between $T(f)$ and $F(f)$. This gives rise to the concept of double Horn functions: A function $f$ is *double Horn* if both $T(f)$ and $F(f)$ are closed under intersection (equivalently, if both $f$ and its complement $\bar{f}$ are Horn). This characterization is paralleled by the one in terms of rules: $f$ is double Horn precisely if both $f$ and its negation $\bar{f}$ can be expressed by collections of Horn rules.

In this paper, we derive syntactic and semantic characterizations of double Horn functions, and give polynomial time algorithms for various associated problems. We also determine the number of such functions.

As another direction of applying double Horn functions, we consider partially defined Boolean functions (pdBfs) and their extensions. A pdBf is a natural generalization of the Boolean function, by allowing that the function values on some input vectors are unknown; it is described by a pair $(T, F)$ of sets $T$ and $F$ of true and false vectors $v \in \{0, 1\}^n$, respectively, where $T \cap F = \emptyset$. A pdBf arises in conjunction with data analysis, where $T$ represents a set of positive examples and $F$ a set of negative examples. A natural and important issue is whether a pdBf $(T, F)$ can be completed to a Boolean function $f$ chosen from a particular class of Boolean functions $\mathscr{C}$; i.e., establish a Boolean function (i.e., *extension*) $f\colon \{0, 1\}^n \mapsto \{0, 1\}$ in $\mathscr{C}$, such that $T(f) \supseteq T$ and $F(f) \supseteq F$.

For example, if a vector $v \in T \cup F$ gives the results of physical tests of a patient for some disease (e.g., $v_1$ denotes whether the patient is male ($v_1 = 1$) or female ($v_1 = 0$), $v_2$ denotes whether the patient is a smoker ($v_2 = 1$) or not ($v_2 = 0$), and so forth), an extension $f$ of pdBf $(T, F)$ is a description of the diagnosis for all possible data vectors $v \in \{0, 1\}^n$. Finding an extension $f$ of a pdBf $(T, F)$ is therefore an important subject in such fields as knowledge acquisition, knowledge discovery, and data mining, which are receiving increasing attention currently. In fact, the existence of an extension in a class $\mathscr{C}$ of Boolean functions is a necessary condition for the truth of a hypothesis that the relationship between different attributes is described by some function in $\mathscr{C}$. In the spirit of Popper's falsification principle, we can refute such a hypothesis if no extension in $\mathscr{C}$ exists. The extension problem is also relevant to machine learning, in which a learning algorithm gradually refines a pdBf $(T, F)$, until it finally outputs a Boolean function. In exact learning, this is a function $f$ which should be learned, where $f$ is known to be from a certain class $\mathscr{C}$ of Boolean functions, while in a probabilistic setting, the output is a hypothesis $f$ obeying certain quality bounds.

The extension problem has been investigated for a number of classes $\mathscr{C}$ of Boolean functions [3, 5, 29]. Among these classes are Horn functions, for which, as shown in [3, 29], an extension can be found in polynomial time. We extend these results to double Horn functions, and investigate related problems such as enumerating all double Horn extensions. A double Horn extension may be considered more natural than a Horn extension in the sense that positive examples $T$ and negative examples $F$ play a symmetric role in characterizing the existence of an extension.

Our main contributions in this paper can be summarized as follows.

• We introduce the class $\mathscr{C}_{DH}$ of double Horn functions and investigate its properties. In particular, we present a useful syntactic characterization and analyze relationships to other classes of Boolean functions. We show that $\mathscr{C}_{DH}$ corresponds 1–1 to a syntactic fragment of the class $\mathscr{C}_{R-1}$ of read-once functions, which are definable by Boolean formulas in which no variable occurs more than once. Read-once functions (resp., formulas) are well known and received a lot of attention, e.g., [8, 14, 15, 21, 23, 32, 33, 36, 37]. Moreover, we show that each double Horn function has the unique irredundant prime DNF, which contains few prime implicants.

• Based on the syntactic characterization, we develop an algorithm that recognizes a double Horn function from a given formula $\varphi$ (not necessarily Horn). The algorithm runs in polynomial time if $\varphi$ is from a class that satisfies some constraints. In particular, it is low-order polynomial for the class of Horn formulas.

• We also present a semantic characterization of double Horn functions in terms of their characteristic sets (or models) [24, 25, 27]. The characteristic set of a Horn function $f$ is the generating set of the false vectors $F(f)$ under intersection, i.e., the minimum set of vectors $C^*(F)$ such that $Cl_\wedge(C^*(F)) = F$, where $Cl_\wedge$ denotes the intersection closure. This semantic characterization can be naturally stated as a graph property, and we obtain a 1–1 correspondence between double Horn functions and oriented complete bipartite graphs.

• We study transformations of double Horn functions, and show that all the considered transformation problems are polynomially solvable. In particular, we show that the problem of dualizing a double Horn function $f$ (i.e., computing the prime DNF of the dual function $f^d$ of $f$ from an arbitrary representation of $f$) can be done in polynomial time.

• The above characterizations of double Horn functions allow us to derive the count of $\mathscr{C}_{DH}$. We show that there are precisely $2^{n+1}$ nonisomorphic double Horn functions on $n$ variables $x_1, ..., x_n$. The total number $\#DH(n)$ of double Horn functions on $n$ variables is much larger and amounts to simple closed expressions referring to (i) the number of ordered partitions of a set and (ii) the number of cycle-free complete bipartite digraphs.

- For the extension problem of a pdBf, we also present efficient algorithms. In particular, we present an algorithm that, given a pdBf $(T, F)$, finds a double Horn extension in polynomial time (if any exists). This positive result is complemented by the fact that computing a shortest double Horn extension (measured by the length of a formula describing it) is intractable. Moreover, we describe an algorithm that enumerates all double Horn extensions of a given pdBf $(T, F)$ with polynomial delay. By means of this result, we obtain that the uniqueness problem (i.e., deciding whether $(T, F)$ implicitly defines a unique double Horn function) is solvable in polynomial time.

Notice that the above results can be fruitfully applied to machine learning. In fact, our results on the extension problem allow us to immediately derive that double Horn functions are PAC (probably approximately correct) learnable [37]. On the other hand, the result on the uniqueness of an extension implies that, given a sample (i.e., a pdBf) consisting of positive and negative instances, it can be decided in polynomial time whether this sample suffices to identify a double Horn function which should be learned in a batch-mode [19], where no information beyond the sample is available. Moreover, a DNF of the function can be output efficiently in this case.

In passing, we note that, besides double Horn functions, other possibilities exist to balance the role of $T(f)$ and $F(f)$. For example, to require that $f$ and $f^d$ are Horn gives rise to bidual Horn functions [9], and to require that $f$ and $f^*$ (the contra-dual of $f$; see Section 2 for a definition) are Horn gives rise to the submodular functions [10].

The rest of this paper is organized as follows. In the next section, we state some preliminaries and fix notations. In Section 3, we introduce double Horn functions, derive a syntactical characterization of such functions, and consider the recognition problem. In Section 4, we study double Horn extensions. In Section 5, we derive a semantic characterization of double Horn functions, tackle the transformation problem, and determine the count of the class $\mathscr{C}_{DH}$. Section 6 concludes the paper by addressing further issues on double Horn functions and stating open problems.

## 2. PRELIMINARIES AND NOTATIONS

We usually use letters $a, b, c$ and $u, v, w$ to denote vectors in $\{0, 1\}^n$, and use $\mathbf{0} = (0, 0, ..., 0)$ and $\mathbf{1} = (1, 1, ..., 1)$. In general, we allow $n = 0$; here, $\{0, 1\}^0 = \{(\ )\}$, where $(\ )$ is the empty vector. For each $a = (a_1, a_2, ..., a_n)$, we define $ON(a) = \{i \mid a_i = 1\}$ and $OFF(a) = \{i \mid a_i = 0\}$, and denote $\bar{a} = (\bar{a}_1, \bar{a}_2, ..., \bar{a}_n)$, where $\bar{a}_i = 1 - a_i$, $i = 1, 2, ..., n$. As usual, $v \wedge w$ denotes the *intersection* (i.e., the componentwise conjunction) of vectors $v$ and $w$; e.g., if $v = (1100)$ and $w = (1010)$, then $v \wedge w = (1000)$. Let $S \subseteq \{0, 1\}^n$. The closure of $S$ under pairwise intersections (called the *intersection closure* of $S$) is denoted by $Cl_\wedge(S)$. Furthermore, for a set

$I \subseteq \{1, 2, ..., n\}$, $S[I]$ denotes the *projection* of $S$ to $I$; by $x^I$ we denote the characteristic vector of $I$, which is defined by $ON(x^I) = I$. For example, let $S = \{(11011), (01010), (00111)\}$. Then, $Cl_\wedge(S) = S \cup \{(00011), (00010)\}$. For $I = \{2, 4\}$, we have $S[I] = \{(11), (01)\}$ and $x^I = (01010)$.

Recall that a *Boolean function* (in short, *function*) is a mapping $f: \{0, 1\}^n \to \{0, 1\}$, $n \geqslant 0$. For $n = 0$, there are precisely two Boolean functions, $f = \bot$ and $f = \top$, which correspond to truth and falsity, respectively. The sets $T(f) = \{v \mid f(v) = 1\}$ and $F(f) = \{v \mid f(v) = 0\}$ are the true vectors and false vectors of $f$, respectively.

For any function $f$, we denote by $\bar{f}$, $f^d$, and $f^*$ its *negation* (or *complement*), *dual*, and *contra-dual*, respectively, which are defined by $T(\bar{f}) = F(f)$, $T(f^d) = \{\bar{a} \mid a \in F(f)\}$, and $T(f^*) = \{\bar{a} \mid a \in T(f)\}$. Note that $f^d = \bar{f}^*$. For any assignment $A = (x_{i_1} \leftarrow a_1, x_{i_2} \leftarrow a_2, ..., x_{i_k} \leftarrow a_k)$ to the variables $x_{i_j}$, where each of $a_1, a_2, ..., a_k$ is either 0 or 1, we denote by $f_A = f_{(x_{i_1} \leftarrow a_1, x_{i_2} \leftarrow a_2, ..., x_{i_k} \leftarrow a_k)}$ the function of $(n-k)$ variables obtained by fixing variables $x_{i_1}, x_{i_2}, ..., x_{i_k}$ as specified by $A$, and use the same notation for formulas $\varphi$.

A *partially defined Boolean function* (pdBf) is a mapping $g: T \cup F \mapsto \{0, 1\}$ defined by $g(v) = 1$ if $v \in T$; 0 if $v \in F$, where $T \subseteq \{0, 1\}^n$ denotes a set of true vectors (or positive examples) and $F \subseteq \{0, 1\}^n$ denotes a set of false vectors (or negative examples) such that $T \cap F = \emptyset$. For simplicity, a pdBf is denoted by a pair of sets $(T, F)$. It can be seen as a representation for all Boolean functions $f$ such that $T(f) \supseteq T$ and $F(f) \supseteq F$; any such $f$ is called an *extension* of $(T, F)$. A pdBf is called *complete* if $T \cup F = \{0, 1\}^n$.

Among the many classes of Boolean functions, the classes $\mathscr{C}_t$ of positive functions and $\mathscr{C}_{Horn}$ of Horn functions are well known, cf. [11, 30, 38]. A function $f$ is *positive* (also called *monotone*) if $v \leqslant w$ implies $f(v) \leqslant f(w)$, where $\leqslant$ is componentwise and $0 \leqslant 1$. A function $f$ is *Horn* if $F(f) = Cl_\wedge(F(f))$ holds, i.e., its false vectors are closed under intersection. This is equivalent to the well-known algebraic characterization

$$f(v \wedge w) \leqslant f(v) \vee f(w), \qquad \text{for all} \quad v, w \in \{0,1\}^n.$$

Assume that Boolean *variables* are from $x_1, x_2, ..., x_n$. A positive (resp., negative) *literal* $L$ is a variable $x_i$ (resp., its complement $\bar{x}_i$). A *term* $t$ is a conjunction $\bigwedge_{i \in P(t)} x_i \wedge \bigwedge_{j \in N(t)} \bar{x}_j$ of literals such that $P(t), N(t) \subseteq \{1, 2, ..., n\}$ and $P(t) \cap N(t) = \emptyset$; in the sequel, we omit conjunction symbols if no confusion arises. The empty term (representing truth) with $P(t) = N(t) = \emptyset$ is denoted by $\top$. Let $V(t) = P(t) \cup N(t)$ denote the indices of variables in $t$. A *disjunctive normal form* (DNF) $\varphi$ is a *disjunction* $\bigvee_{i=1}^k t_i$ of terms; the empty DNF (representing falsity) is denoted by $\bot$. The length of a DNF (or arbitrary formula) $\varphi$, denoted by $|\varphi|$, is the number of symbols in $\varphi$ (here a negative literal $\bar{x}_i$ counts as a single symbol). A term $t$ is *positive* if $N(t) = \emptyset$ and *Horn* if $|N(t)| \leqslant 1$. A DNF $\varphi = \bigvee_i t_i$ is called *positive* if all $t_i$ are positive and *Horn* if all $t_i$ are Horn.

It is well-known (and easy to prove) that a function is *positive* (resp., *Horn*) if and only if it can be represented by some positive (resp., Horn) DNF (see, e.g., [7, 20, 31, 38]). For example, $t_1 = x_1x_2x_4$, $t_2 = x_1x_4\bar{x}_5x_6$ and $t_3 = x_2\bar{x}_3\bar{x}_5$ are terms, while $t_4 = x_2x_4\bar{x}_2$ is not. Term $t_1$ is positive (and hence Horn) and has

$P(t_1)=\{1,2,4\}$ and $N(t_1)=\emptyset$; $t_2$ is Horn and has $P(t_2)=\{1,4,6\}$ and $N(t_2)=\{5\}$; and $t_3$ is neither positive nor Horn. The DNFs $\varphi^{(1)}=x_2 \vee x_1x_3 \vee x_1x_4$ and $\varphi^{(2)}=x_2\bar{x}_3 \vee x_1x_3 \vee \bar{x}_2x_3$ are positive and Horn, respectively.

A term $t$ or a formula $\varphi$ is also viewed as a function that it represents, if no confusion arises. For two functions $f$ and $g$, $f \leqslant g$ denotes $T(f) \subseteq T(g)$. A term $t$ is an *implicant* of a formula $\varphi$ (resp., function $f$) if $t \leqslant \varphi$ (resp., $t \leqslant f$) holds. An implicant $t$ is *prime* if no proper subterm of $t$ is an implicant. A DNF $\varphi=\bigvee_i t_i$ is called *prime* if all terms $t_i$ in $\varphi$ are prime implicants, and *irredundant* if no DNF, which is obtained by dropping some terms $t_i$ in $\varphi$, represents the same function. For example, a DNF $\varphi=x_1\bar{x}_2 \vee x_1\bar{x}_3 \vee x_2\bar{x}_3 \vee x_4$ is prime because all terms $x_1\bar{x}_2$, $x_1\bar{x}_3$, $x_2\bar{x}_3$, and $x_4$ are prime implicants, but it is not irredundant because $\varphi'=x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee x_4$ also represents the same function as $\varphi$.

The prime implicants of a DNF $\varphi$ can be generated by iterated consensus on terms. More precisely, let $t_1$ and $t_2$ be terms such that $P(t_1) \cap N(t_2)=\{l\}$ and $N(t_1) \cap P(t_2)=\emptyset$. A term $t_3$ is called the *consensus* of terms $t_1$ and $t_2$ if $P(t_3)=(P(t_1)\backslash\{l\}) \cup P(t_2)$ and $N(t_3)=N(t_1) \cup (N(t_2)\backslash\{l\})$; note that consensus operation is the dual of *resolution*. E.g., $x_2\bar{x}_3x_4\bar{x}_5x_6$ is the consensus of $x_1\bar{x}_3x_4\bar{x}_5$ and $\bar{x}_1x_2\bar{x}_3x_4x_6$. It is well known [34] that every prime implicant $t$ of an arbitrary function $f$ can be derived from the terms in any DNF $\varphi=\bigvee_i t_i$ for $f$ by applying a consensus procedure. In other words, there is a sequence $t^{(1)}, t^{(2)}, ..., t^{(m)}=t$ of terms such that each $t^{(k)}$ is either in $\varphi$ (i.e., $t^{(k)}=t_i$ for some $i$) or the consensus of two terms $t^{(k_1)}$ and $t^{(k_2)}$ such that $k_1, k_2<k$.

