Canonical Horn Representations and Query Learning*

Marta Arias[1] and José L. Balcázar[2]

[1] LARCA Research Group, Departament LSI
Universitat Politècnica de Catalunya, Spain
marias@lsi.upc.edu

[2] Departamento de Matemáticas, Estadística y Computación
Universidad de Cantabria, Spain
joseluis.balcazar@unican.es

Abstract. We describe an alternative construction of an existing canonical representation for definite Horn theories, the *Guigues-Duquenne* basis (or GD basis), which minimizes a natural notion of implicational size. We extend the canonical representation to general Horn, by providing a reduction from definite to general Horn CNF. We show how this representation relates to two topics in query learning theory: first, we show that a well-known algorithm by Angluin, Frazier and Pitt that learns Horn CNF always outputs the GD basis independently of the counterexamples it receives; second, we build strong polynomial certificates for Horn CNF directly from the GD basis.

1 Introduction

The present paper is the result of an attempt to better understand the classic algorithm by Angluin, Frazier, and Pitt [2] that learns propositional Horn formulas. A number of intriguing questions remain open regarding this algorithm; in particular, we were puzzled by the following one: along a run of the algorithm, queries made by the algorithm depend heavily upon the counterexamples selected as answers to the previous queries. It is therefore natural to expect the outcome of the algorithm to depend on the answers received along the run. However, attempts at providing an example of such behavior consistently fail.

In this paper we prove that such attempts must in fact fail: we describe a canonical representation of Horn functions in terms of implications, and show that the algorithm of [2] always outputs this particular representation. It turns out that this canonical representation is well-known in the field of Formal Concepts, and bears the name of the authors that, to the best of our knowledge, first described it: the *Guigues-Duquenne basis* or GD basis [7, 12]. In addition, the GD basis has the important quality of being of minimum size.

The GD basis is defined for definite Horn formulas only. We extend the notion of GD basis to general Horn formulas by means of a reduction from general to

* Work partially supported by MICINN projects SESAAME-BAR (TIN2008-06582-C03-01) and FORMALISM (TIN2007-66523).

R. Gavaldà et al. (Eds.): ALT 2009, LNAI 5809, pp. 156–170, 2009.

definite Horn formulas. This reduction allows us to lift the characterization of the output of AFP as the generalized GD basis. Furthermore, the generalized GD representation provides the basis for building strong polynomial certificates with $p(m, n) = m$ and $q(m, n) = \binom{m+1}{2} + m + 1 = \binom{m+2}{2}$ for the class of general Horn formulas, extending a similar construction from [4] which applied only to definite Horn.

Some of the technical lemmas and theorems in this paper are based on previous results of [12, 7]; we credit this fact appropriately throughout this presentation. As a general overview, we have adopted the following: the "bullet" operator ($^\bullet$) of Section 3.1 is directly taken from [12], the "star" operator (*) is standard in the field of study of *Closure Spaces*, and the GD basis comes from the *Formal Concept Analysis* literature. We consider that our contribution here is threefold: first, to understand, translate, and interpret the results from these other fields; second, to recognize the connection of these results to our own; third, to draw new insights into our topic of study thanks to the fruitful combination of our own intuitions and knowledge and the adoption of these outside results.

Due to the space limit, a number of proofs, mostly of simple lemmas, have been omitted or just sketched. A longer version containing all proofs is available from the authors' webpages.

2 Preliminaries

We work within the standard framework in logic, where one is given an indexable set X of propositional variables of cardinality n, Boolean functions are subsets of the Boolean hypercube $\{0, 1\}^n$, and these functions are represented by logical formulas over the variable set in the standard way. Assignments are partially ordered bitwise according to $0 \leq 1$ (the usual partial order of the hypercube); the notation is $x \leq y$. Readers not familiar with standard definitions of assignment, assignment satisfaction or formula entailment ($\models$), literal, term, clause, etc. should consult a standard textbook, e.g., [6]. A particularity of our work is that we identify assignments $x \in \{0, 1\}^n$ with variable subsets $\alpha \subseteq X$ in the standard way by connecting the variable subsets with the bits that are set to 1 in the assignments. We denote this explicitly when necessary with the functions $x = \text{BITS}(\alpha)$ and $\alpha = \text{ONES}(x)$. Therefore, $x \models \alpha$ iff $\alpha \subseteq \text{ONES}(x)$ iff $\text{BITS}(\alpha) \leq x$.

We are only concerned with Horn functions, and their representations using conjunctive normal form (CNF). A Horn CNF formula is a conjunction of Horn clauses. A clause is a disjunction of literals. A clause is *definite Horn* if it contains exactly one positive literal, and it is *negative* if all its literals are negative. A clause is *Horn* if it is either definite Horn or negative.

Horn clauses are generally viewed as implications where the negative literals form the antecedent of the implication (a positive term), and the singleton consisting of the positive literal, if it exists, forms the consequent of the clause. Note that both can be empty; if the consequent is empty, then we are dealing with a negative Horn clause. Furthermore, we allow our representations of Horn

CNF to deviate slightly from the standard in that we represent clauses sharing the same antecedent together in one implication. Namely, an implication $\alpha \to \beta$, where both α and β are possibly empty sets of propositional variables, is to be interpreted as the conjunction of definite Horn clauses $\bigwedge_{b \in \beta} \alpha \to b$ if $\beta \neq \emptyset$, and as the negative clause $\alpha \to \Box$ if $\beta = \emptyset$.[1] A semantically equivalent interpretation is to see both sets of variables α and β as positive terms; the Horn formula in its standard form is obtained by distributivity on the variables of β. Note that $x \models \emptyset$ for any assignment x; however, this is not the case with respect to the right hand sides of nondefinite Horn clauses since, there, by convention, $\beta = \emptyset$ stands for the unsatisfiable. □

We refer to our generalized notion of conjunction of clauses sharing the antecedent as *implication*; the term *clause* retains its classical meaning (namely, a disjunction of literals). Notice that an implication may not be a clause, e.g. $(a \to bc)$ corresponds in classical notation to the formula $\neg a \vee (b \wedge c)$. Thus, $(a \to bc)$, $(ab \to c)$ and $(ab \to \emptyset)$ are Horn implications but only the latter two are Horn clauses. Furthermore, we often use sets to denote conjunctions, as we do with positive terms, also at other levels: a generic (implicational) CNF $\bigwedge_i (\alpha_i \to \beta_i)$ is often denoted in this text by $\{(\alpha_i \to \beta_i)\}_i$. Parentheses are mostly optional and generally used for ease of reading.

