

Enumeration Problems Related to Ground Horn Theories

Nachum Dershowitz¹, Guan-Shieng Huang², and Mitchell A. Harris³

¹ School of Computer Science
Tel Aviv University
Ramat Aviv 69978, Israel
`Nachum.Dershowitz@cs.tau.ac.il`

² Department of Computer Science and Information Engineering
National Chi Nan University
Puli, 545 Nantou, Taiwan R.O.C.
`shieng@ncnu.edu.tw`

³ Institute for Theoretical Computer Science,
TU Dresden,
01062 Dresden, Germany.
`harris@tcs.inf.tu-dresden.de`

Abstract.

1 Introduction

2 Canonical Propositional Systems

Let V be a set of n propositional variables. A *nontrivial monomial* over V is the product of some variables in V . We say that both 1 and 0 are the trivial monomials and take the convention that 1 is the product of zero variables. A *binomial equation* is an equality of monomials. Let \mathcal{E} be a set of binomial equations over V . If, say, $V = \{x, y\}$, we would have

$$\mathcal{E} \subseteq \{xy = x, xy = y, xy = 1, xy = 0, x = y, x = 1, x = 0, y = 1, y = 0, 1 = 0\}.$$

We can apply the Gröbner basis construction to $\mathcal{E} \cup \{xx = x\}_{x \in V}$ and a given ordering on monomials, and eventually we will get a unique canonical system for \mathcal{E} and the particular choice of ordering. How many distinct canonical systems are there over n variables? At first glance, it seems that this question is very difficult. There are $2^n(2^n + 1)/2$ \mathcal{E} 's, and some may have the same canonical system. We have to retain those that are in canonical form and rule out those that are not.

So that this will not be the proverbial search for a needle in a haystack, we take the following approach: The key point is, this enumeration problem is in fact equivalent to *counting the number of distinct Horn functions of n variables*. A SAT solver (or more precisely, a SAT enumerator) can help to count this number.

The following lemma reveals the fact that \mathcal{E} indeed defines an equivalent set of Horn clauses.

Lemma 1. *Given any set of binomial equations, there is a set of Horn clauses that defines the same constraint.*

Proof. Given any monomial m , we use $x \in m$ to denote that variable x appears in m . We let **true** $\in 1$ and **false** $\in 0$. Let $c(m)$ be the conjunction of elements in m . Now for any binomial equation $m_1 = m_2$, the corresponding Horn clauses are $c(m_1) \Rightarrow x$ for all $x \in m_2$ and $c(m_2) \Rightarrow y$ for all $y \in m_1$. \square

Conversely, given any set of Horn clauses, the corresponding set of binomial equations can be found by the following transformation: Given $x_1 \wedge \dots \wedge x_j \Rightarrow y$, produce $x_1 \dots x_j = yx_1 \dots x_j$.

Lemma 1 says that each set of binomial equations corresponds to a presentation of Horn clauses. Furthermore, there is a one-one correspondence between the canonical systems of binomial equations and the Boolean functions that satisfy the system. Lemma 1 implies that these functions are in fact constrained by Horn clauses. Let us call a Boolean function *Horn*, if it can be expressed as a conjunction of Horn clauses. As there are many equivalent sets of binomial equations but only one is canonical, there are many equivalent presentations made of Horn clauses but only one Horn function. We are interested in counting canonical sets of binomial equations, or canonical Horn presentations. Either way, the canonical set or canonical presentation is the canonical representation of a Horn function. Therefore, our enumeration problem is identical to count the number of distinct Horn functions over V .

Given a Horn function f , we collect the vectors that f maps to 1 and call that collection the *Horn set* associated with f . Let $\mathbf{t}_1, \mathbf{t}_2$, and \mathbf{t}_3 be vectors in $\{0, 1\}^n$. We say that \mathbf{t}_3 is the *meet* of \mathbf{t}_1 and \mathbf{t}_2 if \mathbf{t}_3 is obtained by performing the *logical-and* operation on each individual coordinate of \mathbf{t}_1 and \mathbf{t}_2 . A well-known theorem (cf. [1]) characterizes the Horn sets as follows:

Lemma 2. *A set of vectors is a Horn set iff it is closed under the meet operation.*

Hence, whether or not a set of vectors is Horn can be tested by the meet criterion in Lemma 2. As there is a one-one correspondence between Horn functions and Horn sets, our problem is further reduced to the problem of counting Horn sets.