## 3. DOUBLE HORN FUNCTIONS

In this section, we introduce double Horn functions and investigate their properties. We start with syntactical characterization of such functions, which will then be used in Section 3.3. Moreover, these characterizations allow us to precisely determine the count of the class of double Horn functions.

### 3.1. Definitions and Characterizations

DEFINITION 1. A Boolean function $f$ is double Horn if and only if $T(f)=Cl_\wedge(T(f))$ and $F(f)=Cl_\wedge(F(f))$ (i.e., $Cl_\wedge(T(f))$ and $Cl_\wedge(F(f))$ satisfy $Cl_\wedge(T(f)) \cap Cl_\wedge(F(f))=\emptyset$ and $Cl_\wedge(T(f)) \cup Cl_\wedge(F(f))=\{0,1\}^n$). The class of all double Horn functions is denoted by $\mathscr{C}_{DH}$.

Equivalently, a function $f$ is double Horn if and only if it satisfies

$$f(x) \wedge f(y) \leqslant f(x \wedge y) \leqslant f(x) \vee f(y),$$

i.e., both $f$ and $\bar{f}$ are Horn. For example,

$$f=\bar{x}_1 \vee x_2x_3\bar{x}_4 \vee x_2x_3x_5x_6\bar{x}_7$$

is double Horn, because

$$\bar{f} = x_1(\bar{x}_2 \vee \bar{x}_3 \vee x_4)(\bar{x}_2 \vee \bar{x}_3 \vee \bar{x}_5 \vee \bar{x}_6 \vee x_7)$$
$$= x_1\bar{x}_2 \vee x_1\bar{x}_3 \vee x_1x_4\bar{x}_5 \vee x_1x_4\bar{x}_6 \vee x_1x_4x_7$$

is also Horn. Notice that the above inequations are different, but not unrelated, to the condition

$$f(x \wedge y) \vee f(x \vee y) \leqslant f(x) \vee f(y), \qquad \text{for every} \quad x, y \in \{0, 1\}^n,$$

which defines the class of *submodular* functions. As pointed out in [10], this condition is equivalent to the property that both $f$ and its contra-dual $f^*$ are Horn.

There is also a simple logical characterization of double Horn functions: Both $f$ and the negation $\bar{f}$ of a double Horn function $f$ can be expressed by collections of Horn rules

$$x_{i_1} \wedge \cdots \wedge x_{i_k} \rightarrow x_{i_0},$$

where the antecedent and/or the consequent of the rule may be empty. In the above example, $f$ is expressed by the rules $x_1 \rightarrow x_2$, $x_1 \rightarrow x_3$, $x_1 \wedge x_4 \rightarrow x_5$, $x_1 \wedge x_4 \rightarrow x_6$, and $x_1 \wedge x_4 \wedge x_7 \rightarrow \square$ (meaning that $x_1$, $x_4$ or $x_7$ is false); the complement $\bar{f}$ is expressed by the rules $\square \rightarrow x_1$ (meaning that $x_1$ is true), $x_2 \wedge x_3 \rightarrow x_4$, and $x_2 \wedge x_3 \wedge x_5 \wedge x_6 \rightarrow x_7$, where $\square$ denotes empty.

### *3.2. Horn DNFs and Read-Once Formulas of Double Horn Functions*

We shall present in this subsection syntactical characterizations of double Horn functions.

LEMMA 3.1. *Let $f$ be a double Horn function. Then $f_A$ is double Horn for every assignment $A$.*

*Proof.* Immediate from the definitions. ∎

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

$$\Gamma_L = \{\bar{x}_{i_1}, x_{i_1}\bar{x}_{i_2}, \; ... \, x_{i_1}x_{i_2} \cdots x_{i_{n-2}}\bar{x}_{i_{n-1}}, x_{i_1}x_{i_2} \cdots x_{i_{n-1}}\bar{x}_{i_n}, x_{i_1}x_{i_2} \cdots x_{i_n}\}.$$

In particular, $\Gamma_L = \{\top\}$ if $n = 0$. Note that all terms in $\Gamma_L$ are mutually orthogonal; i.e., $tt' = \bot$ holds for any $t \neq t'$ in $\Gamma_L$.

LEMMA 3.2. *For every ordering $L$ on $V$ and $S \subseteq \Gamma_L$, a DNF $\varphi = \bigvee_{t \in S} t$ represents a double Horn function $f$.*

*Proof.* Since $\varphi$ is obviously Horn, we show that there exists a DNF $\varphi'$ representing $\bar{f}$ which is also Horn, which proves the lemma. Note that $\bigvee_{t \in \Gamma_L} t = \top$ and $tt' = \bot$ holds for all pairs of two terms $t, t' \in \Gamma_L$. This means that $\bar{f} = \varphi' = \bigvee_{t \in \Gamma_L \setminus S} t$, which is clearly Horn. ∎

LEMMA 3.3. *A double Horn function $f$ can be represented by $f = \bigvee_{t \in S} t$ for some ordering $L$ on $V$ and $S \subseteq \Gamma_L$.*

*Proof.* We prove the lemma by induction on the number of variables $n$. In case of $n=0$, $L$ is empty and $\Gamma_L = \{\top\}$ holds. All functions of zero variables, $\top$ and $\bot$, are double Horn, and hence the lemma holds for $n=0$. Assume that the lemma holds for $n=k$, and consider the case of $n=k+1$. We consider two cases.

First, let us assume $f(\mathbf{0})=1$. Since $f$ is Horn, there exists a Horn term $t \leqslant f$ such that $t(\mathbf{0})=1$, which is either $t=\top$ or $t=\bar{x}_j$ for some $x_j$. Since $\top \geqslant \bar{x}_j$ holds for all $j \in V$, there is an index $j \in V$ such that $\bar{x}_j \leqslant f$. Now $f$ can be represented by

$$f = \bar{x}_j \vee f_{(x_j \leftarrow 1)} x_j. \tag{3.1}$$

By Lemma 3.1, $f_{(x_j \leftarrow 1)}$ is double Horn. Hence by the induction hypothesis, there is an ordering $L$ on $V \backslash \{j\}$ and a subset $S \subseteq \Gamma_L$ such that $f_{(x_j \leftarrow 1)} = \bigvee_{t \in S} t$. Define the ordering $L'$ on $V$ by $L' = j, L$ by ordering $j$ before $L$. Let $S' = \{\bar{x}_j\} \cup \{x_j t \mid t \in S\}$. Then we can see $S' \subseteq \Gamma_{L'}$, and by (3.1), we have $f = \bigvee_{t \in S'} t$, which proves the statement for $k+1$.

Similarly, if $f(\mathbf{0})=0$, then there exists an index $j \in V$ such that $\bar{x}_j \leqslant \bar{f}$, and

$$f = \bar{\bar{f}} = \overline{\bar{x}_j \vee \bar{f}_{(x_j \leftarrow 1)} x_j} = f_{(x_j \leftarrow 1)} x_j. \tag{3.2}$$

Then, the lemma can be proved by a similar argument. ∎

LEMMA 3.4. *A function $f$ is double Horn if and only if $f$ can be represented by a DNF $\varphi = \bigvee_{t \in S} t$ for some ordering $L$ on $V$ and $S \subseteq \Gamma_L$.*

*Proof.* Combine Lemmas 3.2 and 3.3. ∎

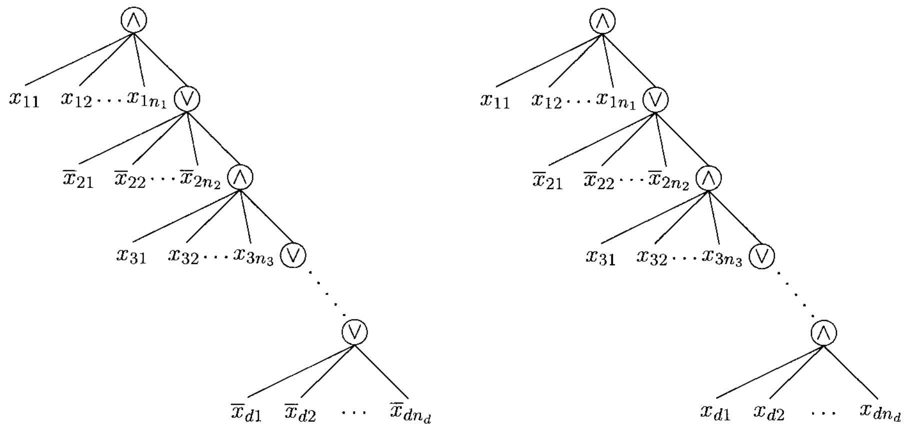
From this lemma, we obtain the following result, which gives an alternative syntactical characterization of double Horn functions. Let $\mathscr{C}_{R-1}$ denote the class of *read-once* functions, where $f$ is called read-once if $f$ can be represented by a read-once formula, i.e., a formula in which every variable occurs at most once (cf. [14, 15, 21, 23, 32, 37]).

THEOREM 3.1. *A function $f$ is double Horn if and only if it can be represented by a read-once formula of the type*

$$\psi = \begin{cases} x_{11}x_{12}\cdots x_{1n_1}(\bar{x}_{21} \vee \bar{x}_{22} \vee \cdots \vee \bar{x}_{2n_2} \\ \qquad \vee (x_{31}x_{32}\cdots x_{3n_3}(\cdots(\bar{x}_{d1} \vee \bar{x}_{d2} \vee \cdots \vee \bar{x}_{dn_d})))) & \text{if } d \text{ is even} \\ x_{11}x_{12}\cdots x_{1n_1}(\bar{x}_{21} \vee \bar{x}_{22} \vee \cdots \vee \bar{x}_{2n_2} \\ \qquad \vee (x_{31}x_{32}\cdots x_{3n_3}(\cdots(x_{d1}x_{d2}\cdots x_{dn_d})))) & \text{if } d \text{ is odd,} \end{cases} \tag{3.3}$$

*where $d \geqslant 0$, $n_1 \geqslant 0$, $n_i \geqslant 1$ for $i = 2, 3, ..., d$, and the variables $x_{11}, x_{12}, ..., x_{dn_d}$ are all different* (*see Fig.* 1). *In particular*, (3.3) *implies $\psi = \bot$ if $d=0$.*

*Proof.* To show the if-part, let $\psi$ be a formula (3.3). Then, by expanding $\psi$, we can see that $\psi$ can be transformed into a Horn DNF. Furthermore, $\bar{\psi}$ can also be transformed into the formula (3.3) (by exchanging $\vee$ and $\wedge$, as well as positive



(i) $d$ is even. (ii) $d$ is odd.

**FIG. 1.** Read-once formulas of a double Horn function.

literals and negative literals). Hence, there is a Horn DNF, which is equivalent to $\bar{\psi}$. Therefore, $\psi$ represents a double Horn function.

We next prove the only-if-part by induction on the number of variables $n$. Check that the statement holds for $n=0$. Assume that it holds for $n=k$, and consider $n=k+1$. By Lemma 3.4, $f$ can be represented by $\varphi = \bigvee_{t\in S} t$ for some ordering $L = i_1, i_2, ..., i_n$ on $V$ and $S \subseteq \Gamma_L$. In the cases where $\varphi$ contains no term (i.e., $f$ is always 0), a single term, or all terms from $\Gamma_L$ (i.e., $f$ is always 1), clearly $f$ is represented by a formula $\psi$ of (3.3). In the remaining case, consider $f' = f_{(x_{i_1} \leftarrow 1)}$. Note that, by Lemma 3.1 and the induction hypothesis, $f'$ can be represented by a formula $\psi'$ of type (3.3), and $\psi' \neq \bot, \top$ holds. If a term $\bar{x}_{i_1}$ occurs in $\varphi$, then choose $\psi = \bar{x}_{i_1} \vee \psi'$, which is a formula (3.3), and represents $f$; otherwise, $x_{i_1}$ appears in all terms $t \in S$ of $\varphi$ (since $S \subseteq \Gamma_L$). Therefore, choose $\psi = x_{i_1}\psi'$, which is again a formula (3.3). In either case, $\psi$ represents $f$. ∎

COROLLARY 3.1. *Every $f \in \mathscr{C}_{DH}$ has the unique prime DNF, which has the form*

$$\varphi = \bigvee_{i=1}^{m} t_1 t_2 \cdots t_i \bar{x}_{\ell_i},$$

*where $t_i$ and $x_{\ell_i}$, $i = 1, 2, ..., m$, are pairwise disjoint positive terms (in this case, literals $x_{\ell_i}$ are also regarded as terms), and some of $t_1, t_2, ..., t_m$ and $x_{\ell_m}$ may be empty. In particular, (3.4) implies $\varphi = \bot$ if $m = 0$. Conversely, every such formula $\varphi$ represents an $f \in \mathscr{C}_{DH}$.*

*Proof.* By Lemma 3.4, or by expanding (3.3) in Theorem 3.1, we have a Horn DNF $\varphi$ of (3.4). To prove the unique primality of $\varphi$ of (3.4), let $t$ be a prime implicant of $f_\varphi$, where $f_\varphi$ is the function $\varphi$ represents. Then, as noted in Section 2, there is a consensus sequence $t^{(1)}, t^{(2)}, ..., t^{(m)} = t$ of terms such that each $t^{(k)}$ is either in $\varphi$ (i.e., $t^{(k)} = t_1 t_2 ... t_i \bar{x}_{l_i}$ for some $i$) or the consensus of two terms $t^{(k_1)}$ and $t^{(k_2)}$ such that $k_1, k_2 < k$. However, $\varphi$ of (3.4) has no pair of terms $t^{(i)} = t_1 t_2 \cdots t_i \bar{x}_{\ell_i}$ and

$t^{(j)} = t_1 t_2 \cdots t_j \bar{x}_{\ell_j}$ such that $P(t^{(i)}) \cap N(t^{(j)}) = \{l\}$ and $N(t^{(i)}) \cap P(t^{(j)}) = \emptyset$, implying that no consensus operation is applicable to $\varphi$. This means that $t$ is a prime implicant of $f_\varphi$ if and only if $t$ is in $\varphi$, and hence $\varphi$ is the unique prime DNF of $f_\varphi$. ∎

This corollary also says that a double Horn function $f$ has a short DNF in the sense that formula $\varphi$ of (3.4) has at most $n$ terms and its length satisfies $|\varphi| \leqslant n^2$. The unique prime DNF $\varphi$ of (3.4) can be computed form the read-once formula of (3.3) by simply expanding it. The required time for this is $O(n^2)$. These results also apply to $\bar{f}$ since the definition of a double Horn function is symmetric between $f$ and $\bar{f}$. Obtaining the unique prime DNF $\varphi'$ for $\bar{f}$ from the unique prime DNF $\varphi$ of $f$ is also straightforward. As implicit in the proof of Lemma 3.2, this can be done by first finding an ordering $L$ on $V = \{1, 2, ..., n\}$ such that $\varphi$ can be regarded as

$$\varphi = \bigvee_{t \in S} t$$

for some $S \subseteq \Gamma_L$, and then by defining

$$\varphi' = \bigvee_{t \in \Gamma_L \setminus S} t.$$

We note that the dual $f^d$ of a double Horn function $f$ also has the unique prime DNF $\varphi''$, which is obtained from $\varphi'$ by complementing all literals. This DNF is not Horn in general (recall that the dual of a double Horn function may not be Horn any more). Summarizing these, we establish the next corollary.

COROLLARY 3.2. *Let $f$ be a double Horn function of $n \geqslant 1$ variables. Then $f$, $\bar{f}$, and $f^d$ have short unique prime DNFs, respectively, which contain at most $n$ terms, and whose lengths are at most $n^2$. The prime DNF $\varphi$ of $f$ can be computed from its read-once formula* (3.3) *in $O(n^2)$ time. Also prime DNFs for $\bar{f}$ and $f^d$ can be obtained in $O(n^2)$ time, respectively.*

### *3.3. Recognition of Double Horn Functions*

We consider the problem of recognizing a double Horn function from a DNF $\varphi$. In particular, we describe an algorithm that solves this problem in polynomial time for classes of formulas that satisfy certain properties and show also that the problem is intractable in general.