Clearly, an assignment $x \in \{0,1\}^n$ satisfies the implication $\alpha \to \beta$, denoted $x \models \alpha \to \beta$, if it either fails the antecedent or satisfies the consequent, that is, $x \not\models \alpha$ or $x \models \beta$ respectively, where now we are interpreting both α and β as positive terms.

A Horn function admits several syntactically different Horn CNF representations; in this case, we say that these representations are equivalent. Such representations are also known as *theories* or *bases* for the Boolean function they represent. The *size* of a Horn function is the minimum number of clauses that a Horn CNF representing it must have. The *implication size* of a Horn size is defined analogously, but allowing formulas to have implications instead of clauses. Clearly, every clause is an implication, and thus the implication size of a given Horn function is always at most that of its standard size as measured in the number of clauses. Not all Boolean functions are Horn. The following semantic characterization is a well-known classic result proved in the context of propositional Horn logic e.g. in [10]:

Theorem 1. *A Boolean function admits a Horn CNF basis if and only if the set of assignments that satisfy it is closed under bit-wise intersection.* □

An implication in a Horn CNF H is *redundant* if it can be removed from H without changing the Horn function represented. A Horn CNF is *irredundant* if it does not contain any redundant implication. Notice that an irredundant H

[1] Notice that this differs from an alternative, older interpretation [11], nowadays obsolete, in which $\alpha \to \beta$ represents the clause $(\neg x_1 \vee \ldots \vee \neg x_k \vee y_1 \vee \ldots \vee y_{k'})$, where $\alpha = \{x_1, \ldots, x_k\}$ and $\beta = \{y_1, \ldots, y_{k'}\}$. Though identical in syntax, the semantics are different; in particular, ours can only represent a conjunction of definite Horn clauses whereas the other represents a general possibly non-Horn clause.

may still contain other sorts of redundancies, such as consequents larger than strictly necessary. Such redundancies are not contemplated in this paper.

Forward chaining. We describe the well-known method of *forward chaining* for definite Horn functions [6]. Notice that it directly extends to our compressed representation where consequents of clauses can contain more than one variable. Given a definite Horn CNF $H = \{\alpha_i \to \beta_i\}_i$ and a subset of propositional variables α, we construct chains of subsets of propositional variables $\alpha = \alpha^{(0)} \subset \alpha^{(1)} \subset \cdots \subset \alpha^{(k)} = \alpha^\star$. Each $\alpha^{(i)}$ with $i > 0$ is obtained from its predecessor $\alpha^{(i-1)}$ in the following way: if $\mathrm{BITS}(\alpha^{(i-1)})$ satisfies all implications in H, then the process can stop with $\alpha^{(i-1)} = \alpha^\star$. If, on the other hand, $\mathrm{BITS}(\alpha^{(i-1)})$ violates some implication $\alpha_j \to \beta_j \in H$, then $\alpha^{(i)}$ is set to $\alpha^{(i-1)} \cup \beta_j$.

Similarly, one can construct an increasing chain of assignments $x = x^{(0)} < x^{(1)} < \cdots < x^{(k)} = x^\star$ using our bijection $\alpha^{(i)} = \mathrm{ONES}(x^{(i)})$ and $x^{(i)} = \mathrm{BITS}(\alpha^{(i)})$ for all i.

See [6] as a general reference for the following well-known results. Theorem 3 in particular refers to the fact that the forward chaining procedure is a sound and complete deduction method for definite Horn CNF.

Theorem 2. *The objects $x^\star$ and $\alpha^\star$ are well-defined and computed by the forward chaining procedure regardless of the order in which implications in H are chosen. Moreover, $x^\star$ and $\alpha^\star$ depend only on the underlying function being represented, and not on the particular choice of representation H; and for each $x^{(i)}$ and $\alpha^{(i)}$ along the way, we have that $(x^{(i)})^\star = x^\star$ and $(\alpha^{(i)})^\star = \alpha^\star$.* □

Theorem 3. *Let h be a definite Horn function, and let α be an arbitrary variable subset. Then $h \models \alpha \to b$ if and only if $b \in \alpha^\star$.* □

Closure operator and equivalence classes. It is easy to see that the $\star$ operator is extensive (that is, $x \le x^\star$ and $\alpha \subseteq \alpha^\star$), monotonic (if $x \le y$ then $x^\star \le y^\star$, and if $\alpha \subseteq \beta$ then $\alpha^\star \subseteq \beta^\star$) and idempotent ($x^{\star\star} = x^\star$, and $\alpha^{\star\star} = \alpha^\star$) for all assignments x, y and variable sets α, β; that is, $\star$ is a closure operator [4]. Thus, we refer to $x^\star$ as the *closure* of x w.r.t. a definite Horn function.

It should be always clear from the text with respect to what definite Horn function we are taking the closure, hence it is omitted from the notation used. An assignment x is said to be *closed* iff $x^\star = x$, and similarly for variable sets. Furthermore, it is not hard to see that closed elements are always positive (by construction via the forward chaining procedure, they must satisfy all implications), and assignments that are not closed are always negative (similarly, they must violate some implication). That is: $x \models H$ if and only if $x^\star = x$. This closure operator induces a partition over the set of assignments $\{0,1\}^n$ in the following straightforward way: two assignments x and y belong to the same class if $x^\star = y^\star$. This notion of equivalence class carries over as expected to the power set of propositional variables: the subsets α and β belong to the same class if $\alpha^\star = \beta^\star$. It is worth noting that each equivalence class consists of a possibly empty set of assignments that are not closed and a single closed set, its representative.

3 The Guigues-Duquenne Basis for Definite Horn

In this section we characterize and show how to build a canonical basis for definite Horn functions that is of minimum implication size. Our construction is based on the notion of *saturation*, a notion that has been used already in the context of Horn functions and seems very natural [4, 5]. It turns out that this canonical form is, in essence, the Guigues-Duquenne basis (the *GD basis*) which was introduced in [7]. Here, we introduce it in a form that is, to our knowledge, novel, although it is relatively close to the approach of [12].

We begin by defining *saturation* and then prove several interesting properties that serve as the basis for our work.