Now we state how to encode our counting problem in SAT. For each vector μ in $\{0, 1\}^n$, we associate with it a predicate P_μ which means μ is included in the current Horn set. For any \mathbf{t}_1 and \mathbf{t}_2 , we generate a clause $\neg P_{\mathbf{t}_1} \vee \neg P_{\mathbf{t}_2} \vee P_{\mathbf{t}_3}$ where \mathbf{t}_3 is the meet of \mathbf{t}_1 and \mathbf{t}_2 . This set of clauses asserts the closure property of the meet operation. And then we could feed the set of clauses into a #SAT solver that counts the number of satisfying models. Note that some clauses may be redundant, since \mathbf{t}_3 may be identical to \mathbf{t}_1 or \mathbf{t}_2 . Therefore we generate fewer than 4^n clauses.

Observe that the generated clauses are indeed Horn clauses. Therefore, a reasonable DPLL-based #SAT solver cannot fail to find a model for any branch of the search, and as one can see the number of models accumulates quickly.

There are four variations for counting the number of canonical systems over n variables:

1. $f(n)$ without constants 1 and 0 in the systems (i.e., no $m = 1$ and $m' = 0$, where m and m' are monomials);
2. $F(n)$ without constant 1 (i.e., no $m = 1$, but may or may not have $m' = 0$);
3. $g(n)$ without constant 0 (i.e., no $m' = 0$, but may or may not have $m = 1$);
4. $G(n)$ with both 1 and 0 (i.e., may or may not have $m = 1$ and $m' = 0$).

Counting semi-lattices (idempotent commutative semigroups) with n generators is Case 1; Case 3 is idempotent commutative monoids; counting the number of Horn theories is Case 4.

There are some relations between these algebras:

$$\begin{aligned} G(n) &= 2g(n) \\ F(n) &= 2f(n), \end{aligned}$$

since $m = 0$ is dual to $m = V$ in the canonical systems, where m is any monomial and V is the product of all variables. This fact can also be seen from their SAT encodings as described above. The monomial $m' = 0$ forbids to select the vector $\mathbf{1}$ (a vector with all 1s) in a Horn set, and selecting the vector $\mathbf{1}$ in a Horn set forbids the existence of $m' = 0$ in the canonical system. And in our SAT encoding, the predicate for $\mathbf{1}$ (i.e., $P_{\mathbf{1}}$) does not appear since it only occurs in the tautology clause which are removed. Therefore $P_{\mathbf{1}}$ is a free variable that can be set to either true or false, and this is just the case for F and G .

Case 3 can also be reduced to Case 1, and Case 4 can be reduced to Case 2, since an equation $m = 1$ can also be resolved. They have the following relationships:

$$\begin{aligned} g(n) &= \sum_{0 \leq k \leq n} C_k^n f(n - k), \\ G(n) &= \sum_{0 \leq k \leq n} C_k^n F(n - k), \end{aligned}$$

where C_k^n is the binomial coefficient. Observe that $m = 1$ forces all variables appearing in m to be 1. Suppose k variables are forced to be 1. Those variables can be removed; hence we get a Case (2) system with $n - k$ variables. There are C_k^n ways to select k out of n variables to set 1. Therefore we only have one system to tackle.

We have written a small program that generates clauses that guarantee closure, which are then sent to a CNF SAT solver to count the total number of satisfiable truth assignments. Here are some of the computed numbers:

n	0	1	2	3	4	5
$g(n)$	1	2	7	61	2480	1,385,552
$f(n)$	1	1	4	45	2271	1,373,701

3 Discussion

References

1. Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley Publishing Company, Reading, MA, 1994.