LEMMA 3.5. *Let $f$ be a function of $n \geqslant 1$ variables. Then $f \in \mathscr{C}_{DH}$ if and only if* (i) *either $\bar{x}_j \leqslant f$ or $\bar{x}_j \leqslant \bar{f}$ holds for some $j$, and* (ii) *$f_{(x_j \leftarrow 1)} \in \mathscr{C}_{DH}$ for all such $j$.*

*Remark* 3.1. Note that, if a function $f$ satisfies $\bar{x}_j \leqslant f$, then it has a formula $\bar{x}_j \vee \varphi$ for some $\varphi$, and if $f$ satisfies $\bar{x}_j \leqslant \bar{f}$, then it has a formula $x_j \varphi$. ∎

*Proof.* To prove the only-if-part, assume that $f$ is double Horn. Then consider vector $\mathbf{0}$. If $f(\mathbf{0}) = 1$ (resp., $f(\mathbf{0}) = 0$), then there exists a $j$ such that $\bar{x}_j \leqslant f$ (resp., $\bar{x}_j \leqslant \bar{f}$) by the discussion in the proof of Lemma 3.3. Thus (i) holds. Furthermore, Lemma 3.1 tells that (ii) holds.

Conversely, let us assume that (i) and (ii) hold. Let $W_1 = \{j \mid \bar{x}_j \leqslant f\}$ and $W_2 = \{j \mid \bar{x}_j \leqslant \bar{f}\}$. Since $f\bar{f} = \perp$, it is easy to see that either (I) ($W_1 \neq \emptyset$ and $W_2 = \emptyset$) or (II) ($W_1 = \emptyset$ and $W_2 \neq \emptyset$) holds. We consider case (I) only, since the other case is similar. Then $f$ and $\bar{f}$ can be represented by

$$f = \bar{x}_j \vee f_{(x_j \leftarrow 1)} x_j \qquad \text{and} \qquad \bar{f} = \overline{f_{(x_j \leftarrow 1)}} x_j \tag{3.5}$$

for any $j \in W_1$. By (ii), $f_{(x_j \leftarrow 1)}$ is double Horn, that is, both $f_{(x_j \leftarrow 1)}$ and $\overline{f_{(x_j \leftarrow 1)}}$ can be represented by Horn DNFs. Therefore $f$ and $\bar{f}$ can be represented by Horn DNFs. ▌

Let a function $f$ be represented by a formula $\varphi$. Then the above conditions $\bar{x}_j \leqslant \varphi$ and $\bar{x}_j \leqslant \bar{\varphi}$ are equivalent to $\varphi_{(x_j \leftarrow 0)} \equiv \top$ and $\varphi_{(x_j \leftarrow 0)} \equiv \perp$, respectively. Whether the condition $\varphi' \equiv \top$ and $\varphi' \equiv \perp$ can be checked in polynomial time or not depends on how formula $\varphi'$ is given. If $\varphi'$ is a DNF, for example, then it is trivial to check if $\varphi' \equiv \perp$ holds, but it may not be trivial to check if $\varphi' \equiv \top$ holds (problem TAUTOLOGY). If $\varphi'$ is a Horn DNF, both $\varphi' \equiv \top$ and $\varphi' \equiv \perp$ can be checked in linear time [6].

The previous lemma can be exploited for an algorithm CHECK-DH recognizing a double Horn function, which proceeds as follows. It checks the condition (i), by testing whether the current $\varphi$ satisfies $\varphi_{(x_j \leftarrow 0)} \equiv \top$ or $\varphi_{(x_j \leftarrow 0)} \equiv \perp$ for some $j$, and if so, applies the decomposition of $\varphi = \bar{x}_j \vee \varphi_{(x_j \leftarrow 1)}$ or $\varphi = x_j \varphi_{(x_j \leftarrow 1)}$ recursively, until finally condition $\varphi_{(x_j \leftarrow 1)} \equiv \top$ or $\perp$ is reached.

Since CHECK-DH may need to check all the current variables $x_j$ for the condition of $\varphi_{(x_j \leftarrow 0)} \equiv \top$ or $\varphi_{(x_j \leftarrow 0)} \equiv \perp$ in each iteration, and the level of recursion is at most $n$, $O(n^2)$ is an upper bound on the number of tests for $\varphi \equiv \top$ and $\varphi \equiv \perp$ in CHECK-DH. For a more detailed formulation, see [9]. This establishes the next theorem.

Theorem 3.2. *Let $\mathscr{F}$ be a class of formulas, which is closed under assignments and for which checking if $\varphi \equiv \top$ and $\varphi \equiv \perp$, respectively, can be done in $O(t(n, |\varphi|))$ time for any $\varphi \in \mathscr{F}$, where $n$ is the number of variables.*[1] *Then, deciding whether a given $\varphi \in \mathscr{F}$ represents a double Horn function can be done in $O(n^2 t(n, |\varphi|))$ time.*

Notice that if $t(n, |\phi|)$ is a polynomial, then the recognition of a double Horn function from a given formula $\phi \in \mathscr{F}$ is polynomial.

Example 3.1. Let us apply algorithm CHECK-DH to a Horn DNF

$$\varphi^{(1)} = x_1 x_2 \bar{x}_3 \vee \bar{x}_2 \vee x_1 x_2 x_4 \vee x_1 \bar{x}_3 x_4.$$

Since $\bar{x}_2 \leqslant \varphi^{(1)}$ (other $x_j$ satisfy neither $\bar{x}_j \leqslant \varphi^{(1)}$ nor $\bar{x}_j \leqslant \overline{\varphi^{(1)}}$), $\varphi^{(1)} \equiv \bar{x}_2 \vee \varphi^{(2)}$ holds, where $\varphi^{(2)} = \varphi^{(1)}_{(x_2 \leftarrow 1)} = x_1 \bar{x}_3 \vee x_1 x_4 \vee x_1 \bar{x}_3 x_4$. Since $\varphi^{(2)}$ satisfies $\bar{x}_1 \leqslant \overline{\varphi^{(2)}}$, we have $\varphi^{(2)} \equiv x_1 \varphi^{(3)}$, where $\varphi^{(3)} = \varphi^{(2)}_{(x_1 \leftarrow 1)} = \bar{x}_3 \vee x_4 \vee \bar{x}_3 x_4$. In the next iteration,

[1] It is assumed that $t(n, |\varphi|)$ is monotonic in both arguments, i.e., fewer variables or shorter formulas do not increase running time, which applies to all reasonable classes of formulas.

we see that $\varphi^{(3)}$ satisfies $\bar{x}_3 \leqslant \varphi^{(3)}$, and we have $\varphi^{(3)} \equiv \bar{x}_3 \vee \varphi^{(4)}$, where $\varphi^{(4)} = \varphi^{(3)}_{(x_3 \leftarrow 1)} = x_4$. Finally, $\varphi^{(4)} \equiv x_4 \varphi^{(5)}$ holds, where $\varphi^{(5)} \equiv \top$. Therefore, $\varphi^{(5)} \equiv \top$, $\varphi^{(4)} \equiv x_4$, $\varphi^{(3)} \equiv \bar{x}_3 \vee x_4$, $\varphi^{(2)} \equiv x_1(\bar{x}_3 \vee x_4)$, and $\varphi^{(1)} \equiv \bar{x}_2 \vee x_1(\bar{x}_3 \vee x_4)$. Thus $\varphi^{(1)}$ represents a double Horn function.

If we restrict ourselves to the class of Horn formulas, then it is known that $t(n, |\varphi|) = |\varphi|$ holds [6]. Therefore, the complexity in Theorem 3.2 becomes $O(n^2 |\varphi|)$. This complexity can be further improved to linear time $O(|\varphi|)$ by exploiting the data structure developed for handling Horn DNFs [6], and by devising clever and efficient bookkeeping of data. The details are, however, technical, and omitted; the interested readers are requested to consult [9].

Finally, we note that, for a general class of formulas, for which $t(n, |\varphi|)$ in Theorem 3.2 may not be polynomial, it is very unlikely to have a polynomial time algorithm, because we have the following negative result.

THEOREM 3.3. *Given an arbitrary DNF $\varphi$, checking if $\varphi$ represents a double Horn function is coNP-complete.*

*Proof.* The problem is in coNP, since a guess for vectors $u$ and $v$ such that either $\varphi(u) = \varphi(v) = 0$ and $\varphi(u \wedge v) = 1$, or $\varphi(u) = \varphi(v) = 1$ and $\varphi(u \wedge v) = 0$, which proves that $\varphi$ is not double Horn, can be easily verified in polynomial time.

To show the hardness, let $\psi$ be a DNF involving variables $x_1, x_2, ..., x_n$, and define a DNF $\varphi$ by $\varphi = \psi \vee \bar{x}_{n+1} \bar{x}_{n+2}$. We claim that $\varphi$ is double Horn if and only if $\psi$ is a tautology (i.e., $\psi \equiv \top$). If $\psi$ is a tautology, then clearly, $\varphi$ is a tautology, and hence $\varphi$ is double Horn. Conversely, if $\psi \not\equiv \top$, then $\bar{x}_{n+1} \bar{x}_{n+2}$ is a prime implicant of $\varphi$, since $\varphi_{(x_{n+1} \leftarrow 0)}, \varphi_{(x_{n+2} \leftarrow 0)} \not\equiv \top$ (i.e., neither $\bar{x}_{n+1}$ nor $\bar{x}_{n+2}$ is an implicant of $\varphi$). Since such $\varphi$ cannot be represented by a DNF of (3.4), Corollary 3.1 tells that $\varphi$ does not represent a double Horn function, which completes our claim.

Since deciding whether a given DNF $\psi$ is a tautology is known to be co-NP-hard [12], it follows that deciding whether a given DNF $\psi$ is double Horn is co-NP-hard. ∎

## 4. DOUBLE HORN EXTENSIONS

In this section, we study double Horn extensions of partially defined Boolean functions. We first present a criterion for the existence of a double Horn extension of a pdBf $(T, F)$ in terms of a simple declarative condition on intersection closures. This criterion leads to a polynomial time algorithm for the double Horn extension problem. This positive result is complemented by the negative result that finding a shortest double Horn extension, i.e., a double Horn extension that has a shortest DNF (or even arbitrary) formula $\varphi$ is NP-hard. Finally, we consider enumerating all double Horn extensions $\varphi_1, \varphi_2, ...$ of $(T, F)$ with polynomial delay. As a consequence, we obtain that deciding whether a pdBf $(T, F)$ has the unique double Horn extension, i.e., $(T, F)$ implicitly defines the unique double Horn function $f$, is solvable in polynomial time.

### 4.1. *Existence of a Double Horn Extension*

The existence of a double Horn extension is described by a surprisingly simple criterion. Revisiting the definition of double Horn, $Cl_\wedge(T(f)) = T(f)$ and $Cl_\wedge(F(f)) = F(f)$, we obtain an immediate necessary condition

$$Cl_\wedge(T) \cap Cl_\wedge(F) = \varnothing \tag{4.6}$$

for the existence of a double Horn extension of a pdBf $(T, F)$. This condition also turns out to be sufficient.

THEOREM 4.1. *Let $(T, F)$ be a pdBF. Then $(T, F)$ has an extension $f \in \mathscr{C}_{DH}$ if and only if $Cl_\wedge(T) \cap Cl_\wedge(F) = \varnothing$.*

*Proof.* To prove the only-if-part, assume that $f \in \mathscr{C}_{DH}$ is an extension of $(T, F)$. Then clearly $Cl_\wedge(T(f)) \cap Cl_\wedge(F(f)) = \varnothing$ holds. Since $T \subseteq Cl_\wedge(T(f))$ and $F \subseteq Cl_\wedge(F(f))$, this implies $Cl_\wedge(T) \cap Cl_\wedge(F) = \varnothing$.

For the converse direction, assume that $Cl_\wedge(T) \cap Cl_\wedge(F) = \varnothing$, but no extension $f \in \mathscr{C}_{DH}$ exists. We will derive a contradiction. Let $T^*, F^* \subseteq \{0, 1\}^n$ be maximal subsets (with respect to inclusion) such that $T \subseteq T^*$, $F \subseteq F^*$, and $Cl_\wedge(T^*) \cap Cl_\wedge(F^*) = \varnothing$ (but $Cl_\wedge(T^*) \cup Cl_\wedge(F^*) \neq \{0, 1\}^n$ by the assumption). Maximality of $T^*$ and $F^*$ implies that $T^* = Cl_\wedge(T^*)$ and $F^* = Cl_\wedge(F^*)$, and hence $v^{(1)} = \bigwedge_{v \in T^*} v \in T^*$ and $v^{(2)} = \bigwedge_{v \in F^*} v \in F^*$ satisfy $v^{(1)} \neq v^{(2)}$. Consider two possible cases:

*Case* 1. $v^{(1)}$ and $v^{(2)}$ are comparable. We assume $v^{(1)} \leqslant v^{(2)}$ without loss of generality. By the maximality of $T^*$, it follows that every vector $w^{(1)}$ such that $w^{(1)} \ngeqslant v^{(2)}$ is contained in $T^*$, and hence $v^{(1)} = \mathbf{0}$. Let $T_1^* = \{w \in T^* \mid w > v^{(2)}\}$. If $T_1^* = \varnothing$, then the maximality of $F^*$ implies that every $w^{(2)}$ such that $w^{(2)} \geqslant v^{(2)}$ satisfies $w^{(2)} \in F^*$. This means $T^* \cup F^* = \{0, 1\}^n$, which is a contradiction. Otherwise, let $v^{(3)} = \bigwedge_{v \in T_1^*} v$. Note that $v^{(2)} < v^{(3)}$ must hold by assumption $Cl_\wedge(T^*) \cap Cl_\wedge(F^*) = \varnothing$. Then, by a similar argument, it follows that every vector $w^{(2)}$ such that $w^{(2)} \geqslant v^{(2)}$ but $w^{(2)} \ngeqslant v^{(3)}$ must be in $F^*$. Continuing this argument, we obtain a finite chain $v^{(1)} < v^{(2)} < \dots < v^{(l)}$ in $\{0, 1\}^n$ such that all vectors $w^{(i)}$ satisfying $w^{(i)} \geqslant v^{(i)}$ but $w^{(i)} \ngeqslant v^{(i+1)}$ belong to $T^*$ (resp., $F^*$) if $v^{(i)} \in T^*$ (resp., $F^*$), where $v^{(l+1)} = \mathbf{1}$ is assumed for convenience. Consequently, $T^* \cup F^* = \{0, 1\}^n$. This implies that there exists a double Horn extension $f$ of $(T, F)$ such that $T(f) = T^*$ and $F(f) = F^*$, which is a contradiction.

*Case* 2. $v^{(1)} \nleqslant v^{(2)}$, $v^{(1)} \ngeqslant v^{(2)}$. Then obviously $\mathbf{0} \notin T^* \cup F^*$, but $Cl_\wedge(T^* \cup \{\mathbf{0}\}) \cap F^* = \varnothing$ holds. This is a contradiction to the maximality of $T^*$ and $F^*$. ∎

This criterion shows a similarity between the existence problems of double Horn extensions and arbitrary Horn extensions, because $T \cap Cl_\wedge(F) = \varnothing$ is a necessary and sufficient condition for the latter problem [3, 29].

We remark that the proof of Theorem 4.1 provides another characterization of double Horn functions. Namely, for any double Horn function $f$ there is a collection of vectors $v^{(1)} < v^{(2)} \cdots < v^{(k)}$ and a $b \in \{0, 1\}$ such that $f(x)$ can be expressed as:

if $x \not\geqslant v^{(1)}$ then $f(x) = b$,

else

if $x \not\geqslant v^{(2)}$ then $f(x) = 1 - b$,

else

if $x \not\geqslant v^{(3)}$ then $f(x) = b$,

...

### 4.2. *Finding a Double Horn Extension*

The condition in Theorem 4.1 can be checked in polynomial time, exploiting the following observation: A vector $v$ belongs to $Cl_\wedge(T) \cap Cl_\wedge(F)$ only if $v \geqslant v_{TF}$, where $v_{TF} = (\bigwedge_{u \in T} u) \vee (\bigwedge_{u \in F} u)$. This gives rise to a natural algorithm which checks the condition $Cl_\wedge(T) \cap Cl_\wedge(F) = \emptyset$, by starting with a test $(\bigwedge_{u \in T} u) \neq (\bigwedge_{u \in F} u)$ and then iteratively reducing $T$ and $F$ to the vectors $\geqslant v_{TF}$ [9]. We now show below that this algorithm can be adapted to run in $O(n^2(|T| + |F|))$ time. However, a faster linear time $O(n(|T| + |F|))$ algorithm is possible, but its details are not given here for simplicity; we refer to [9] for the interested reader.