Definition 1. *Let $B = \{\alpha_i \to \beta_i\}_i$ be a basis for some definite Horn function.*

- *We say that B is* left-saturated *if the following 2 conditions hold:*
  1. $\mathrm{BITS}(\alpha_i) \not\models \alpha_i \to \beta_i$*, for all i;*
  2. $\mathrm{BITS}(\alpha_i) \models \alpha_j \to \beta_j$*, for all $i \neq j$.*

  Alternatively, it can be more succintly described by the following equivalence: a basis $\{\alpha_i \to \beta_i\}_i$ is left-saturated if $i = j \Leftrightarrow \mathrm{BITS}(\alpha_i) \not\models \alpha_j \to \beta_j$.
- *We say that B is* right-saturated *if for all i, $\beta_i = \alpha_i^\star$. Accordingly, we denote right-saturated bases with $\{\alpha_i \to \alpha_i^\star\}_i$.*
- *We say that a basis B is* saturated *iff it is left- and right-saturated.*

Example 1. Let $H = \{a \to b, b \to c, ad \to e\}$.

- H is not left-saturated: for example, the antecedent of $ad \to e$ is such that $\mathrm{BITS}(ad) \not\models a \to b$. One can already see that by including b in the antecedent of the third clause, one avoids this particular violation.
- H is not right-saturated because $a^\star = abc$ and, for example, the implication $a \to b$ is missing ac from its right-hand-side.
- The equivalent $H' = \{a \to abc, b \to bc, abcd \to abcde\}$ is saturated.

Lemma 1. *Let $B = \{\alpha_i \to \beta_i\}_i$ be a basis for some definite Horn function h.*

1. *If B is left-saturated then B is irredundant.*
2. *If B is irredundant, then $\mathrm{BITS}(\alpha_i) \not\models \alpha_i \to \beta_i$ for all i.*
3. *If B is saturated, then $\mathrm{BITS}(\alpha_i) \not\models h$ and $\mathrm{BITS}(\alpha_i^\star) \models h$ hold for all i.*
4. *If B is saturated, then $\alpha_i \subseteq \alpha_j \Rightarrow \alpha_i \subset \alpha_j$, for all $i \neq j$.*

Lemma 2. *Let $B = \{\alpha_i \to \alpha_i^\star\}_i$ be an irredundant, right-saturated basis. Then, B is left-saturated if and only if the following implication is true for all $i \neq j$: $\alpha_i \subset \alpha_j \Rightarrow \alpha_i^\star \subseteq \alpha_j$.*

The following Lemma is a variant of a result of [12] translated into our notation. We include the proof that is, in fact, missing from [12].

Lemma 3. *Let $B = \{\alpha_i \to \alpha_i^\star\}_i$ be a saturated basis for a definite Horn function. Then for all i and β it holds that $(\beta \subseteq \alpha_i$ and $\beta^\star \subset \alpha_i^\star) \Rightarrow \beta^\star \subseteq \alpha_i$.*

Proof. Let us assume that the conditions of the implication are true, namely, that $\beta \subseteq \alpha_i$ and $\beta^\star \subset \alpha_i^\star$. We proceed by cases: if β is closed, then $\beta^\star = \beta$ and the implication is trivially true since $\beta \subseteq \alpha_i$ clearly implies $\beta^\star \subseteq \alpha_i$ when $\beta^\star = \beta$. Otherwise, β is not closed. Let $\beta = \beta^{(0)} \subset \beta^{(1)} \subset \cdots \subset \beta^{(k)} = \beta^\star$ be the series of elements constructed by the forward chaining procedure described in Section 2. We argue that if $\beta^{(l)} \subseteq \alpha_i$ and $\beta^{(l)} \subset \beta^\star$, then $\beta^{(l+1)} \subseteq \alpha_i$ as well. By repeatedly applying this fact to all the elements along the chain, we arrive at the desired conclusion, namely, $\beta^\star \subseteq \alpha_i$. Let $\beta^{(l)}$ be such that $\beta^{(l)} \subseteq \alpha_i$ and $\beta^{(l)} \subset \beta^\star$. Thus $\beta^{(l)}$ violates some implication $(\alpha_k \to \alpha_k^\star) \in B$. Our forward chaining procedure assigns $\beta^{(l+1)}$ to $\beta^{(l)} \cup \alpha_k^\star$. The following inequalities hold: $\alpha_k \subseteq \beta^{(l)}$ because $\beta^{(l)} \not\models \alpha_k \to \alpha_k^\star$, $\beta^{(l)} \subseteq \alpha_i$ by assumption; hence $\alpha_k \subseteq \alpha_i$. Using Lemma 2, and noticing the fact that, actually, $\alpha_k \subset \alpha_i$ since $\beta^{(l)} \subset \alpha_i$ (otherwise we could not have $\beta^\star \subset \alpha_i^\star$), we conclude that $\alpha_k^\star \subseteq \alpha_i$. We have that $\alpha_k^\star \subseteq \alpha_i$ and $\beta^{(l)} \subseteq \alpha_i$ so that $\beta^{(l+1)} = \beta^{(l)} \cup \alpha_k^\star \subseteq \alpha_i$ as required. □

The next result characterizes our version of the canonical basis based on the notion of saturation. The proof does rely heavily on Lemma 3, which is adapted from a result from [12]. The connection to saturation and our proof technique are indeed novel.

Theorem 4. *Definite Horn functions have a unique saturated basis.*

Proof. Let B_1 and B_2 be two equivalent and saturated bases. Let $a \to a^\star$ be an arbitrary implication in B_1. We show that $a \to a^\star \in B_2$ as well. By symmetry, this implies that $B_1 = B_2$.

By Lemma 1(2), we have that $\text{BITS}(a) \not\models B_1$ and thus $\text{BITS}(a)$ must violate some implication $b \to b^\star \in B_2$, hence it must hold that $b \subseteq a$ but $b^\star \not\subseteq a$. The rest of the proof is concerned with showing that assuming $b \subset a$ leads to a contradiction. If so, then $b = a$ and thus $a \to a^\star \in B_2$ as well as desired.