The algorithm recursively exploits the decompositions of Lemma 3.5, which are always possible for double Horn functions.

ALGORITHM. DH-EXTENSION
*Input.* A pdBf $(T, F)$, where $T, F \subseteq \{0, 1\}^n$.
*Output.* A read-once formula $\psi$ of (3.3) for $f$ if there is a double Horn extension $f$ of $(T, F)$; otherwise "No".

**Step 1.** Call DH-AUX($T, F, \{1, 2, \cdots, n\}$) and output the obtained result; Halt.

*Procedure* DH-AUX($T, F, I$)
*Input.* $T, F \subseteq \{0, 1\}^n$ and a set $I \subseteq \{1, 2, \cdots, n\}$ such that the pair of projections $(T[I], F[I])$ is a pdBf.
*Output.* A read-once formula $\psi$ of (3.3) for $f$ on variables $x_i$, $i \in I$, if there is a double Horn extension $f$ of $(T[I], F[I])$; otherwise "No."

**Step 1.** **if** $T[I] = \emptyset$ **then** return $\psi := \bot$ (exit) (* there are no true vectors *)
**elsif** $F[I] = \emptyset$ **then** $\psi := \top$ (exit) (* there are no false vectors *)
**fi**;

**Step 2.** $I^+ := \bigcap_{v \in T[I]} ON(v)$; (* each $i \in I^+$ tells that $\psi$ can be $x_i(\cdots)$ *)
**if** $I^+ = \emptyset$ **then** go to Step 3
**else** (* assume that $\psi$ has form $\bigwedge_{i \in I^+} x_i(\cdots)$, and decompose *)
$T' := T$, $F' := F \backslash \{w \in F \mid OFF(w) \cap I^+ \neq \emptyset\}$ and $I' := I \backslash I^+$;
Call DH-AUX($T', F', I'$);
**if** the answer is $\psi'$ $(\neq \top)$ **then** return $\psi := \bigwedge_{i \in I^+} x_i \wedge \psi'$ and exit
**elsif** the answer is $\top$ **then** return $\psi := \bigwedge_{i \in I^+} x_i$ and exit

**else** return "No" and exit (* decomposition failed, and there is no extension *)
**fi**
**fi**;

**Step 3.** $J^+ := \bigcap_{w \in F[I]} ON(w)$; (* each $i \in J^+$ tells that $\psi$ can be $\bar{x}_i \vee (\cdots)$ *)
**if** $J^+ = \emptyset$ **then** (* decomposition failed, and there is no extension *)
return "No" and exit
**else** (* assume $\psi$ has form $\bigvee_{i \in J^+} \bar{x}_i \vee (\cdots)$, and decompose *)
$T' := T \backslash \{v \in T \mid OFF(v) \cap J^+ \neq \emptyset\}$, $F' := F$ and $I' := I \backslash J^+$;
Call DH-AUX($T'$, $F'$, $I'$);
**if** the answer is $\psi'$ ($\neq \perp$) **then** return $\psi := \bigvee_{i \in J^+} \bar{x}_i \vee \psi'$ and exit
**elsif** the answer is $\perp$ **then** return $\psi := \bigvee_{i \in J^+} \bar{x}_i$ and exit
**else** return "No" and exit (* decomposition failed *)
**fi**
**fi**.

Observe that in consecutive recursive calls of DH-AUX, the execution alternates between steps 2 and 3.

EXAMPLE 4.1. Let us apply algorithm DH-EXTENSION to a pdBf $(T, F)$ defined by $T = \{(1111), (1010)\}$ and $F = \{(1101), (1110)\}$. As we see in the following, it outputs a double Horn extension $\psi = x_1 x_3(\bar{x}_2 \vee x_4)$ of $(T, F)$.

(*DH-EXTENSION*) *Step* 1. Call DH-AUX with the above $T, F$ and $I = \{1, 2, 3, 4\}$.

(*DH-AUX* (1)) *Step* 1. Since $T[I] \neq \emptyset$, $F[I] \neq \emptyset$ and $I \neq \emptyset$, the computation continues.
*Step* 2. $I^+ := \{1, 3\}$, $T' := \{(1111), (1010)\}$, $F' = \{(1110)\}$, and $I' = \{2, 4\}$. Call DH-AUX($T'$, $F'$, $I'$).

(*DH-AUX* (2)) *Step* 1. Continue to Step 2.
*Step* 2. $I^+ := \emptyset$, hence continue to Step 3.
*Step* 3. $J^+ := \{2\}$, $T' := \{(1111)\}$, $F' := \{(1110)\}$, $I' := \{4\}$. Call DH-AUX($T'$, $F'$, $I'$).

(*DH-AUX* (3)) *Step* 1. Continue to Step 2.
*Step* 2. $I^+ := \{4\}$, $T' := \{(1111)\}$, $F' := \emptyset$, $I' := \emptyset$. Call DH-AUX($T'$, $F'$, $I'$).

(*DH-AUX* (4)) *Step* 1. Since $F[I] = \emptyset$, it returns $\psi = \top$.

(*DH-AUX* (3)) As $\psi = \top$, it returns $\psi = x_4$.

(*DH-AUX* (2)) As $\psi' = x_4$, it returns $\psi = \bar{x}_2 \vee x_4$.

(*DH-AUX* (1)) As $\psi' \neq \top$, it returns $\psi = x_1 x_3(\bar{x}_2 \vee x_4)$.

(*DH-EXTENSION*) Output $\psi = x_1 x_3(\bar{x}_2 \vee x_4)$.

THEOREM 4.2. *Given a pdBf* $(T, F)$, *where* $T, F \subseteq \{0, 1\}^n$, *algorithm* DH-EXTENSION *correctly finds a read-once formula* $\psi$ *of* (3.3) *for a double Horn extension* (*if such an extension exists*) *or outputs "No"* (*if no such extension exists*), *in* $O(n^2(|T| + |F|))$ *time.*

*Proof.* For the correctness part, we prove by induction on $|I|$ that DH-AUX is correct, which clearly implies the correctness of DH-EXTENSION.

If $|I| = 0$, then either $T[I]$ or $F[I]$ is empty and the returned value is obviously correct. Assume that DH-AUX is correct for $|I| \leqslant k-1$, and then consider the case of $|I| = k$.

*Case* 1. $I^+ \neq \emptyset$. Notice that all vectors in $T' \cup F'$ have value 1 at all components $i \in I^+$. By the induction hypothesis, DH-AUX$(T', F', I')$ tells correctly whether $(T'[I'], F'[I'])$, where $I' = I \backslash I^+$, has a double Horn extension. If DH-AUX$(T', F', I')$ returns $\psi'$ $(\neq \top)$, then by the definition of $T'$ and $F'$, the constructed formula $\psi$ $(= \bigwedge_{i \in I^+} x_i \wedge \psi')$ satisfies $T[I] \subseteq T(\psi)$ and $F[I] \cap T(\psi) = \emptyset$. Thus $\psi$ represents an extension of $(T[I], F[I])$. Furthermore, by Theorem 3.1, this $\psi$ represents a double Horn function. Hence the returned value is correct. Similarly, in case that DH-AUX$(T', F', I')$ returns $\top$, we can show that DH-AUX$(T, F, I)$ returns the correct value. Finally, consider the case that DH-AUX$(T', F', I')$ returns "No." For this case, assume toward a contradiction that $(T[I], F[I])$ has a double Horn extension $f$. By Lemma 3.1, $f_A \in \mathscr{C}_{DH}$ holds for the assignment $A = (x_i \leftarrow 1 \mid i \in I^+)$. Therefore, $(T'[I'], F'[I'])$ has a double Horn extension. Thus, by the induction hypothesis, DH-AUX$(T', F', I')$ does not output "No." This is a contradiction.

*Case* 2. $J^+ \neq \emptyset$. Analogously to case 1, it is shown that the returned value is correct.

*Case* 3. $I^+ \cup J^+ = \emptyset$. This means that for each component $i \in I$, there are two vectors $v \in T[I]$ and $w \in F[I]$ such that $v_i = w_i = 0$. Thus no extension $f$ of $(T[I], F[I])$ satisfies $\bar{x}_i \leqslant f$ or $\bar{x}_i \leqslant \bar{f}$ for any $i \in I$. Therefore, Lemma 3.5 tells that there is no double Horn extension of $(T[I], F[I])$. Therefore, the returned value "No″ is correct.

Finally, concerning the time complexity, it is easily checked that the body of DH-AUX can be executed in $O(n(|T| + |F|))$ time. Since the recursion is linear and its depth is bounded by $n$, it follows that DH-EXTENSION runs in $O(n^2(|T| + |F|))$ time. ∎

We remark here that the running time of the above algorithm can be further improved to $O(n(|T| + |F|))$ time (i.e., linear time with respect to input length). Basically, the method is to use lists with cross-references and counters to achieve that the same bit of the input is looked up only once after a linear-time initialization phase. The details are however omitted, as they can be found in [9].

### *4.3. PAC Learnability of Double Horn Functions*

The above result on the extension problem enables us to derive as a corollary, applying a general result of [2], that the class of double Horn functions is PAC-learnable.

A hypothesis class $\mathscr{H}$ of Boolean functions is *probably approximately correct* (*PAC*) *learnable* (with respect to a fixed encoding of the hypotheses) [2, 37], if there exist a learning algorithm for $\mathscr{H}$ and a minimal sample size $m(\varepsilon, \delta, n)$ polynomial in $1/\varepsilon$, $1/\delta$, and $n$, where $n$ is the size of the encoding of the hypothesis, such that

(a) for all $f \in \mathscr{H}$ which have encoding of size at most $n$, and all distributions $P$ on the sample space $X$ (in our context, the set of all Boolean vectors), given $m(\varepsilon, \delta, n)$ observations of $f$ (in our terms, a pdBF $(T, F)$ which has $f$ as a possible extension), the algorithm produces a hypothesis with error at most $\varepsilon$ with probability at least $1-\delta$, and

(b) the algorithm produces its hypothesis in time polynomial in the length of the given sample.

As shown in [2], PAC learnability of a class $\mathscr{H}$ is guaranteed if it possesses a so-called Occam-algorithm. Formally, an *Occam*-algorithm [2] for $\mathscr{H}$ with constant parameters $c \geqslant 1$ and $0 \leqslant \alpha < 1$ is a learning algorithm which

(i) produces a hypothesis of complexity at most $n^c m^\alpha$ when given a sample of size $m$ of any function in $\mathscr{H}$ of complexity at most $n$, and

(ii) runs in time polynomial in the length of the sample.

Here, the complexity of a function is its size with respect to the chosen encoding, measured in bits.

It is easy to see that algorithm DH-EXTENSION is an Occam-algorithm. We thus obtain the following result.

THEOREM 4.3. *The class $\mathscr{C}_{DH}$ is PAC learnable.*

The underlying reason for this result is the fact that the class $\mathscr{C}_{DH}$ is small and has only $O(2^{p(n)})$ functions, where $p(n)$ is a polynomial, on $n$ variables (for a detailed analysis, see Section 5.3).

Notice that Theorem 4.3 is known, as it appears to be a special case of a previously developed learning result on nested differences of concepts [19]. In that paper, the authors introduced a framework for constructing learning algorithms and employed a master algorithm TOTAL RECALL, which is capable of learning any concept class whose members can be expressed as nested differences of concepts from an intersection-closed class. The class $\mathscr{C}_{DH}$ fits this framework.

For example, it is easy to see that the double Horn function $f = x_1 x_2 (\bar{x}_3 \vee \bar{x}_4 x_5)$ can be expressed as a nested difference $x_1 x_2 \backslash (x_3 x_4 \backslash x_5)$ (where the set operation is taken over the true vectors of $x_1 x_2$, $x_3 x_4$, and $x_5$, respectively).

Furthermore, it turns out that our algorithm DH-EXTENSION is a close relative of algorithm TOTAL RECALL. However, notice the following differences. First, algorithm TOTAL RECALL outputs a hypothesis for a function $f$ which is *known* to be from $\mathscr{C}_{DH}$, while the aim of algorithm DH-EXTENSION, which was independently developed for a different purpose, has to *decide* whether membership of $f$ in $\mathscr{C}_{DH}$ is possible (and output some extension in case it is). Moreover, the refined version of our algorithm is linear time which is not discussed in [19]. Finally, we have an extended algorithm for enumerating all different extensions in $\mathscr{C}_{DH}$ with polynomial delay, which allows to solve the exact learning problem in a batch-mode setting in polynomial time (see Section 4.5).

### 4.4. Computing a Shortest Double Horn Extension

Since computing one double Horn extension of a pdBf $(T, F)$ is fast, the following natural question arises: How complex is computing a double Horn extension with the shortest DNF ? This problem is intractable, however.

THEOREM 4.4. *Given a pdBf* $(T, F)$, *computing a shortest Horn DNF* (*or even any formula*) $\varphi$ *representing a double Horn extension of* $(T, F)$ *is* NP-*hard.*

*Proof.* We reduce the classical problem of deciding whether a graph $G=(V, E)$ has a vertex cover of size at most $k$ [12] to this problem. Suppose that $V=\{1, 2, ..., n\}$, and define a pdBf $(T, F)$ by $T=\{\mathbf{1}\}$ and $F=\{x^{V\setminus\{i, j\}} \mid \{i, j\} \in E\}$. We claim that $(T, F)$ has a double Horn DNF $\varphi$ that contains at most $k$ literals if and only if $G$ has a vertex cover of size at most $k$. Indeed, if $C \subseteq V$ is a vertex cover of $G$, then $\varphi=\bigwedge_{i \in C} x_i$ represents an extension, which is clearly double Horn. This proves the if-part. To show the only-if-part, assume that $G$ has no vertex cover of size at most $k$, and let $\varphi$ represent any Horn (in particular, double Horn) extension. Then $t(\mathbf{1})=1$ holds for some Horn implicant $t$ of $\varphi$. Let $t=\bigwedge_{j \in P(t)} x_j \times \bigwedge_{j \in N(t)} \bar{x}_j$. Since $N(t)=\varnothing$ must hold to satisfy $t(\mathbf{1})=1$, $P(t)$ is a vertex cover of $G$. Hence $|\varphi| \geqslant |P(t)| \geqslant k+1$, a contradiction. This proves the claim, from which the result clearly follows. ∎

### 4.5. Computing all Double Horn Extensions

In this section, we consider the problem of computing all double Horn extensions of a pdBf$(T, F)$ and describe an algorithm that outputs the corresponding read-once formulas with *polynomial delay* [22]; 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 the start (resp., before halt) of the algorithm. A polynomial delay algorithm is of course *polynomial total time* [22], that is, its running time is polynomial in the size of the input and output. These concepts take into account that the output size can be much larger (in particular, exponentially longer) than the input size.

The algorithm, ALL-DH-EXTENSIONS, is a slight variant of algorithm DH-EXTENSION in Subsection 4.2. It builds read-once formulas $\gamma$ of all extensions from left to right in the order to be defined later by making use of Lemma 3.5. Each extension output has a maximum common prefix with the immediately preceding extension. The formula $\gamma$ of (3.3) is represented by a list of its literals, which is preceded for technical convenience by a special literal $x_0$ that represents $\top$. For example, $\gamma=\bar{x}_1 \vee x_3\bar{x}_2$ is represented by an ordered list of literals $L=x_0, \bar{x}_1, x_3, \bar{x}_2$, which is obtained by simply listing all the literals in $\gamma$ in the order of their appearance, after the initial special literal $x_0$. We call this the *list representation* of $\gamma$. It is easy to see that there is a one-to-one correspondence between formulas $\gamma$ of (3.3) and ordered lists of literals. (In this case, we regard formulas $\gamma=\bar{x}_1 \vee \bar{x}_2$ and $\gamma=\bar{x}_2 \vee \bar{x}_1$ as being different, for example, even if they represent the same functions.)

EXAMPLE 4.2. Let us consider $T=\{(111),(101)\}$ and $F=\{(110)\}$ (i.e., $n=3$). It will turn out later that this $(T, F)$ has eight double Horn extensions, represented by the following formulas of (3.3):

$$\begin{aligned}
\gamma^{(1)} &= x_1 x_3, && (L=x_0, x_1, x_3)\\
\gamma^{(2)} &= x_1(\bar{x}_2 \vee x_3), && (L=x_0, x_1, \bar{x}_2, x_3)\\
\gamma^{(3)} &= x_3, && (L=x_0, x_3)\\
\gamma^{(4)} &= x_3(\bar{x}_2 \vee x_1), && (L=x_0, x_3, \bar{x}_2, x_1)\\
\gamma^{(5)} &= \bar{x}_1 \vee x_3, && (L=x_0, \bar{x}_1, x_3)\\
\gamma^{(6)} &= \bar{x}_1 \vee \bar{x}_2 \vee x_3, && (L=x_0, \bar{x}_1, \bar{x}_2, x_3)\\
\gamma^{(7)} &= \bar{x}_2 \vee x_1 x_3, && (L=x_0, \bar{x}_2, x_1, x_3)\\
\gamma^{(8)} &= \bar{x}_2 \vee x_3. && (L=x_0, \bar{x}_2, x_3).
\end{aligned}$$

Figure 2 shows the computational tree built by ALL-DH-EXTENSIONS, which lists all double Horn extensions of $(T, F)$. Scanning this tree from left to right in the depth-first manner, extensions are generated corresponding to the paths from root $x_0$ to the nodes marked with "*"; e.g., the leftmost path $x_0, x_1, x_3^*$ represents $\gamma^{(1)}$, the second path $x_0, x_1, \bar{x}_2, x_3^*$ represents $\gamma^{(2)}$, the third path $x_3^*$ represents $\gamma^{(3)}$, ..., and the rightmost path $x_0, \bar{x}_2, x_3^*$ represents $\gamma^{(8)}$.

Before describing the main algorithm, we first explain an algorithm for testing whether a given pdBf $(T[I], F[I])$, where $T, F \subseteq \{0,1\}^n$ and $I \subseteq \{1, 2, ..., n\}$, has a double Horn extension $f$ under some additional restriction saying that the starting literals must be chosen from a given set of positive (or negative) literals.

More precisely, let $P$ be a set of positive literals such that $\{j \mid x_j \in P\} \subseteq \bigcap_{v \in T} ON(v) \cap I$, and let $N$ be a set of negative literals such that $\{j \mid \bar{x}_j \in N\} \subseteq \bigcap_{w \in F} ON(w) \cap I$. Notice that we use the convention that an intersection $\bigcap_{S \in \mathscr{S}} S$, where $\mathscr{S}$ is a collection of subsets $S \subseteq B$ of a base set $B$, yields $B$, i.e., the unit with respect to intersection, if $\mathscr{S}$ is empty. Thus, in particular $\bigcap_{v \in T} ON(v) = \{1, 2, ..., n\}$ if $T = \emptyset$, and $\bigcap_{w \in F} OFF(w) = \{1, 2, ..., n\}$ if $F = \emptyset$. The base set $B$ is always clear from the context.

Then, property REST$_\wedge$ (resp., REST$_\vee$) is defined as follows.

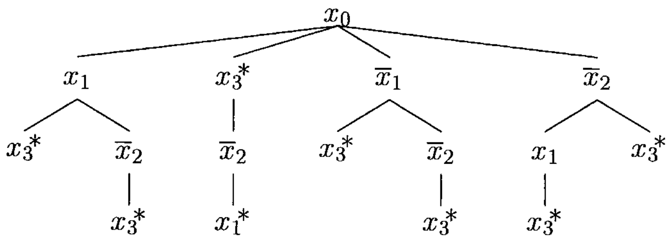



**FIG. 2.** The tree representing all extensions of $(T, F)$, which is constructed by ALL-DH-EXTENSIONS.

*Property* $REST_{\wedge}(I, P)$: The pdBf $(T[I], F[I])$ has an extension $f \neq \bot$. Moreover, if $n_1 > 0$ holds in the decomposition of (3.3), then the starting (outermost) level of the decomposition (i.e., literals $x_{11}, x_{12}, ..., x_{1n_1}$) must have only literals from $P$.

*Property* $REST_{\vee}(I, N)$: The pdBf $(T[I], F[I])$ has an extension $f \neq \top$. Moreover, if $n_1 = 0$ holds in the decomposition of (3.3), then the starting (outermost) level of the decomposition (i.e., literals $\bar{x}_{21}, \bar{x}_{22}, ..., \bar{x}_{2n_2}$) must have only literals from $N$.

For example, suppose that $T = \{110, 111\}$, $F = \{101, 001\}$, $I = \{1, 2, 3\}$, and $P = \{x_2\}$ are given. Then extensions $f = \bar{x}_3 \vee x_2$ and $g = x_2$ have property $\mathrm{REST}_{\wedge}(I, P)$, but $h = x_1 x_2$ does not. Let $I_P$ and $I_N$, respectively, denote $\{j \mid x_j \in P\}$ and $\{j \mid \bar{x}_j \in N\}$.

The properties $\mathrm{REST}_{\wedge}$ and $\mathrm{REST}_{\vee}$ can be checked by the following symmetric algorithms.

Algorithm $\mathrm{REST}_{\wedge}$-DH-EXTENSION($(T, F), I, P$)

*Input.* A pdBf $(T[I], F[I])$, given by $T, F \subseteq \{0, 1\}^n$ and $I \subseteq \{1, 2, ..., n\}$, and a set of positive literals $P$ such that $I_P = \{j \mid x_j \in P\} \subseteq \bigcap_{v \in T} ON(v) \cap I$.

*Return.* "Yes," if $(T[I], F[I])$ has a double Horn extension having property $\mathrm{REST}_{\wedge}(I, P)$; otherwise, "No."

**Step 1.** **if** $(T[I], F[I])$ has no double Horn extension **then** return "No" and exit **fi**;
(* this can be checked by DH-EXTENSION *)

**Step 2.** **for** each $S \subseteq I_P$ such that $|S| \geqslant |I_P| - 1$ **do**
$F_S := \{w \in F \mid ON(w) \supseteq S\}$;
**if** $F_S = \emptyset$ or $\bigcap_{w \in F_S} ON(w) \cap (I \backslash S) \neq \emptyset$
**then** return "Yes" and exit **fi**
**end**{for};

**Step 3.** return "No" and exit.

Algorithm $\mathrm{REST}_{\vee}$-DH-EXTENSION($(T, F), I, N$)

*Input.* A pdBf $(T[I], F[I])$, given by $T, F \subseteq \{0, 1\}^n$, $I \subseteq \{1, 2, ..., n\}$, and a set of negative literals $N$ such that $I_N = \{j \mid \bar{x}_j \in N\} \subseteq \bigcap_{w \in F} ON(w) \cap I$.

*Return.* "Yes," if $(T[I], F[I])$ has a double Horn extension having property $\mathrm{REST}_{\vee}(I, N)$; otherwise, "No."

**Step 1.** **if** $(T[I], F[I])$ has no double Horn extension **then** return "No" and exit **fi**;

**Step 2.** **for** each $S \subseteq I_N$ such that $|S| \geqslant |I_N| - 1$ **do**
$T_S := \{v \in T \mid ON(v) \supseteq S\}$;
**if** $T_S = \emptyset$ or $\bigcap_{v \in T_S} ON(v) \cap (I \backslash S) \neq \emptyset$
**then** return "Yes" and exit **fi**
**end**{for};

**Step 3.** return "No" and exit.

LEMMA 4.1. *The algorithms* REST$_\wedge$-DH-EXTENSION *and* REST$_\vee$-DH-EXTENSION *correctly output the answer in* $O(n(|T|+|F|))$ *time.*

*Proof.* We prove only the correctness of REST$_\wedge$-DH-EXTENSION, since the other case is similar. It is easy to see that Step 1 is correct. For the correctness of Step 2, assume first that $(T[I], F[I])$ has a double Horn extension and that the algorithm outputs "Yes." Then, there is some $S \subseteq I_P$ such that $|S| \geqslant |I_P|-1$ and one of the following two cases applies.

*Case* (a). $F_S = \emptyset$. Clearly $\varphi = \bigwedge_{i \in S} x_i$ represents a double Horn extension of $(T[I], F[I])$ and satisfies REST$_\wedge(I, P)$.

*Case* (b). $F_S \neq \emptyset$ and $I' = \bigcap_{w \in F_S} ON(w) \cap (I \backslash S) \neq \emptyset$. Then, it holds that every $w \in F_S$ fulfills $ON(w) \supseteq I_P$. This is obvious for $|S| = |I_P|$; if $|S| = |I_P|-1$ (in this case, we assume that $I_P$ satisfies $F_S \neq \emptyset$ and $\bigcap_{w \in F_{I_P}} ON(w) \cap (I \backslash I_P) = \emptyset$), then $S = I_P \backslash \{j\}$ for some $j \in I_P$ and hence $I' = \{j\}$, which means $ON(w) \supseteq I_P$.

Consider now two cases for $T$. If $T = \emptyset$, then $\varphi = \bigwedge_{i \in S} x_i\ (\bigvee_{i \in I'} \bar{x}_i)$ represents a double Horn extension of $(T[I], F[I])$ and satisfies REST$_\wedge(I, P)$. Otherwise (i.e., $T \neq \emptyset$), because all $w \in F_S$ satisfy $ON(w) \supseteq I_P$, also all $v \in T$ satisfy $ON(v) \supseteq I_P$, and since $(T[I], F[I])$ has a double Horn extension, $(T, F_S)$ has a double Horn extension $f$ projected to $I \backslash S$, and hence so does $(F_S, T)$ (since $\bar{f}$ is also double Horn). Thus by executing procedure DH-AUX$(F_S, T, I \backslash S)$ of Subsection 4.2, we have a read-once formula $\psi$ representing a double Horn extension of $(F_S, T)$ projected to $I \backslash I_S$. Furthermore, since $T \neq \emptyset$ and $I' \neq \emptyset$, the formula $\psi$ output by DH-AUX satisfies that the literals of its first decomposition are positive. Let $\psi'$ be a read-once formula equivalent to $\bar{\psi}$. Now we can easily see that $\varphi = \bigwedge_{i \in S} x_i \wedge \psi'$ represents a double Horn extension of $(T[I], F[I])$ and that the first positive decomposition of $\varphi$ uses literals in $P \backslash \{x_j\}$. This means that $\varphi$ satisfies REST$_\wedge(I, P)$. Hence Step 2 is correct.

Now assume that algorithm REST$_\wedge$-DH-EXTENSION outputs "No" in Step 3, but there is a double Horn extension $f$ of $(T[I], F[I])$, satisfying REST$_\wedge(I, P)$. Let $\psi$ be a read-once formula for $f$. Then $\psi' = \psi_{(x_i \leftarrow 1 \mid i \in I_P)}$ represents a double Horn extension of $(T, F_S)$, where $S = I_P$, projected to $I \backslash I_P$. Then the following two cases occur.

*Case* (c). $\psi' = \bot$. Obviously $\psi \neq \top$, and furthermore, $\psi \neq \bot$ holds since $T \neq \emptyset$ must hold (alternatively, we can conclude $\psi \neq \bot$ from property REST$_\wedge(I, P)$). Let us now consider the first positive literals (i.e., literals $x_{11}, x_{12}, ..., x_{1n_1}$) and first negative literals (i.e., literals $\bar{x}_{21}, \bar{x}_{22}, ..., \bar{x}_{2n_2}$) in the decomposition of (3.3) for $\psi$. By REST$_\wedge(I, P)$, the first positive literals are contained in $P$. We can see that $n_2 \neq 0$ and that the first negative literals $\bar{x}_l$ are also contained in $P$, since $n_2 = 0$ and $\bar{x}_l \notin P$ would imply $\psi' = \top$ and $\psi' = \bar{x}_l \vee ...$ or $\top$, respectively. Thus $\psi$ has an implicant $t = (\bigwedge_{p=1}^{l} x_{i_p})\bar{x}_{i_{l+1}}$, where $i_p \in I_P$ for $p = 1, 2..., l+1$. However, since Step 2 is executed, for $S = I_P \backslash \{i_{l+1}\}$ there exists a $w \in F$ such that $ON(w) \cap I_P = S$. Such a $w$ satisfies $t(w) = 1$, and hence $\psi(w) = 1$, which is a contradiction.

*Case* (d). $\psi' \neq \bot$. Obviously $\psi' \neq \top$ holds as for $S = I_P$, $F_S \neq \emptyset$. Furthermore, the first literal in the read-once formula $\psi'$ is positive. Otherwise (i.e., $\psi' = \bar{x}_i \vee \varphi$), since by assumption for $S = I_P$ it holds that $\bigcap_{w \in F_S} ON(w) \cap (I \backslash S) = \emptyset$, there is a $w \in F_S$ such that $\psi'(w) = 1$, which is a contradiction. Therefore, let $\psi' = x_i \varphi$, where $i \notin I_P$. We claim that $x_i$ is used in the first group of positive literals in the decomposition of (3.3) for $\psi$, i.e., in $x_{11}, x_{12}, ..., x_{1n_1}$, which derives a contradiction to property $\text{REST}_\wedge(I, P)$. Suppose this were not the case. Then, $\psi$ has $d \geqslant 3$ and $n_2 \geqslant 1$, and since $\psi' = x_i \varphi$, every literal $\bar{x}_{2k}$ from the first negative group must use a variable $x_j$ such that $j \in I_P$, and at least one such $j$ exists. Thus $\psi$ has an implicant $t = (\bigwedge_{p=1}^{l} x_{i_p})\, \bar{x}_j$, where $i_p \in I_P$ for $p = 1, 2..., l$ and $j \in I_P$. However, there is $w \in F$ such that $ON(w) \cap I_P = I_P \backslash \{j\}$. Such a $w$ satisfies $t(w) = 1$, and hence $\psi(w) = 1$, which is a contradiction. This proves the correctness of $\text{REST}_\wedge$-DH-EXTENSION.

Finally, we consider the time complexity of the algorithms. By applying a faster version of algorithm DH-EXTENSION, Step 1 of $\text{REST}_\wedge$-DH-EXTENSION (resp., $\text{REST}_\vee$-DH-EXTENSION) can be executed in $O(|I|(|T| + |F|)) = O(n(|T| + |F|))$ time. Step 2 can be done in $O(|P|\,|F|) = O(n\,|F|)$ (resp., $O(|N|\,|T|) = O(n\,|T|)$) time. (To see this, let $F' = \{w \in F \mid |ON(w) \cap I_P| = |I_P| - 1\}$; then, for $S = I_P$, we have $F_S = \{w \in F \mid |ON(w) \cap I_P| = |I_P|\}$, and for every $S_j = I_P \backslash \{j\}$ such that $j \in I_P$, $F_{S_j} = F_{I_P} \cup \{w \in F' \mid j \in OFF(w)\}$.) Finally, Step 3 requires constant time.

Totally, algorithm $\text{REST}_\wedge$-DH-EXTENSION and $\text{REST}_\vee$-DH-EXTENSION require $O(n(|T| + |F|))$ time. ▮

The main algorithm ALL-DH-EXTENSIONS for finding all double Horn extensions is now described in detail below. It initializes the variables and takes care of the special case of extension $f = \bot$. Then, in the case that a double Horn extension exists, it recursively calls an auxiliary procedure, ALL-DH-AUX, for outputting all double Horn extensions subject to a restriction on the prefix of the extensions. The above $\text{REST}_\wedge$-DH-EXTENSION and $\text{REST}_\vee$-DH-EXTENSION are used as subprocedures in ALL-DH-AUX.

ALGORITHM ALL-DH-EXTENSIONS

*Input.* A pdBf $(T, F)$, where $T, F \subseteq \{0, 1\}^n$.

*Output.* List representations of read-once formulas $\psi_1, \psi_2, ..., \psi_m$ of (3.3) for all double Horn extensions of $(T, F)$.

**Step 1.** **if** $T = \emptyset$ **then** output $\bot$ (continue)
  **else** call DH-EXTENSION for $(T, F)$;
    **if** its answer is "No" **then** halt **fi**
  **fi**;
  $L_0 := x_0$; $I := \{1, 2, ..., n\}$;
  $P = \{x_j \mid j \in \bigcap_{v \in T} ON(v)\}$;
  $N = \{\bar{x}_j \mid j \in \bigcap_{w \in F} ON(w)\}$;
  Call ALL-DH-AUX($(T, F)$, $L_0$, $I$, $P$, $N$).

*Procedure* ALL-DH-AUX($(T, F)$, $L$, $I$, $P$, $N$)

*Input.* A pdBf $(T, F)$, formula $\gamma$ of form (3.3) given by list $L = L_0, L_1, ..., L_i$ of literals (in this order), set $I$ of the available variable indices and sets of literals $P$ and $N$ allowed for decomposition (where $I_P\,(= \{j \mid x_j \in P\}) \subseteq I$ and $I_N\,(= \{j \mid \bar{x}_j \in N\}) \subseteq I$ are assumed).