Let us assume then that $b \subset a$ so that, by monotonicity, $b^\star \subseteq a^\star$. If $b^\star \subset a^\star$, then we can use Lemma 3 with $\alpha_i = a$ and $\beta = b$ and conclude that $b^\star \subseteq a$, contradicting the fact that $\text{BITS}(a) \not\models (b \to b^\star)$. Thus, it should be that $b^\star = a^\star$. Now, consider $b \to a^\star \in B_2$. Clearly b is negative (otherwise, $b = b^\star$, and then $b \to b^\star$ is redundant) and thus it must violate some implication $c \to c^\star \in B_1$, namely, $c \subseteq b$ but $c^\star \not\subseteq b$. If $c = b$, then we have $a \to a^\star \in B_1$ and $c \to c^\star$ with $c \subset a$ and $c^\star = b^\star = a^\star$ contradicting the fact that B_1 is irredundant. Thus, $c \subset b$ and so $c^\star \subseteq b^\star$. If $c^\star \subset b^\star$ then we use Lemma 3 as before but with $\alpha_i = b$ and $\beta = c$ and we conclude that $c^\star \subseteq b$. Again, this means that $b \models c \to c^\star$ contradicting the fact that b violates this implication. So the only remaining case is $c^\star = b^\star$ but this means that we have the implications $a \to a^\star \in B_1$ and $c \to c^\star \in B_1$ with $c \subset a$ but $a^\star = c^\star$ which again makes B_1 redundant. □

3.1 Constructing the GD Basis

So far, our definition of saturation only tests whether a given basis is actually saturated; we study now a saturation process to obtain the GD basis. New definitions are needed. Let H be any Horn CNF, and α any variable subset. Let

$H(\alpha)$ be those clauses of H whose antecedents fall in the same equivalence class as α, namely, $H(\alpha) = \{\alpha_i \to \beta_i \mid \alpha_i \to \beta_i \in H \text{ and } \alpha^\star = \alpha_i^\star\}$.

Given a Horn function H and a variable subset α, we introduce a new operator $\bullet$:$\alpha^\bullet$ is the closure of α *with respect to the subset of clauses* $H \setminus H(\alpha)$. That is, in order to compute $\alpha^\bullet$ one does forward chaining starting with α but one is not allowed to use the clauses in $H(\alpha)$. This operator has been used in the literature before in related contexts, for example in [12].

Example 2. Let $H = \{a \to b, a \to c, c \to d\}$. Then, $(ac)^\star = abcd$ but $(ac)^\bullet = acd$ since $H(ac) = \{a \to b, a \to c\}$ and we are only allowed to use the clause $c \to d$ when computing $(ac)^\bullet$.

Computing the GD basis of a definite Horn H. First, saturate every clause $C = \alpha \to \beta$ in H by replacing it with the implication $\alpha^\bullet \to \alpha^\star$. Then, remove possibly redundant implications, namely: (1) remove implications s.t. $\alpha^\bullet = \alpha^\star$, and (2) remove duplicates, and (3) remove subsumed implications, i.e., implications $\alpha^\bullet \to \alpha^\star$ for which there is another implication $\beta^\bullet \to \beta^\star$ s.t. $\alpha^\star = \beta^\star$ but $\beta^\bullet \subset \alpha^\bullet$.

Let us denote with $GD(H)$ the implicational definite Horn CNF obtained by applying this procedure to input H. Note that this algorithm is designed to work when given a definite Horn CNF both in implicational or standard form.

The procedure can be computed in quadratic time, since finding the closures of antecedent and consequent of each clause can be done in linear time w.r.t. the size of the initial Horn CNF H.

Example 3. Let $H = \{a \to b, a \to c, ad \to e, ab \to e\}$. We compute the closures of the antecedents: $a^\star = abce$, $(ad)^\star = abcde$, and $(ab)^\star = abce$. Therefore, $H(a) = \{a \to b, a \to c, ab \to e\}$, $H(ad) = \{ad \to e\}$, and $H(ab) = H(a)$. Thus, $a^\bullet = a$, $(ad)^\bullet = abcde$, and $(ab)^\bullet = abce$. After saturation of every clause in H, we obtain $H' = \{a \to abce, a \to abce, abcde \to abcde, abce \to abce\}$. It becomes clear that the third clause was, in fact, redundant. Also, the fourth implication is subsumed by the first two (after right-saturation) and we can group the first and second implications together into a single one. Hence, $GD(H) = \{a \to abce\}$.

In the remainder of this Section we show that the given algorithm computes the unique saturated representation of its input. First, we need a simple lemma:

Lemma 4. *Let H be any basis for a definite Horn CNF over variables $X = \{x_1, \ldots, x_n\}$. For any $\alpha, \beta, \gamma \subseteq X$, the following statements hold:*

1. $\alpha \subseteq \alpha^\bullet \subseteq \alpha^\star$;
2. *If $H \models \beta \to \gamma$, $\beta \subseteq \alpha^\bullet$; but $\beta^\star \subset \alpha^\star$, then $\gamma \subseteq \alpha^\bullet$.*

Lemma 5. *The algorithm computing $GD(H)$ outputs the GD basis of H for any definite Horn formula H.*

Proof. Let H be the input to the algorithm, and let H' be its output. We show that H' must be saturated. Let $\alpha \to \beta$ be an arbitrary implication in the output H'. Because of the initial saturation process, we can refer to this implication as $\alpha^\bullet \to \alpha^\star$. Clearly, $(\alpha^\bullet)^\star = \alpha^\star$, and H' is right-saturated. It is only left to show that H' is left-saturated. By Lemma 4, it must be that $\alpha^\bullet \subseteq \alpha^\star$, but the removal of implications of type (1) guarantees that $\alpha^\bullet \subset \alpha^\star$, thus we have that $\mathrm{BITS}(\alpha^\bullet) \not\models \alpha^\bullet \to \alpha^\star$ and Condition 1 of left-saturation is satisfied. Now let $\beta^\bullet \to \beta^\star$ be any other implication in H'. We need to show that $\mathrm{BITS}(\alpha^\bullet) \models \beta^\bullet \to \beta^\star$. Assume by way of contradiction that this is not so, and $\mathrm{BITS}(\alpha^\bullet) \models \beta^\bullet$ but $\mathrm{BITS}(\alpha^\bullet) \not\models \beta^\star$. That is, $\beta^\bullet \subseteq \alpha^\bullet$ but $\beta^\star \not\subseteq \alpha^\bullet$. If $\beta^\bullet = \alpha^\bullet$, then $\beta^\star = \alpha^\star$, contradicting the fact that both implications have survived type (2) of removal of implications in the algorithm. Thus, $\beta^\bullet \subset \alpha^\bullet$, and therefore $\beta^\star \subseteq \alpha^\star$ must hold as well. It cannot be that $\beta^\star = \alpha^\star$ because we would have that $\alpha^\bullet \to \alpha^\star$ is subsumed by $\beta^\bullet \to \beta^\star$ and thus removed from the output H' during removal of implications of type (3) (and it is not). Thus, it can only be that $\beta^\bullet \subset \alpha^\bullet$ and $\beta^\star \subset \alpha^\star$. But if $\beta^\star \subset \alpha^\star$, Lemma 4 and the fact that $H \models \beta^\bullet \to \beta^\star$ (notice that saturating clauses does not change the logical value of the resulting formula) guarantee that $\beta^\star \subseteq \alpha^\bullet$ contradicting our assumption that $\beta^\star \not\subseteq \alpha^\bullet$. It follows that H' is saturated as required. □