*Output.* List representations of read-once formulas $\gamma \vee \psi$ or $\gamma \wedge \psi$, where $\psi$ are the read-once formulas of (3.3) 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 $\gamma$ as prefix) such that the first literal $L_{i+1}$ after $L$ is chosen from $P \cup N$.

**Step 1.** (* Check if current formula $\gamma$ is an extension, and output $\gamma$ if it is the case. *)
**if** $L_i$ is positive and $F = \varnothing$, or $L_i$ is negative and $T = \varnothing$
**then** output $\gamma$ (continue) **fi**;

**Step 2.** (* Expand $\gamma$ by a positive literal. *)
**while** there is a literal $x_j \in P$ **do**
  $I' := I \backslash \{j\}$; $\underline{P' := P := P \backslash \{x_j\}}$;
  $T' := T$ and $F' := \{v \in F \mid v_j = 1\}$;
  Call REST$_\wedge$-DH-EXTENSION($(T', F'), I', P'$);
  **if** the answer is "Yes"
  **then** (* expand $\gamma$ by $x_j$ *)
    $L_{i+1} := x_j$ and $L' := L, L_{i+1}$;
    $N' := \{\bar{x}_i \mid i \in I' \cap \bigcap_{w \in F'} ON(w)\}$;
    Call recursively ALL-DH-AUX($(T', F'), L', I', P', N'$)
  **fi**
**end**{while}.

**Step 3.** (* Expand $\gamma$ by a negative literal. *)
**while** there is a literal $\bar{x}_j \in N$ **do**
  $I' := I \backslash \{j\}$; $\underline{N' := N := N \backslash \{\bar{x}_j\}}$;
  $T' := \{v \in T | v_j = 1\}$ and $F' := F$;
  Call REST$_\vee$-DH-EXTENSION($(T', F'), I', N'$);
  **if** the answer is "Yes"
  **then** (* expand $\gamma$ by $\bar{x}_j$ *)
    $L_{i+1} := \bar{x}_j$ and and $L' := L, L_{i+1}$;
    $P' := \{x_i | i \in I' \cap \bigcap_{v \in T'} ON(v)\}$;
    Call recursively ALL-DH-AUX($(T', F'), L', I', P', N'$)
  **fi**
**end**{while}.

Step 1 of ALL-DH-AUX outputs the extension currently at hand. In particular, if $L = x_0$, we have by convention $\gamma = \top$. Steps 2 and 3 try to expand the current formula given by list $L$, in all possible ways. By recursively restricting the possibilities, it is achieved that each extension is output only once. The crux in the correctness of the algorithm relies on the double assignments $P' := P \backslash \{x_j\}$ and $N' := N \backslash \{\bar{x}_j\}$ in Steps 2 and 3, respectively.

The computation of all double Horn extensions is illustrated in the following example.

EXAMPLE 4.3. We apply ALL-DH-EXTENSIONS to the pdBf $(T, F)$, where $T = \{(111), (101)\}$ and $F = \{(110)\}$ as in Example 4.2. The entire computational process is represented by the tree in Fig. 2. It outputs the eight formulas $\gamma^{(i)}$, $i=1, 2, ..., 8$, of Example 4.2, which represent all distinct double Horn extensions of $(T, F)$.

*Step* 1. DH-EXTENSION for $(T, F)$ answers "Yes" (e.g., $f=x_3$ is a double Horn extension), and thus ALL-DH-AUX is called with $(T, F)$, $L=x_0$, $I=\{1, 2, 3\}$, $P=\{x_1, x_3\}$ and $N=\{\bar{x}_1, \bar{x}_2\}$;

(*All-DH-AUX* (1)) *Step* 1. No output, as $L_i=x_0$ (which represents $\gamma=\top$) is positive and $F\neq\emptyset$.

*Step* 2. Expand $\gamma=\top$ by $x_1$: $T':=\{(111), (101)\}$, $F':=\{(110)\}$, $I':=\{2, 3\}$, $P':=\{x_3\}$; The call of REST$_\wedge$-DH-EXTENSION$((T', F'), I', P')$ answers "Yes." (Indeed, $(T'[I'], F'[I'])=(\{(11), (01)\}, \{(10)\})$ has a double Horn extension (Step 1), and in Step 3, $F^+=\{(110)\}$ and $ON((110))\cap(\{2, 3\}\backslash\{3\})\neq\emptyset$.) Thus, ALL-DH-AUX is called with $(T', F')$, $L'=x_0, x_1$, $I'=\{2, 3\}$, $P'=\{x_3\}$, and $N'=\{\bar{x}_2\}$.

(*ALL-DH-AUX* (2)) *Step* 1. No output, as $L_i=x_1$ is positive and $F\neq\emptyset$.

*Step* 2. Expand $\gamma=x_1$ by $x_3$: $T':=\{(111), (101)\}$, $F':=\emptyset$, $I':=\{2\}$; $P':=\emptyset$; The call of REST$_\wedge$-DH-EXTENSION$((T', F'), I', P')$ answers "Yes"; hence, ALL-DH-AUX is called with $(T', F')$, $L'=x_0, x_1, x_3$, $I'=\{2\}$, $P'$, and $N'=\emptyset$. The first step of this call outputs $\gamma$ (as $L_i=x_3$ is positive and $F=\emptyset$), i.e., "$x_1x_3$"; the second and third steps are void.

*Step* 3. Expand $\gamma=x_1$ by $\bar{x}_2$: $T':=\{(111)\}$; $F':=\{(110)\}$; $I':=\{3\}$; $N':=\emptyset$ (we have $P'=\emptyset$ as a result of Step 2). The call of REST$_\vee$-DH-EXTENSION$((T', F'), I', N')$ answers "Yes"; thus, ALL-DH-AUX is called with $(T', F')$, $L'=x_0, x_1, \bar{x}_2$, $I'=\{3\}$, $P'=\{x_3\}$, and $N'=\emptyset$.

(*ALL-DH-AUX* (3)) *Step* 1. No output, as $L_i=\bar{x}_2$ is negative and $T\neq\emptyset$.

*Step* 2. Expand $\gamma=\bar{x}_2$ by $x_3$: $T':=\{(111)\}$, $F':=\emptyset$; $I':=\emptyset$; $P':=\emptyset$ (we have $N'=\emptyset$). The call of REST$_\wedge$-DH-EXTENSION$((T', F'), I', P')$ answers "Yes." The subsequent call of ALL-DH-AUX with $(T', F')$, $L'=x_0, x_1, \bar{x}_2, x_3$, $I'=\emptyset$, $P'=\emptyset$, and $N'=\emptyset$ outputs $\gamma$ in the first step, i.e., "$x_1(\bar{x}_2\vee x_3)$"; the second and third steps are void.

*Step* 3. Void.
(*end of ALL-DH-AUX* (3))

(*end of ALL-DH-AUX* (2))

(*Step* 2. *continued*) Expand $\gamma=\top$ by $x_3$: ...

*Step* 3. Expand $\gamma=\top$ by $\bar{x}_1$: ...
Expand $\gamma=\top$ by $\bar{x}_2$: ...
(*end of ALL-DH-AUX* (1))

(*end of ALL-DH-EXTENSIONS*)

THEOREM 4.5. *Algorithm* ALL-DH-EXTENSIONS *correctly outputs list representations of read-once formulas* $\varphi_1, \varphi_2, ..., \varphi_m$ *for all double Horn extensions of* $(T, F)$, *with* $O(n^3(|T|+|F|))$ *delay, i.e., polynomial delay, where* $\varphi_i\not\equiv\varphi_j$ *for* $i\neq j$.

*Proof.* It can be shown by induction on $|I|$ that all formulas output indeed represent double Horn extensions of $(T, F)$, similarly to the correctness of DH-EXTENSION (Theorem 4.2). The additional part required here is the argument that all extensions are actually found, and no one is output more than once.

For the former, we observe that for every double Horn extension $\psi$ of $(T, F)$, there is a "computation path" in the procedure which outputs $\psi$. Indeed, if $\psi$ has decomposition (3.3) where $n_1 \geqslant 1$, then some $x_{1j}$ will be picked by ALL-DH-AUX in Step 2, and recursively the formula $\psi_{(x_{1j} \leftarrow 1)}$ is constructed and $\psi = x_{1j} \wedge \psi_{(x_{1j} \leftarrow 1)}$ is output. Similarly, if $n_1 = 0$, then some $\bar{x}_{2j}$ is picked in Step 3 and recursively the extension for $\psi_{(x_{2j} \leftarrow 1)}$ is constructed and $\psi = \bar{x}_{2j} \vee \psi_{(x_{2j} \leftarrow 1)}$ is output. A rigorous proof by induction is straightforward.

Let us next assume that ALL-DH-EXTENSIONS outputs two lists $L = L_0$, $L_1, ..., L_m$ and $L' = L'_0, L'_1, ..., L'_m$, where $L_i = L'_i$ for $i \leqslant k$ but $L_{k+1} \neq L'_{k+1}$, and $L$ and $L'$ represent the same double Horn function $f$. For example, $L = x_0, x_1, x_2, \bar{x}_3$, $\bar{x}_4$ and $L = x_0, x_1, x_2, \bar{x}_4, \bar{x}_3$ with $k = 2$ represent the same function $x_1 x_2 (\bar{x}_3 \vee \bar{x}_4)$. Note first that $L_{k+1}$ and $L'_{k+1}$ must have the same polarity (i.e., both are positive or negative), since otherwise $L$ and $L'$ represent different functions. Now consider the execution of ALL-DH-AUX for the common sublist $L^{(k)} = L_0, L_1, ..., L_k$ $(= L'_0, L'_1, ..., L'_k)$ of $L$ and $L'$. Let us assume without loss of generality that $L_{k+1}$ and $L'_{k+1}$ are both positive and that $L_{k+1}$ is checked before $L'_{k+1}$ in expanding $L^{(k)}$ in Step 2. In expanding $L^{(k)}$ by $L'_{k+1}$, ALL-DH-AUX is recursively called with $L^{(k+1)}$ $(= L_0, L_1, ..., L_k, L'_{k+1})$, $I'$, $P'$ and $N'$. Note that this initial $P'$ does not contain $L_k$. Furthermore, in each subsequent recursion call, the generated $P'$ does not contain $L_k$ unless some prior recursion step has already expanded the formula by a negative literal. This implies that the expansion of $L^{(k)}$ by $L'_{k+1}$ never produces $L'$, which represents the same function as $L$. Consequently, no two list representations $L$ and $L'$, representing the same function, are output.

Concerning the complexity, Step 1 of algorithm ALL-DH-EXTENSIONS can be executed in $O(n(|T| + |F|))$ time by using a faster version of DH-EXTENSION. For procedure ALL-DH-AUX, Step 1 requires constant time. In Step 2 (resp., Step 3), for each $x_j \in P$ (resp., $\bar{x}_j \in N$), the body of the while-loop apart from the recursive call of ALL-DH-AUX can be clearly done in $O(n(|T| + |F|))$ time, as REST$_\wedge$-DH-EXTENSION (resp., REST$_\vee$-DH-EXTENSION) runs in $O(n(|T| + |F|))$ time by Lemma 4.1. Between two consecutive outputs, at most $n^2$ checks by REST$_\wedge$-DH-EXTENSION (resp., REST$_\vee$-DH-EXTENSION) may fail in the worst case, since, in each recursion step, at most $n$ checks may be required, and the recursion depth is bounded by $n$. Therefore, the algorithm outputs extensions with $O(n^3(|T| + |F|))$ delay. ▮

COROLLARY 4.1. *There is a polynomial delay algorithm for enumerating prime DNFs for all double Horn extensions of a given pdBf $(T, F)$.*

*Proof.* By Corollary 3.2, prime DNF for a read-once formula $\varphi$ of (3.3) can be obtained from $\varphi$ in $O(n^2)$ time. ▮

Clearly, the double Horn extensions of the pdBf $(T, F)$, where $T = F = \varnothing$, are all double Horn functions. Thus, an immediate consequence of the previous corollary is the following one.

COROLLARY 4.2. *There is a polynomial delay algorithm for enumerating prime DNFs of all functions in the class $\mathscr{C}_{DH}(n)$, i.e., the class of all double Horn functions of n variables.*

COROLLARY 4.3. *Given a pdBf $(T, F)$, deciding whether it has the unique double Horn extension is possible in polynomial time.*

The last result has an immediate application in the context of concept learning as described in Section 1. It means that, given a sample for a double Horn function $f$, i.e., sets $T \subseteq T(f)$ and $F \subseteq F(f)$, respectively, it is possible to decide in polynomial time whether the examples uniquely determine $f$, i.e., allow for only one double Horn function compatible with the examples. Moreover, a DNF expression for $f$ can be output in polynomial time in this case.

Thus, we obtain that exact learning of a double Horn function $f$ in a batch-mode (cf. [19]), in which the algorithm is given a sample and then has to extract the function $f$ from it without further information, is feasible in polynomial time, i.e., either the function $f$ is output or the algorithm reports that the sample is ambiguous.

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 the 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 extension $x_1 x_2 \ldots x_n$. Although this is true even if arbitrary vectors except $\mathbf{1}$ are added to $F$, checking the fact appears to be more difficult.

## 5. CHARACTERISTIC SETS

### 5.1. Characteristic Sets of Double Horn Functions

A vector $x \in X \subseteq \{0, 1\}^n$ is called *extreme* [7] with respect to a set $X$ if $x \notin Cl_\wedge(X \setminus \{x\})$. The set of all extremal vectors of $X$ is called the *characteristic set* of $X$ [24, 25, 27] (or its *base* [7]), and is denoted by $C^*(X)$. Note that every set $X \subseteq \{0, 1\}^n$ has the unique characteristic set $C^*(X)$, and that $C^*(X) \subseteq X$ is the minimum set satisfying $Cl_\wedge(C^*(X)) = Cl_\wedge(X)$. For convenience, we use the notation $X[j] = \{v \in X \mid |OFF(v)| = j\}$ throughout this section for any set of vectors $X$ and $j = 0, 1, \ldots, n$.

Let us first give the following lemma, which was proved in [9].

LEMMA 5.1. *Let $T, F \subseteq \{0, 1\}^n$. Then $Cl_\wedge(T) \cup Cl_\wedge(F) = \{0, 1\}^n$ holds if and only if*

$$Cl_\wedge(T) \cup Cl_\wedge(F) \supseteq \{x^{V \setminus W} \mid |W| \leqslant 2, W \subseteq V\} \tag{5.7}$$

*holds, where $V = \{1, 2, \ldots, n\}$.*

*Proof.* The only-if-part is obvious. For the if-part, from the assumption that (5.7) holds and there is a vector

$$v \in \{0, 1\}^n \setminus (Cl_\wedge(T) \cup Cl_\wedge(F)), \tag{5.8}$$

we derive a contradiction. By (5.7), $v$ satisfies $|OFF(v)| \geqslant 3$. Let $X^T = \{w \in Cl_\wedge(T) \mid w \geqslant v\}$, $X^F = \{w \in Cl_\wedge(F) \mid w \geqslant v\}$, and denote $V^T = OFF(X^T[1])$ and $V^F = OFF(X^F[1])$, where $OFF(X) = \bigcup_{w \in X} OFF(w)$.

Then by (5.7), we have

$$V^T \cup V^F = OFF(v). \tag{5.9}$$

Furthermore, $V^T \setminus V^F \neq \emptyset$ and $V^F \setminus V^T \neq \emptyset$ hold. In fact, if $V^T \setminus V^F = \emptyset$ (resp., $V^T \setminus V^F = \emptyset$), then (5.9) implies $V^F = OFF(v)$ (resp., $V^T = OFF(v)$); this means $v = \bigwedge_{u \in X^F[1]} u$ (resp., $v = \bigwedge_{u \in X^T[1]} u$), which is a contradiction to (5.8).

For every $i \in V^T \setminus V^F$, define $Q_i = \{x^{V \setminus \{i, j\}} \mid j \in V_1^F \setminus V_1^T\}$. Then, for each $i \in V^T \setminus V^F$, some vector $w^{(i)} \in Q_i$ satisfies $w^{(i)} \in X^F[2]$, since otherwise, (5.7) implies that all vectors $w \in Q_i$ are in $X^T[2]$, and hence $v = \bigwedge_{u \in X^T[1]} u \wedge \bigwedge_{u \in X^T[2]} u$ holds, which is a contradiction to (5.8). Now it is easy to see that by choice of the $w^{(i)}$ and (5.9),

$$v = \bigwedge_{u \in X^F[1]} u \wedge \bigwedge_{i \in V^T \setminus V^F} w^{(i)}$$

holds; this is again a contradiction to (5.8). ∎

By applying the above lemma, we can characterize the characteristic sets of double Horn functions through graph-theoretical properties.

THEOREM 5.1. *Let $T^*, F^* \subseteq \{0, 1\}^n$ and $V = \{1, 2, ..., n\}$, and denote $V_{T^*} = \{j \mid x^{V \setminus \{j\}} \in T^*\}$, $V_{F^*} = \{j \mid x^{V \setminus \{j\}} \in F^*\}$, $T^*[2] = \{v \in T^* \mid |OFF(v)| = 2\}$ and $F^*[2] = \{w \in F^* \mid |OFF(w)| = 2\}$. Then $T^* = C^*(T(f))$ and $F^* = C^*(F(f))$ hold for some double Horn function $f$ if and only if the following conditions hold.*

(i) *$T^* \cap F^* = \emptyset$ and $T^* \cup F^* \subseteq \{x^{V \setminus W} \mid |W| \leqslant 2\}$.*

(ii) $\{x^{V \setminus W} \mid |W| \leqslant 2\} \setminus (T^* \cup F^*) = \{x^{V \setminus W} \mid |W| = 2, \quad W \subseteq V_{T^*}\} \cup \{x^{V \setminus W} \mid |W| = 2, W \subseteq V_{F^*}\}$.

(iii) *Let $T^*[2]$ and $F^*[2]$ define an orientation to the complete bipartite graph $G = (V_{T^*}, V_{F^*}, E = V_{T^*} \times V_{F^*})$ by directing the edges $\{i, j\}$ corresponding to $x^{V \setminus \{i, j\}} \in T^*[2]$ from $V_{F^*}$ to $V_{T^*}$ and the edges $\{i, j\}$ corresponding to $x^{V \setminus \{i, j\}} \in F^*[2]$ from $V_{T^*}$ to $V_{F^*}$. Then $G$ has no directed cycle.*

*Proof.* Let us first show the only-if-part. (i) and (ii) are immediate from Definition 3 and Lemma 5.1. For (iii), note that if $V_{T^*} = \emptyset$ or $V_{F^*} = \emptyset$, then clearly (iii) holds. Hence we assume $V_{T^*} \neq \emptyset$ and $V_{F^*} \neq \emptyset$. Then by (i) and (ii), $T^*[2]$ and $F^*[2]$ give orientation to all edges in $G$. Assume that the oriented graph $G^+$ contains a directed cycle $C$, and we will derive a contradiction. Denote by $V_C$ the set of vertices in $C$, and by $C_{T^*}$ (resp., $C_{F^*}$) the set of arcs $\{i, j\}$ in $C$ such that

$x^{V\setminus\{i,j\}} \in T^*[2]$ (resp., $x^{V\setminus\{i,j\}} \in F^*[2]$). From the definition of $G^+$, it is immediate that every vertex $j \in V_C$ occurs both in $C_{T^*}$ and $C_{F^*}$. This means that

$$\begin{aligned} x^{V\setminus V_C} &= \bigwedge_{\{i,j\} \in C_{T^*}} x^{V\setminus\{i,j\}} \ (\in Cl_\wedge(T^*)) \\ &= \bigwedge_{\{i,j\} \in C_{F^*}} x^{V\setminus\{i,j\}} \ (\in Cl_\wedge(F^*). \end{aligned}$$

Consequently, $x^{V\setminus V_C} \in Cl_\wedge(T^*) \cap Cl_\wedge(F^*)$ and hence no double Horn function $f$ exists such that $T^* = C^*(T(f))$ and $F^* = C^*(F(f))$.

To prove the if-part, suppose that (i), (ii) and (iii) hold. Then it follows by Lemma 5.1 that $Cl_\wedge(T^*) \cup Cl_\wedge(F^*) = \{0,1\}^n$. To show that $Cl_\wedge(T^*) \cap Cl_\wedge(F^*) = \emptyset$, we assume that $x^{V\setminus W} \in Cl_\wedge(T^*) \cap Cl_\wedge(F^*)$ and derive a contradiction. Denote $W_{T^*} = W \cap V_{T^*}$ and $W_{F^*} = W \cap V_{F^*}$. Notice that $W_{T^*}, W_{F^*} \neq \emptyset$ hold by (i) and $|W| \geqslant 3$ holds by (i) and (ii). Let $G^+_W$ denote the directed subgraph of $G^+$ induced by $W_{T^*} \cup W_{F^*}$. Since $x^{V\setminus W} \in Cl_\wedge(T^*)$, for every $i \in W_{F^*}$, there is a $j \in W_{T^*}$ such that $G^+$ contains an arc $\{i,j\}$, which corresponds to $x^{V\setminus\{i,j\}} \in T^*[2]$. Hence the outdegree of $i \in W_{F^*}$ is at least one in $G^+_W$. Similarly, since $x^{V\setminus W} \in Cl_\wedge(F^*)$, the outdegree of $i \in W_{T^*}$ is at least one in $G^+_W$. Therefore, $G^+_W$ contains a directed cycle, which is a contradiction. ∎

COROLLARY 5.1. *If $f$ is double Horn, then*

$$C^*(T(f)) = T(f)[0] \cup T(f)[1] \cup (T(f)[2] \setminus Cl_\wedge(T(f)[1]))$$

*and*

$$C^*(F(f)) = F(f)[0] \cup F(f)[1] \cup (F(f)[2] \setminus Cl_\wedge(F(f)[1])).$$

We comment here that, by this corollary, we can check whether a given pdBf $(T, F)$ satisfies $T = C^*(T(f))$ and $F = C^*(F(f))$ for some double Horn function $f$ in linear time.

### 5.2. *Transformations between Characteristic Sets and DNFs*

In this subsection, we study the transformation problems for a double Horn function $f$ between various representations such as $C^*(T(f))$, $C^*(F(f))$ and DNFs for $f$, $f^d$, and $\bar{f}$.

LEMMA 5.2. *Let $f$ be a double Horn function of $n$ varaibles. Then $C^*(F(f))$ can be computed from $C^*(T(f))$, and vice versa, in $O(n^3)$ time.*

*Proof.* Given $T^* = C^*(T(f)) \subseteq \{x^{V\setminus W} \mid |W| \leqslant 2\}$, compute the set $F^*$ that satisfies conditions (i), (ii), and (iii) of Theorem 5.1, and let this $F^*$ be $C^*(F(f))$. In fact, $F^*$ contains $\mathbf{1}$ if $\mathbf{1} \notin T^*$, all vectors $x^{V\setminus\{i\}}$ where $i \in V_{F^*} = V \setminus V_{T^*}$ and $V_{T^*} = \{j \mid x^{V\setminus\{j\}} \in T^*\}$, and every vector $x^{V\setminus\{i,j\}}$ such that $i \in V_{T^*}$, $j \in V_{F^*}$, and $x^{V\setminus\{i,j\}} \notin T^*$. To determine those $x^{V\setminus\{i,j\}}$ fast, build for all $i \in V_{T^*}$ sorted lists $L_i$ containing all $j \in V_{F^*}$ such that $x^{V\setminus\{i,j\}} \in T^*$. Then, for each $i \in V_{T^*}$ and $j \in V_{F^*}$, select vector $x^{V\setminus\{i,j\}}$ for $F^*$ if $j$ does not occur in $L_i$. Clearly, construction of all $L_i$

and selection of all $x^{V\setminus\{i,j\}}$ can be done in $O(n|T^*|+n|V_{T^*}|\,|V_{F^*}|)=O(n^3)$ time. Hence, the overall time to conduct the computation of $F^*$ is $O(n^3)$; note that $C^*(T(f))$ and $C^*(F(f))$ may have $\Theta(n^2)$ vectors. ▮

The computation of the characteristic set of a general Horn function has been considered in [24–27] and in a different setting in [7]. In general, the computation may take exponential time, even from a Horn formula, since the characteristic set may be exponentially large. The result in Theorem 5.1 identifies a nontrivial subclass of Horn functions for which characteristic sets are small and, as will be shown by the next theorem, polynomially computable. Moreover, it will be shown in [9] that this property also extends to renamings of double Horn functions.

Characteristic sets appear to be useful for manipulating a double Horn function $f$ represented by a general formula $\varphi$. If a function $f$ is known to be double Horn (which cannot be checked in polynomial time), we will show below that its characteristic sets $C^*(T(f))$ and $C^*(F(f))$, and its prime DNF $\varphi$ can be computed in polynomial time. Note that this is not straightforward since reducing a DNF to a prime DNF, for example, is computationally expensive in general.

THEOREM 5.2. *Let $f$ be a double Horn function of $n$ variables.*

(i) *If $f$ is given by an arbitrary formula $\varphi$, then its characteristic sets $C^*(T(f))$ and $C^*(F(f))$ can be computed in $O(n^2|\varphi|)$ time.*

(ii) *If $f$ is given by its characteristic set $C^*(T(f))$ or $C^*(F(f))$, then the unique prime DNFs of $f$, $\bar{f}$, and $f^d$ can be computed in $O(n^3)$ time.*

(iii) *If $f$ is given by an arbitrary formula $\varphi$, then the unique prime DNFs of $f$, $\bar{f}$, and $f^d$ can be computed in $O(n^2\max(|\varphi|, n))$ time.*

*Proof.* (i) The following algorithm computes $C^*(T(f))$ and $C^*(F(f))$. Initialize $T^*=\emptyset$ and $F^*=\emptyset$. Then, for each vector $v$ such that $|OFF(v)|\leqslant 1$, test whether $\varphi(v)=1$; if so, then add $v$ to $T^*$, otherwise to $F^*$. If either $T^*[1]$ or $F^*[1]$ is empty, then output $T^*$ and $F^*$ and halt. Otherwise set $V_{T^*}=\bigcup_{v\in T^*}OFF(v)$ and $V_{F^*}=\bigcup_{w\in F^*}OFF(w)$. Then check for each vector $v$ with $OFF(v)=\{i,j\}$, where $i\in V_{T^*}$ and $j\in V_{F^*}$, whether $\varphi(v)=1$; if so, then add $v$ to $T^*$, otherwise add $v$ to $F^*$. Output $T^*$ and $F^*$.

The correctness of this algorithm follows from Theorem 5.1. Deciding whether $\varphi(v)=1$ for a given vector $v$ can be done, in time $O(|\varphi|)$ for any formula $\varphi$, and furthermore, this algorithm computes $\varphi(v)$ for at most $n^2$ vectors. Thus it takes $O(n^2|\varphi|)$ time.

(ii) First compute $C^*(F(f))$ (resp., $C^*(T(f))$) from $C^*(T(f))$ (resp., $C^*(F(f))$). By Lemma 5.2, this can be done in $O(n^3)$ time. Then compute an extension of pdBf $(T^*, F^*)$, where $T^*=C^*(T(f))$ and $F^*=C^*(F(f))$. As we have see in Subsection 4.1, it can be computed in $O(n^3)$ time. This extension is equal to $f$ in this case, and its read-once formula is output. As noted in Corollary 3.2, such a formula can be transformed to the unique prime DNF of $f$ in $O(n^2)$ time, which can then be transformed to the unique prime DNFs of $\bar{f}$ and $f^d$ in $O(n^2)$ time, respectively.

(iii) First compute the characteristic sets $T^* = C^*(T(f))$ and $F^* = C^*(F(f))$ in $O(n^2|\varphi|)$ time by applying the result of (i). Then, proceed to the procedure of (ii) to compute the prime DNFs in $O(n^3)$ time. In total, this computation takes $O(n^2|\varphi| + n^3) = O(n^2 \max(|\varphi|, n))$ time. ∎

In passing, we remark that the complexity of (iii) can be improved to $O(\max(|\varphi|, n^2))$ time if the original $\varphi$ is a Horn DNF.

COROLLARY 5.2. *Let $\varphi$ be a Horn DNF representing a double Horn function $f$ of $n$ variables. Let $\varphi_1$, $\varphi_2$, and $\varphi_3$, respectively, be the unique prime DNFs of $f$, $\bar{f}$, and $f^d$. Then $\varphi_1$ can be computed in $O(|\varphi|)$ time, and $\varphi_2$ and $\varphi_3$ can be computed in $O(\max(|\varphi|, n^2))$ time.*

*Proof.* First apply the faster algorithm for checking if a Horn DNF $\varphi$ represents a double Horn function, which was described before Theorem 3.3 in Subsection 3.3. This returns the read-once formula $\psi_1$ of (3.3) in $O(|\varphi|)$ time. Then, by Corollary 3.2 in Subsection 3.2, the prime DNFs of $f$, $\bar{f}$, and $f^d$ can be computed from $\varphi$, in $O(|\varphi_1|)$, $O(|\varphi_2|)$, and $O(|\varphi_3|)$ time, respectively. The total time required for computing $\varphi_i$ is $O(\max(|\varphi|, |\varphi_i|))$. Since $|\varphi_1| \leqslant |\varphi|$, $\varphi_1$ can be computed in $O(|\varphi|)$ time, and since $|\varphi_2|, |\varphi_3| \leqslant n^2$, $\varphi_2$ and $\varphi_3$ can be computed in $O(\max(|\varphi|, n^2))$ time. ∎

### *5.3. Counting Double Horn Functions*

Let $\mathscr{C}_{DH}(n)$ denote the class of double Horn functions of $n$ variables, and let $\hat{\mathscr{C}}_{DH}(n)$ denote the class of double Horn functions which depend on exactly $n$ variables. Then clearly $\mathscr{C}_{DH} = \bigcup_n \mathscr{C}_{DH}(n)$ and $\mathscr{C}_{DH}(n) = \bigcup_{m=0}^{n} \hat{\mathscr{C}}_{DH}(m)$. Furthermore, let $\#DH(n) = |\mathscr{C}_{DH}(n)|$ and $\#\widehat{DH}(n) = |\hat{\mathscr{C}}_{DH}(n)|$.

In this subsection, we give simple closed expressions for $\#DH(n)$ and $\#\widehat{DH}(n)$. Any double Horn function $f$ can be defined by $T = C^*(T(f))$ and $F = C^*(F(f))$, which satisfy the conditions in Theorem 5.1 (see Subsection 5.1). Hence we have the following observation for $\mathscr{C}_{DH}(n)$, where $n \geqslant 2$. For vector $\mathbf{1}$, there are two possibilities, either $\mathbf{1} \in T$ or $\mathbf{1} \in F$. Furthermore, assume that $m$ vectors $v$ with $|OFF(v)| = 1$ satisfy $f(v) = 1$ (i.e., $m = |V_T|$). If $m \neq 0, n$, then, since $T_2$ and $F_2$ define a complete acyclic digraph $G^+$, there are $CF_{m,n-m}$ possibilities, where $CF_{i,j}$ for $i, j > 0$ denotes the number of orientations of the complete bipartite graph $K_{i,j}$ which have no directed cycle. Adding the cases of $m = 0$ and $m = n$, we derive the next proposition.