It is clear that $GD(H)$ has at most as many implications as H. Thus, if H is of minimum size, then so is $GD(H)$. This, together with the fact that the GD basis is unique, implies:

Theorem 5. *[7] The GD basis of a definite Horn function is of minimum implicational size.* □

4 The Guigues-Duquenne Basis in Query Learning

The classic query learning algorithm by Angluin, Frazier, and Pitt [2] is able to learn Horn CNF with membership and equivalence queries. It was proved in [2] that the outcome of the algorithm is always equivalent to the target concept. However, the following questions remain open: (1) which of the Horn CNF, among the many equivalent candidates, is output? And (2) does this output depend on the specific counterexamples given to the equivalence queries? Indeed, each query depends on the counterexamples received so far, and intuitively the final outcome should depend on that as well.

Our main result from this section is that, contrary to our first intuition, the output is always the same Horn CNF: namely, the GD basis of the target Horn function. This section assumes that the target is definite Horn, further sections in the paper lift the "definite" constraint.

4.1 The AFP Algorithm for Definite Horn CNF

We recall some aspects of the learning algorithm as described in [4], which bears only slight, inessential differences with the original in [2]. The algorithm maintains a set P of all the positive examples seen so far. The fact that the target is definite Horn allows us to initialize P with the positive example 1^n. The algorithm maintains also a sequence $N = (x_1, \ldots, x_t)$ of representative negative examples (these become the antecedents of the clauses in the hypotheses). The argument of an equivalence query is prepared from the list $N = (x_1, \ldots, x_t)$ of negative examples combined with the set P of positive examples. The query corresponds to the following intuitive bias: everything is assumed positive unless some (negative) $x_i \in N$ suggests otherwise, and everything that some x_i suggests negative is assumed negative unless some positive example $y \in P$ suggests otherwise. This is exactly the intuition in the hypothesis constructed by the AFP algorithm.

For the set of positive examples P, denote $P_x = \{y \in P \mid x \leq y\}$. The hypothesis to be queried, given the set P and the list $N = (x_1, \ldots, x_t)$, is denoted $H(N, P)$ and is defined as $H(N, P) = \{\text{ONES}(x_i) \to \text{ONES}(\bigwedge P_{x_i}) \mid x_i \in N\}$.

A positive counterexample is treated just by adding it to P. A negative counterexample y is used to either refine some x_i into a smaller negative example, or to add x_{t+1} to the list. Specifically, let

$$i := \min(\{j \mid MQ(x_j \wedge y) \text{ is negative, and } x_j \wedge y < x_j\} \cup \{t + 1\})$$

and then refine x_i into $x'_i = x_i \wedge y$, in case $i \leq t$, or else make $x_{t+1} = y$, subsequently increasing t. The value of i is found through membership queries on all the $x_j \wedge y$ for which $x_j \wedge y < x_j$ holds.

```
AFP()
N ← ()                          ▷ /* empty list */
P ← {$1^n$}                     ▷ /* top element */
t ← 0
while EQ(H(N, P)) = ("no", y)                       ▷ /* y is the counterexample */
    do if $y \not\models H(N, P)$
        then add y to P
        else find the first i such that             ▷ /* $N = (x_1, \ldots, x_t)$ */
                 $x_i \wedge y < x_i$, and          ▷ /* that is, $x_i \not\leq y$ */
                 $x_i \wedge y$ is negative          ▷ /* use membership query */
             if found
                 then $x_i \leftarrow x_i \wedge y$  ▷ /* replace $x_i$ by $x_i \wedge y$ in N */
                 else $t \leftarrow t + 1$; $x_t \leftarrow y$  ▷ /* append y to end of N */
return H(N, P)
```

Fig. 1. The AFP learning algorithm for definite Horn CNF

The AFP algorithm is described in Figure 1. In order to prove that its output is indeed the GD basis, we need the following lemmas from [4]:

Lemma 6 (Lemma 2 from [4]). *Along the running of the AFP algorithm, at the point of issuing the equivalence query, for every x_i and x_j in N with $i < j$ there exists a positive example z such that $x_i \wedge x_j \leq z \leq x_j$.* □

Lemma 7 (Variant of Lemma 1 from [4]). *Along the running of the AFP algorithm, at the point of issuing the equivalence query, for every x_i and x_j in N with $i < j$ and $x_i \leq x_j$, it holds that $\bigwedge P_{x_i} \leq x_j$.*

Proof. At the time x_j is created, we know it is a negative counterexample to the current query, for which it must be therefore positive. That query includes the implication $\mathrm{ONES}(x_i) \to \mathrm{ONES}(\bigwedge P_{x_i})$, and x_j must satisfy it, and then $x_i \leq x_j$ implies $\bigwedge P_{x_i} \leq x_j$. From that point on, further positive examples may enlarge P_{x_i} and thus reduce $\bigwedge P_{x_i}$, keeping the inequality. Further negative examples y may reduce x_i, again possibly enlarging P_{x_i} and keeping the inequality; or may reduce x_j into $x_j \wedge y$. If $x_i \not\leq x_j \wedge y$ anymore, then there is nothing left to prove. Finally, if $x_i \leq x_j \wedge y$, then $x_i \leq y$, and y is again a negative counterexample that must satisfy the implication $\mathrm{ONES}(x_i) \to \mathrm{ONES}(\bigwedge P_{x_i})$ as before, so that $\bigwedge P_{x_i} \leq x_j \wedge y$ also for the new value of x_j. □

Our key lemma for our next main result is:

Lemma 8. *All hypotheses $H(N, P)$ output by the AFP learning algorithm in equivalence queries are saturated.*

Proof. Recall that $H(N, P) = \{\mathrm{ONES}(x_i) \to \mathrm{ONES}(\bigwedge P_{x_i}) \bigm| x_i \in N\}$, where $P_{x_i} = \{y \in P \bigm| x_i \leq y\}$. Let $\alpha_i = \mathrm{ONES}(x_i)$ and $\beta_i = \mathrm{ONES}(\bigwedge P_{x_i})$ for all i so that $H(N, P) = \{\alpha_i \to \beta_i \bigm| 1 \leq i \leq t\}$.