PROPOSITION 5.1.

$$\#DH(n) = 2\left(\sum_{m=1}^{n-1} \binom{n}{m} CF_{m,n-m} + 2\right)$$

$$= 2 \sum_{m=0}^{n} \binom{n}{m} CF_{m,n-m}, \tag{5.10}$$

*where $CF_{0,j} = CF_{i,0} = 1$.*

*Proof.* The above observation tells that (5.10) holds for $n \geqslant 2$. For $n = 0$ and $n = 1$, the right-hand sides of (5.10) are 2 and 4, respectively. For $n = 0, 1$, it is obvious that $|\mathscr{C}_{DH}(0)| = 2$ ($\top$ and $\bot$ are double Horn) and $|\mathscr{C}_{DH}(1)| = 4$ ($\top$, $\bot$, $x_1$ and $\bar{x}_1$) hold. ∎

The number $\# \widehat{DH}(n)$ can be derived from the read-once formulas (3.3). Let $P_n$ denote the number of all ordered partitions of $n$ labeled elements into labeled nonempty sets $S_1, S_2 ..., S_d$, $d > 0$, where $P_0$ is defined to be 1. (Precisely, $P_n$ is the number of all surjections from $\{1, 2, ..., n\}$ onto any initial segment $1, 2, ..., d$ of the natural numbers.) Then,

PROPOSITION 5.2.

$$\# \widehat{DH}(n) = 2 \sum_{d=0}^{n} d! \left\{ {n \atop d} \right\} = 2P_n, \qquad n \geqslant 0.^2 \tag{5.11}$$

*Proof.* Clearly, (5.11) holds for $n = 0$. For $n > 0$, we consider formula (3.3), in which groups of literals with the same polarity alternate and form at most $n$ levels, each of which contains at least one literal. The polarity of all groups is determined by the polarity of the deepest nested level, for which there are two possibilities. Forming this and other levels amounts to partitioning the variables into nonempty sets (since literals within the same level commute in the formula). Thus the number of all formulas (3.3) on $n$ variables is given by (5.11). ∎

Notice that $P_n$ is a well-known combinatorial entity [28, Exercise 4, p. 195]. It obeys the recurrence

$$P_n = \binom{n}{1} P_{n-1} + \binom{n}{2} P_{n-2} + \cdots + \binom{n}{n} P_0 = \sum_{i=1}^{n} \binom{n}{i} P_{n-i}, \qquad n > 0. \tag{5.12}$$

Thus, $P_0 = 1$, $P_1 = 1$, $P_2 = 3$, and so on. A table of $P_n$ for $n \leqslant 14$ can be found in [13]. Using known relations among $P_n$ [13, 28], it is not difficult to derive the following results.

THEOREM 5.3.

$$\begin{aligned} \# DH(n) &= \sum_{m=0}^{n} \binom{n}{m} \# \widehat{DH}(m), && n \geqslant 0 \\ &= 4P_n, && n > 0 \\ &= 2n!(\ln 2)^{-n-1} + 4n! \Sigma_{k \geqslant 1} \Re((\ln 2 + 2\pi i k)^{-n-1}), && n > 0 \\ &\sim 2n!(\ln 2)^{-n-1}, && n \to \infty, \end{aligned}$$

*where $\Re(z)$ is the real part of the complex number $z$.* ∎

$\# DH(n)$ and $P_n$ are listed in Table 1 for $n \leqslant 7$.

[2] Here $\left\{ {n \atop m} \right\}$ are the Stirling numbers of the second kind, i.e., the number of ways to partition a set with $n$ elements into $m$ nonempty subsets; by definition, $\left\{ {0 \atop 0} \right\} = 1$.

TABLE I

Values of $\# DH(n)$ for $n \leqslant 7$

| $n$ | $\# DH(n)$ | $P_n$ |
|---|---|---|
| 0 | 2 | $1$ |
| 1 | 4 | $\binom{1}{0}1=1$ |
| 2 | 12 | $\binom{2}{0}1+\binom{2}{1}1=3$ |
| 3 | 52 | $\binom{3}{0}1+\binom{3}{1}1+\binom{3}{2}3=13$ |
| 4 | 300 | $\binom{4}{0}1+\binom{4}{1}1+\binom{4}{2}3+\binom{4}{3}13=75$ |
| 5 | 2164 | $\binom{5}{0}1+\binom{5}{1}1+\binom{5}{2}3+\binom{5}{3}13+\binom{5}{4}75=541$ |
| 6 | 18732 | $\binom{6}{0}1+\binom{6}{1}1+\binom{6}{2}3+\binom{6}{3}13+\binom{6}{4}75+\binom{6}{5}541=4683$ |
| 7 | 189172 | $\binom{7}{0}1+\binom{7}{1}1+\binom{7}{2}3+\binom{7}{3}13+\binom{7}{4}75+\binom{7}{5}541+\binom{7}{6}4683=47293$ |

*Remark* 5.1. We obtain an interesting side result, namely the identity

$$\sum_{m=1}^{n-1} \binom{n}{m} CF_{m,\, n-m} = \sum_{m=0}^{n} \binom{n}{m} P_m - 2 \;=\; 2P_n - 2, \qquad \text{for} \quad n > 0,$$

where $CF_{i,\,j}$ for $i, j > 0$ denotes the number of orientations of the complete bipartite graph $K_{i,\,j}$ which have no directed cycle. Consequently, the number of cycle-free orientations on complete bipartite graphs on $n$ vertices, given by the left-hand side, is $2P_n - 2$.

Let us next consider the number of nonisomorphic double Horn functions $\# DH^{\cong}(n)$ of $n$ variables, where $f$ and $g$ are isomorphic if and only if they become identical after a permutation of the arguments. Similarly, $\# \widehat{DH}^{\cong}(n)$ denotes such numbers when the functions depend on exactly $n$ variables.

THEOREM 5.4.

$$\# \widehat{DH}^{\cong}(n) = \begin{cases} 2 & \textit{for} \quad n = 0 \\ 2^n & \textit{for} \quad n > 0, \end{cases}$$

$$\# \widehat{DH}^{\cong}(n) = 2^{n+1} \qquad \textit{for} \quad n \geqslant 0.$$

*Proof.* Let us first consider $\# \widehat{DH}^{\cong}(n)$. For $n = 0$, we have two nonisomorphic functions $\top$ and $\bot$. Thus $\# \widehat{DH}^{\cong}(n) = 2$ holds. For $n > 0$, consider the read-once formula of (3.3). If the number of levels $d$ is fixed in the form (3.3), the number of nonisomorphic classes of double Horn functions is the number of all partitions of $n$ nonlabeled elements into labeled sets $S_1, S_2 \ldots, S_d$, where $|S_1| \geqslant 0$ and $|S_i| \geqslant 1$ for $i \geqslant 2$. Obviously, the functions with different $d$ are nonisomorphic. Thus

$$\begin{aligned} \# \widehat{DH}^{\cong}(n) &= \sum_{d=1}^{n} \binom{n-1}{d-1} + + \sum_{d=2}^{n+1} \binom{n-1}{d-2} \\ &= 2 \sum_{m=0}^{n-1} \binom{n-1}{m} \\ &= 2^n, \end{aligned} \tag{5.13}$$

where the first and second terms in (5.13) are the cases of $|S_1| \geqslant 1$ and $|S_1| = 0$, respectively.

Concerning $\# DH^{\cong}(n)$, we have

$$\# DH^{\cong}(n) = \sum_{m=0}^{n} \# \widehat{DH}^{\cong}(m) = 2 + \sum_{m=1}^{n} 2^m = 2^{n+1}. \quad \blacksquare \tag{5.14}$$

## 6. CONCLUSION AND FUTURE WORK

In this paper, we have introduced double Horn functions, which are Boolean functions $f$ such that both $f$ and its complement $\bar{f}$ are Horn. We have studied their properties and computational aspects, focusing (i) on the recognition problem, i.e., deciding whether a given (possibly restricted) formula $\varphi$ represents such a function, (ii) on their characteristic sets, and transformations between characteristic sets and formulas, and (iii) on the extension problem; i.e., given a partially defined Boolean function (pdBf) $(T, F)$, decide whether it has a double Horn extension and output a formula representing it.

In the course of this investigation, we have presented a syntactic characterization of double Horn functions. It turned out that the class $\mathscr{C}_{DH}$ of double Horn functions constitutes a fragment of the class $\mathscr{C}_{R\text{-}1}$ of read-once functions and that each double Horn function has a small and unique prime DNF. Based on this result, we derived the count of this class; there are approximately $2n!(\ln 2)^{-n-1}$ double Horn functions on $n$ variables and exactly $2^{n+1}$ nonisomorphic ones. This shows that the class $\mathscr{C}_{DH}$ is quite small (recall that there are $2^{2^n}$ Boolean functions on $n$ variables). Furthermore, we have presented a semantic characterization of double Horn functions in terms of their characteristic sets. This semantic characterization can be naturally stated as a graph property, and we obtain a 1–1 correspondence between double Horn functions and oriented complete bipartite graphs. This, combined with the above result, gives us an interesting relation between the number of ordered partitions of a set and the number of cycle-free complete bipartite digraphs.

On the computational side, we have shown that double Horn functions can be recognized efficiently from Horn formulas, that the dualizing a double Horn function $f$ (e.g., computing a DNF and/or characteristic sets of the dual function $f^d$ of $f$ from an arbitrary representation of $f$) can be done in polynomial time. However, finding a shortest double Horn extension is proved to be NP-hard. Furthermore, we have presented an algorithm that finds a double Horn extension of a pdBf $(T, F)$ in polynomial time and an algorithm that enumerates all double Horn extensions of $(T, F)$ with polynomial time delay. Utilizing this algorithm, the existence of the unique double Horn extension can also be decided in polynomial time.

Further properties of double Horn functions have been investigated in [9]. In particular, the effect of renamings (i.e., a change of polarity in part of the variables) on double Horn functions is considered there. Both the recognition problem and the existence problem for the closure of $\mathscr{C}_{DH}$ under renamings, $\mathscr{C}^R_{DH}$, and furthermore the enumeration problem, are solvable in polynomial time. This is of particular significance, since $\mathscr{C}^R_{DH}$ is a fragment of several well-known classes of

Boolean functions, namely the read-once functions, the threshold functions, the 2-monotonic functions, the renamable Horn functions, and the unate (renamable positive) functions; for all these classes except the class of threshold functions, the extension problem is known to be NP-complete [5, 3].

Some problems remain to be addressed in further work. One issue is the search for faster or simpler algorithms, for example, whether all double Horn extensions of a pdBf can be enumerated with linear time delay. Furthermore, variants of the enumeration problem (e.g., output only nonisomorphic extensions, or in a particular order) are of interest. Another issue is approximation of shortest double Horn extensions.

## ACKNOWLEDGMENTS

The authors thank I. Gessel, Ch. Krattentaler, and R. Stanley for bibliographic hints on preferential arrangements, and A. Gartus for comments on the algorithms. Moreover, they are grateful to the anonymous referees for their helpful and constructive comments which improved the presentation of this paper. In particular, they suggested a discussion on PAC learnability of double Horn functions and pointed out Refs. [2, 19] to us.

The authors gratefully acknowledge the support of the Scientific Grant in Aid by the Ministry of Education, Science, and Culture of Japan, which made the visit of the first author to Kyoto University possible. Moreover, T. Eiter acknowledges support by the Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, in the context of this work. K. Makino has been supported by research fellowships of the Japan Society for the Promotion of Science for Young Scientists.

Received January 31, 1997; final manuscript received December 18, 1997

## REFERENCES

1. Angluin, D., Frazier, M., and Pitt, L. (1990), Learning conjunctions of Horn clauses, *in* "31st Annual Symposium on Foundations of Computer Science," pp. 186–192.
2. Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. (1987), Occam's razor, *Inform. Process. Lett.* **24**, 377–380.
3. Boros, E., Ibaraki, T., and Makino, K. (1998), Error-free and best-fit extensions of partially defined Boolean functions, *Inform. and Comput.* **140**, 254–283.
4. Ceri, S., Gottlob, G., and Tanca, L. (1990), "Logic Programming and Databases," Springer-Verlag, Berlin/New York.
5. Crama, Y., Hammer, P. L., and Ibaraki, T. (1988), Cause-effect relationships and partially defined Boolean functions, *Ann. Oper. Res.* **16**, 299–326.
6. Dowling, D. W., and Gallier, J. H. (1984), Linear-time algorithms for testing the satisfiability of propositional Horn formulae, *J. Logic Program.* **3**, 267–284.
7. Dechter, R., and Pearl, J. (1992), Structure identification in relational data, *Artificial Intelligence* **58**, 237–270.
8. Eiter, T. (1995), Generating boolean $\mu$-expressions, *Acta Inform.* **32**, 171–187.

9. Eiter, T., Ibaraki, T., and Makino, K. (1996), Multi-Face Horn Functions, Technical Report CD-TR 96/95, Information Systems Department, Technical University of Vienna.

10. Ekin, O., Hammer, P. L., and Peled, U. N. (1997), Horn functions and submodular Boolean functions, *Theoret. Comput. Sci.* **175**, 257–270.

11. Golumbic, M., Hammer, P. L., Hansen, P., and Ibaraki, T. (Eds.) (1990), Horn logic, search and satisfiability, *Ann. Math. Artificial Intelligence* **1**.

12. Garey, M. R., and Johnson, D. S. (1979), "Computers and Intractability," Freeman, New York.

13. Gross, O., (1962), Preferential arrangements, *Amer. Math. Monthly* **69**, 4–8.

14. Gurvich, V. A., (1991), On repetition-free boolean functions, *Uspekhi Math. Nauk* **32**, 183–184. [Russian]

15. Gurvich, V. A., (1991), Criteria for repetition-freeness of functions in the algebra of logic, *Soviet Math. Dokl.* **43**, 721–726. [English translation of *Dokl. Akad. Nauk SSSR*, **318** (1991).]

16. Hammer, P. L., and Kogan, A. (1993), Optimal compression of propositional Horn knowledge bases: Complexity and approximation, *Artificial Intelligence* **64**, 131–145.

17. Hammer, P. L., and Kogan, A. (1992), Horn functions and their DNFs, *Inform. Process. Lett.* **44**, 23–29.

18. Hammer, P. L., and Kogan, A. (1995), Quasi-acyclic propositional Horn knowledge bases: Optimal compression, *IEEE Trans. Knowledge Data Engrg.* **7**, 751–762.

19. Helmbold, D., Sloan, R., and Warmuth, M. (1990), Learning nested differences of intersection-closed concept classes, *Mach. Learning* **5**, 165–190.

20. Horn, A. (1951), On sentences which are true of direct unions of algebras, *J. Symbolic Logic* **16**, 14–21.

21. Hunt III, H., and Stearns, R. (1990), The complexity of very simple Boolean formulas with applications, *SIAM J. Comput.* **19**, 44–70.

22. Johnson, D. S., Yannakakis, M., and Papadimitriou, C. H. (1988), On generating all maximal independent sets, *Inform. Process. Lett.* **27**, 119–123.

23. Karchmer, M., Linial, N., Newman, I., Saks, M., and Wigderson, A. (1993), Combinatorial characteriza1tion of read-once formulae, *Discrete Math.* **114**, 275–282.

24. Kautz, H.. A., Kearns, M. J., and Selman, B. (1993), Reasoning with characteristic models, *in* "Proc. Eleventh National Conference on Artificial Intelligence," pp. 34–39.

25. Kautz, H..A., Kearns, M. J., and Selman, B. (1995), Horn approximations of empirical data, *Artificial Intelligence* **74**, 129–145.

26. Kavvadias, D., Papadimitriou, C. H., and Sideri, M. (1996), On Horn envelopes and hypergraph transversals, *in* "ISAAC'93, Algorithms and Computation" (K. W. Ng *et al.*, Eds.), Lecture Notes in Computer Science, Vol. 762, pp. 399–405, Springer-Verlag, Berlin/New York.

27. Khardon, R., (1995), Translating between Horn representations and their characteristic models, *J. Artificial Intelligence Res.* **3**, 349–372.

28. Knuth, D. E. (1973), Sorting and Searching, *The Art of Computer Programming*, Vol. 3, Addison–Wesley, Reading, MA.

29. Makino, K., Hatanaka, K., and Ibaraki, T. (1995), Horn extensions of a partially defined Boolean function, RUTCOR Research Report RRR 27–95, Rutgers University.

30. Makowsky, J. (1987), Why Horn formulas matter for computer science: initial structures and generic examples, *J. Comput. System Sci.* **34**, 266–292.

31. McKinsey, J. C. C. (1943), The decision problem for some classes of sentences without quantifiers, *J. Symbolic Logic* **8**, 61–76.

32. Mundici, D. (1989), Functions computed by monotone boolean formulas with no repeated variables, *Theoret. Comput. Sci.* **66**, 113–114.

33. Pitt, L., and Valiant, L. (1988), Computational limitations on learning from examples, *Assoc. Comput. Mach.* **35**, 965–984.

34. Quine, W. (1955), A way to simplify truth functions, *Amer. Math. Monthly* **62**, 627–631.

35. Selman, B., and Kautz, H. (1991), Knowledge compilation using Horn approximation, *in* "Proc. Ninth National Conference on Artificial Intelligence," 904–909.

36. Trakhtenbrot, B. A. (1958), On the theory of repetition-free contact schemes, *Trudy Mat. Inst. Steklov.* **51**, 226–269. [in Russian]

37. Valiant, J. G. (1984), A theory of the learnable, *Comm. ACM* **27**, 1134–1142.

38. Wegener, I. (1987), "The Complexity of Boolean Functions," Wiley and Teubner, New York.