First we show that $H(N, P)$ is left-saturated. To see that $x_i \not\models \alpha_i \to \beta_i$ it suffices to note that $x_i < \bigwedge P_{x_i}$ since x_i is negative but $\bigwedge P_{x_i}$ is positive by Theorem 1, being an intersection of positive examples; thus, these two assignment must be different.

Now we show that $x_i \models \alpha_j \to \beta_j$, for all $i \neq j$. If $x_i \not\models \alpha_j$, then clearly $x_i \models \alpha_j \to \beta_j$. Otherwise, $x_i \models \alpha_j$ and therefore $x_j \leq x_i$. If $i < j$, then by Lemma 6 we have that $x_i \wedge x_j \leq z \leq x_j$ for some positive z. Then, $x_i \wedge x_j = x_j \leq z \leq x_j$, so that $x_j = z$, contradicting the fact that x_j is negative whereas z is positive. Otherwise, $j < i$. We apply Lemma 7: it must hold that $\bigwedge P_{x_j} \leq x_i$. Thus, in this case, $x_i \models \alpha_j \to \beta_j$ as well because $x_i \models \beta_j = \mathrm{ONES}(\bigwedge P_{x_j})$.

It is only left to show that $H(N, P)$ is right-saturated. Clearly, $H(N, P)$ is consistent with N and P, that is, $x \not\models H(N, P)$ for all $x \in N$ and $y \models H(N, P)$ for all $y \in P$. Take any $x \in N$ contributing the implication $\mathrm{ONES}(x) \to \mathrm{ONES}(\bigwedge P_x)$ to $H(N, P)$. We show that it is right-saturated, i.e., $\bigwedge P_x = x^\star$, where the closure is taken with respect to $H(N, P)$. We note first that $H(N, P) \models \mathrm{ONES}(x) \to (\mathrm{ONES}(x))^\star$ since the closure is taken w.r.t. implications in $H(N, P)$. By the construction of $H(N, P)$, all examples $y \in P_x$ must satisfy it, hence they must satisfy the implication $\mathrm{ONES}(x) \to (\mathrm{ONES}(x))^\star$ as well. Therefore, since $y \models \mathrm{ONES}(x)$ we must have that $y \models (\mathrm{ONES}(x))^\star$, or equivalently, that $x^\star \leq y$. This is true for every such y in P_x and thus $x^\star \leq \bigwedge P_x$. On the other hand, it is obvious that $\bigwedge P_x \leq x^\star$ since the implication $\mathrm{ONES}(x) \to \mathrm{ONES}(\bigwedge P_x)$ of $H(N, P)$ guarantees that all the variables in $\bigwedge P_x$ are included in the forward chaining process in the final $x^\star$. So we have $x^\star \leq \bigwedge P_x \leq x^\star$ as required. □

Putting Theorem 4 and Lemma 8 together, we obtain:

Theorem 6. *AFP, run on a definite Horn target, always outputs the GD basis of the target concept.* □

5 A Canonical Basis for General Horn

Naturally, we wish to extend the notion of saturation and GD basis to general Horn functions. We do this via a a prediction-with-membership reduction [3] from general Horn to definite Horn, and use the corresponding intuitions to define a GD basis for general Horn. We use this reduction to generalize our AFP algorithm to general Horn CNF, and as a consequence one obtains that the generalized AFP always outputs a saturated version of the target function. Indeed, for the generalized AFP it is also the case that the output is only dependent on the target, and not on the counterexamples received along the run. Finally, we contruct strong polynomial certificates for general Horn functions direclty in terms of the generalized GD basis, thus generalizing our earlier result of [4].

5.1 Reducing General Horn CNF to Definite Horn CNF

In this section we describe the intuition of the representation mapping, which we use in the next section to obtain a canonical basis for general Horn functions.

For any general Horn CNF H over n propositional variables, e.g. $X = \{x_i \mid 1 \leq i \leq n\}$, we construct a definite Horn H' over the set of $n + 1$ propositional variables $X' = X \cup \{\mathbf{f}\}$, where $\mathbf{f}$ is a new "dummy" variable; in essence $\mathbf{f}$ represents the *false* (that is, empty) consequent of the negative clauses in H. The relationship between the assignments for H and H' are as follows: for assignments of $n + 1$ variables xb where x assigns to the variables in X and b is the truth value assigned to $\mathbf{f}$, $x0 \models H'$ if and only if $x \models H$, whereas $x1 \models H'$ if and only if $x = 1^n$.

Define the implication $C_\mathbf{f}$ as $\mathbf{f} \rightarrow X'$. Let H_d be the set of definite Horn clauses in H, and $H_n = H \setminus \{H_d\}$ the negative ones. Define the mapping g as

$$g(H) = H_d \cup \{\neg C \rightarrow X' \mid C \in H_n\} \cup \{C_\mathbf{f}\}.$$

That is, $g(H)$ includes the definite clauses of H, the special implication $C_\mathbf{f}$, and the clauses C that are negative are made definite by forcing all the positive literals, including $\mathbf{f}$, into them. Clearly, the resulting $g(H)$ is definite Horn. Observe that that the new implication $C_\mathbf{f}$ is saturated and the ones coming from H_n are right-saturated. Observe also that g is injective: given $g(H)$, we recover H by removing the implication $C_\mathbf{f}$, and by removing all positive literals from any implications containing $\mathbf{f}$. Clearly, $g^{-1}(g(H)) = H$, since g^{-1} is removing all that g adds.

5.2 Constructing a GD-like Basis for General Horn CNF

The notion of left-saturation translates directly into general Horn CNF:

Definition 2. *Let $B = \{\alpha_i \rightarrow \beta_i\}_i$ be a basis for some general Horn function. Notice that now β_i can possibly be empty (it is empty for the negative clauses). Then, B is* left-saturated *if the following two conditions hold:*

1. $\mathrm{BITS}(\alpha_i) \not\models \alpha_i \rightarrow \beta_i$, *for all i;*

2. $\mathrm{BITS}(\alpha_i) \models \alpha_j \to \beta_j$*, for all* $i \neq j$*.*

For a definite Horn CNF H, right-saturating a clause $\alpha \to \beta$ essentially means that we add to its consequent everything that is implied by its antecedent, namely $\alpha^\star$. This can no longer be done in the case of general Horn CNF, since we need to take special care of the negative clauses. If $\beta = \emptyset$, we cannot set β to $\alpha^\star$ without changing the underlying Boolean function being represented. The closure $x^\star$ of an assignment x is defined as the closure with respect to all definite clauses in the general Horn CNF. It is useful to continue to partition assignments x in the Boolean hypercube according to their closures $x^\star$; however, in the general Horn case, we distinguish a new class (the negative class) of closed assignments that are actually negative, that is, it is possible now that $x^\star \not\models H$. These assignments are exactly those that satisfy all definite clauses of H but violate negative ones. Based on this, the negative clauses (those with antecedent α such that $\mathrm{BITS}(\alpha^\star) \not\models B$) should be left unmodified, and the definite clauses (those whose antecedents α are such that $\mathrm{BITS}(\alpha^\star) \models B$) should be right-saturated. Thus, the definition is:

Definition 3. *Let* $B = \{\alpha_i \to \beta_i\}_i$ *be a basis for some general Horn function. Then,* B *is* right-saturated *if, for all* i*,* $\beta_i = \emptyset$ *if* $\alpha_i^\star \not\models B$*, and* $\beta_i = \alpha_i^\star$ *otherwise.*

As for the definite case, "saturated" means that the general Horn CNF in question is both left- and right-saturated. We must see that this is the "correct" definition in some sense:

Lemma 9. *A basis* H *is saturated iff* $H = g^{-1}(GD(g(H)))$*.*

Proof. First let us note that the expression $g^{-1}(GD(g(H)))$ is well-defined. We can always invert g on $GD(g(H))$, since saturating $g(H)$ does not modify $C_{\mathbf{f}}$ (already saturated) and it does not touch the positive literals of implications containing $\mathbf{f}$ since these are right-saturated. Therefore, we can invert it since the parts added by g are left untouched by the construction of $GD(g(H))$.

We prove first that if H is saturated then $H = g^{-1}(GD(g(H)))$. Assume, then, that H is saturated but $H \neq g^{-1}(GD(g(H)))$. Applying g, which is injective, this can only happen if $GD(g(H)) \neq g(H)$, namely, $g(H)$, as a definite Horn CNF, differs from its own GD basis and, hence, it is not saturated: it must be because some implications other than $C_{\mathbf{f}}$ is not saturated, since this last one is saturated by construction. Also the ones containing $\mathbf{f}$ in their consequents are right-saturated, so no change happens in the right-hand-sides of these implications when saturating $g(H)$). This means that when saturating we must add a literal different from $\mathbf{f}$ to the right-hand-side of an implication not containing $\mathbf{f}$ or to the left-hand-side of an implication. In both cases, this means that the original H could not be saturated either, contradicting our assumption.

It is only left to show that an H such that $H = g^{-1}(GD(g(H)))$ is indeed saturated. By way of contradiction, assume that H is not saturated but $H = g^{-1}(GD(g(H)))$. Applying g to both sides, we must have that $g(H) = GD(g(H))$ so that $g(H)$ is actually saturated. Notice that the only difference between H

and $g(H)$ is in the implication $C_{\mathbf{f}}$ and the right-hand-sides of negative clauses in H; $g(H)$ being left-saturated means that so must be H since the left-hand-sides of H and $g(H)$ coincide exactly (ignoring $C_{\mathbf{f}}$ naturally). Therefore, H is left-saturated as well. It must be that H is not right-saturated, that is, it is either missing some variable in some non-empty consequent, or some clause that should be negative is not. In the first case, then $g(H)$ is missing it, too, and it cannot be saturated. In the second case, then there is a redundant clause in H contradicting the fact that H is left-saturated (see Lemma 1(1)). In both cases we arrive at a contradiction, thus the lemma follows. □

This gives us a way to compute the saturation (that is, the GD basis) of a given general Horn CNF:

Theorem 7. *General Horn functions have a unique saturated basis. This basis, which we denote $GD(H)$, can be computed by $GD(H) = g^{-1}(GD(g(H)))$.*

Proof. If H is saturated then $H = g^{-1}(GD(g(H)))$. The uniqueness of such an H follows from the following facts: first, $g(H)$ and $g(H')$ are equivalent whenever H and H' are equivalent; second, $GD(g(H))$ is unique for the function represented by H (Theorem 4) and third, g^{-1} is univocally defined since g is injective. □

Example 4. Let H be the general Horn CNF $\{a \to b, a \to c, abc \to \emptyset\}$. Then,

- $g(H) = \{a \to b, a \to c, abc \to abc\mathbf{f}, \mathbf{f} \to abc\mathbf{f}\}$;
- $GD(g(H)) = \{a \to abc\mathbf{f}, \mathbf{f} \to abc\mathbf{f}\}$;
- $GD(H) = g^{-1}(GD(g(H))) = \{a \to \emptyset\}$.

Similarly to the case of definite Horn functions, $GD(H)$ does not increase the number of new implications, and therefore if H is of minimum size, $GD(H)$ must be of minimum size as well. This, together with the uniqueness of saturated representation implies that:

Theorem 8. *The GD basis of a general Horn function is of minimum implicational size.* □

5.3 The AFP Algorithm for General Horn CNF

We study now the AFP algorithm operating on general Horn CNF, by following a detour: we obtain it via reduction to the definite case.

We consider, therefore, an algorithm that, for target a general Horn function H, simulates the version of AFP algorithm from Figure 1 on its definite transformation $g(H)$, where g is the representation transformation from Section 5.1. It has to simulate the membership and equivalence oracles for definite Horn CNF that the underlying algorithm expects, by using the oracles that it has for general Horn.

Initially, we set $P = \{1^{n+1}\}$, and $N = (0^n 1)$ since we know that $g(H)$ is definite and must contain the implication $\mathbf{f} \to X \cup \{\mathbf{f}\}$ by construction. In essence, the positive assignment $1^{n+1} = \mathbf{f}^\star$ and the negative $0^n 1 = \mathbf{f}^\bullet$ guarantee that the implication $C_{\mathbf{f}}$ is included in every hypothesis $H(N, P)$ that the simulation outputs as an equivalence query.

In order to deal with the queries, we use two transformations: we must map examples over the $n+1$ variables, asked as membership queries, into examples over the original example space over n variables, although in some cases are able to answer the query directly as we shall see. Upon asking $x0$ as membership query for $g(H)$, we pass on to H the membership query about x. Membership queries of the form $x1$ are answered always negatively, except for 1^{n+1} which is answered positively (in fact query 1^{n+1} never arises anyway, because that example is in P from the beginning). Conversely, n-bit counterexamples x from the equivalence query with H are transformed into $x0$. The equivalence queries themselves are transformed according to g^{-1}. It is readily checked that all equivalence queries belong indeed to the image set of g since $C_{\mathbf{f}} \in H(N, P)$.

All together, these functions constitute a prediction-with-membership (pwm) reduction from general Horn to definite Horn, in the sense of [3]. It is interesting to note that if we unfold the simulation, we end up with the original algorithm by Angluin, Frazier and Pitt [2] (obviously, with no explicit reference to our "dummy" $\mathbf{f}$).

Therefore, the outcome of AFP on a general Horn target H comes univocally determined by the outcome of AFP on the corresponding definite Horn function $g(H)$; combining this fact with Theorems 6 and 7 leads to:

Theorem 9. *The AFP algorithm always outputs the GD basis of the target concept.* □

5.4 Certificates for General Horn CNF

The certificate dimension of a given concept class is closely related to its learnability in the model of learning from membership and equivalence queries [1,8,9]. Informally, a certificate for a class $\mathcal{C}$ of concepts of size at most m is a set of (labeled) assignments that proves that concepts consistent with it must be outside $\mathcal{C}$. The polynomial $q(m, n)$ used below quantifies the cardinality of the certificate set in term of m, the size of the class, and n, the number of variables in the class. The polynomial $p(m, n)$ quantifies the expansion in size allowed in the hypotheses. In this paper, $p(m, n) = m$ and thus we construct *strong* certificates.

In [4] we show how to build strong certificates for definite Horn CNF. Here, we extend this to general Horn CNF, and describe the certificates directly in terms of the generalized GD basis. Due to space limit, we only sketch the proof.

Theorem 10. *The class of general Horn CNF has strong polynomial certificates with $p(m, n) = m$ and $q(m, n) = \binom{m+1}{2} + m + 1 = \binom{m+2}{2}$.*

Proof (Sketch). The argumentation follows, essentially, the same steps as the analogous proof in [4], because, by Lemma 9, the GD basis in the general case is saturated, and therefore all required facts carry over to the general case. Let f be a Boolean function that cannot be represented with m Horn implications.

If f is not Horn, then three assignments $x, y, x \wedge y$ such that $x \models f$, $y \models f$ but $x \wedge y \not\models f$ suffice. Otherwise, f is a general Horn CNF of implicational size strictly greater than m. Assume that f contains at least $m+1$ non-redundant and possibly negative implications $\{\alpha_i \rightarrow \beta_i\}$. We define the certificate for f:

$$\begin{aligned} Q_f = &\{x_i^\bullet, x_i^\star \mid 1 \leq i \leq m+1, x_i = \mathrm{BITS}(\alpha_i), \beta_i \neq \emptyset\} \\ &\cup \{x_i^\bullet \mid 1 \leq i \leq m+1, x_i = \mathrm{BITS}(\alpha_i), \beta_i = \emptyset\} \\ &\cup \left\{x_i^\bullet \wedge x_j^\bullet \;\middle|\; 1 \leq i < j \leq m+1\right\} \end{aligned}$$

It is illustrative to note the relation between this set of certificates for f and its GD basis: the assignments $x_i^\bullet$ and $x_i^\star$ correspond exactly to the left and right-hand-sides of the (saturated) definite implications in $GD(f)$. For negative clauses, only the (saturated) left-hand-side of the implication $x_i^\bullet$ matters. □

References

1. Angluin, D.: Queries revisited. Theoretical Computer Science 313, 175–194 (2004)
2. Angluin, D., Frazier, M., Pitt, L.: Learning conjunctions of Horn clauses. Machine Learning 9, 147–164 (1992)
3. Angluin, D., Kharitonov, M.: When won't membership queries help? Journal of Computer and System Sciences 50(2), 336–355 (1995)
4. Arias, M., Balcázar, J.L.: Query learning and certificates in lattices. In: Freund, Y., Györfi, L., Turán, G., Zeugmann, T. (eds.) ALT 2008. LNCS (LNAI), vol. 5254, pp. 303–315. Springer, Heidelberg (2008)
5. Arias, M., Feigelson, A., Khardon, R., Servedio, R.A.: Polynomial certificates for propositional classes. Inf. Comput. 204(5), 816–834 (2006)
6. Chang, C.-L., Lee, R.C.-T.: Symbolic Logic and Mechanical Theorem Proving. Academic Press, Inc., Orlando (1973)
7. Guigues, J.L., Duquenne, V.: Familles minimales d'implications informatives resultants d'un tableau de donnees binaires. Math. Sci. Hum. 95, 5–18 (1986)
8. Hegedüs, T.: On generalized teaching dimensions and the query complexity of learning. In: Proceedings of the Conference on Computational Learning Theory, pp. 108–117. ACM Press, New York (1995)
9. Hellerstein, L., Pillaipakkamnatt, K., Raghavan, V., Wilkins, D.: How many queries are needed to learn? Journal of the ACM 43(5), 840–862 (1996)
10. Khardon, R., Roth, D.: Reasoning with models. Artificial Intelligence 87(1-2), 187–213 (1996)
11. Wang, H.: Toward mechanical mathematics. IBM Journal for Research and Development 4, 2–22 (1960)
12. Wild, M.: A theory of finite closure spaces based on implications. Advances in Mathematics 108, 118–139 (1994)