# Yale University
# Department of Computer Science

P.O. Box 208205
New Haven, CT 06520–8285


**Counting Predicates of Conjunctive Complexity One**


Michael Fischer René Peralta



YALEU/DCS/TR-1222
December 2001
Revised February 2002



This research was supported in part by NSF grant CCR–0081823.

# Counting Predicates of Conjunctive Complexity One

Michael Fischer
fischer-michael@cs.yale.edu

René Peralta
peralta-rene@cs.yale.edu

## 1 Introduction

The number of conjunctions necessary and sufficient to build a circuit for a Boolean function $f : GF_2^n \to GF_2^m$ over the basis $(\oplus, 1, \wedge)$ is called the *conjunctive complexity* of $f$, denoted by $C_\wedge(f)$. The conjunctive complexity of functions is important to cryptographic applications. So-called "discreet proofs of knowledge" (of a secret $x$ defined by a function $f$ and a value $f(x)$) have length linear in $C_\wedge(f)$. (See [1, 2].)

Another reason to study conjunctive complexity is that it provides an upper bound on the computational complexity of Boolean functions. If a function $f$ has conjunctive complexity $O(\log(n))$ then, for all $x$ in the domain of $f$, an element of the pre-image of $f(x)$ can be found in polynomial-time. This is because once the values of inputs to all conjunctions in a circuit are known, an input corresponding to a given output can be found using Gaussian elimination over $GF_2$. Thus, one-way functions (if they exist) have super-logarithmic conjunctive complexity. Functions with "low" conjunctive complexity may be invertible in practice via heuristics that search through feasible input values for conjunctions on the corresponding circuits.

A *Boolean predicate* is a $\{0, 1\}$-valued Boolean function. In this note, we consider the question of how many predicates can be computed with only one conjunction $(\wedge)$ and an arbitrary number of exclusive-or $(\oplus)$ gates. It is known that there are at most $2^{k^2+2k+2kn+n+1}$ predicates on $n$ bits which can be computed with $k$ conjunctions [3]. For $k = 1$ this upper bound yields $16 \cdot 2^{3n}$. We show that the actual number of such predicates is exactly $2\binom{2^n}{3}$ for $n \geq 2$.

## 2 Polynomials

To count the number of predicates with conjunctive complexity $k$, we establish a bijection between predicates and a class of polynomials, find the cardinality of that class, and apply Theorem 1. For this approach to work, we need a clear notion of polynomial and when two polynomials are considered to be distinct.

Let $\mathbf{x} = x_1, \ldots, x_n$ be an $n$-tuple of variables. A *monomial* over $\mathbf{x}$ is an expression $q = x_1^{k_1} \cdot \ldots \cdot x_n^{k_n}$. A *polynomial* $p$ over $\mathbf{x}$ with coefficients in $GF_2$ is a linear combination of monomials over $\mathbf{x}$. That is,

$$p(\mathbf{x}) = \sum_{k_1=0}^{t} \cdots \sum_{k_n=0}^{t} a_{k_1,\ldots,k_n} x_1^{k_1} \cdot \ldots \cdot x_n^{k_n},$$

This research was supported in part by NSF grant CCR–0081823.

where $a_{k_1,\ldots,k_n} \in \{0,1\}$. We call $a_{k_1,\ldots,k_n} x_1^{k_1} \cdot \ldots \cdot x_n^{k_n}$ a *term* of $p$ if $a_{k_1,\ldots,k_n} \neq 0$. Two polynomials are *equal* if they have the same terms. Thus, $x+x=0$ since $x+x=2x=0x=0$ over $GF_2$. However $x^2 \neq x$ since $x$ and $x^2$ are distinct monomials. $p$ is the *zero polynomial* if it has no terms. The *degree* (resp. *total degree*) of a non-zero polynomial $p$ is the largest integer $t$ such that some term in $p$ has degree (resp. *total degree*) $t$. The zero polynomial has degree 0.

Each polynomial $p$ over $\mathbf{x}$ *represents* a unique predicate $f_p : GF_2^n \to GF_2$ defined by the usual polynomial evaluation rules over $GF_2$. That is, if $u_1, \ldots, u_n \in GF_2$, then $f_p(u_1, \ldots, u_n)$ is obtained by replacing each $x_i$ in $p$ by $u_i$ and then evaluating the resulting expression in $GF_2$. Thus, the polynomial $x+y$ represents the addition predicate over $GF_2$, and $x+1$ represents the Boolean negation predicate. Note that each predicate in general has many polynomial representations. For example, $x$ and $x^2$ both represent the identity function over $GF_2$.

Polynomials $p$ and $q$ over $\mathbf{x}$ are *equivalent* over $GF_2$, written $p \equiv q$, if $f_p = f_q$. A polynomial is *square-free* if its degree is at most one, that is, no variable appears to a power higher than one. Because $f_x = f_{x^2}$ over $GF_2$, it is easy to see that replacing all non-zero exponents in $p$ with 1 results in an equivalent square-free polynomial. Hence, every Boolean predicate can be represented by a square-free polynomial. A simple counting argument shows that there are exactly $2^{2^n}$ square-free polynomials over $x_1, \ldots, x_n$ and the same number of $n$-argument Boolean predicates. Hence, the mapping $p \mapsto f_p$ is a bijection from square-free polynomials to Boolean predicates. In the rest of this paper, we will assume that all polynomials are square-free. Where there is no confusion, we will identify polynomials with the functions that they represent.

A predicate $f(\mathbf{x})$ is said to be *positive* (resp. *negative*) if $f(\mathbf{0}) = 0$ (resp. $f(\mathbf{0}) = 1$). Similarly, a polynomial $p$ over $GF_2$ is said to be *positive* (resp. *negative*) if the predicate $f_p$ is *positive* (resp. *negative*). Thus, a polynomial is positive if and only if it has no constant term. A polynomial $p$ is *linear* if its total degree is at most 1. Note that a positive linear polynomial is just a sum of distinct variables. Let $\nu(p)$ be the set of variables appearing in $p$. Polynomials $p$ and $q$ are *disjoint* if $\nu(p) \cap \nu(q) = \emptyset$.

# 3 Circuits and Conjunctive Complexity

A *circuit* $\mathcal{C}$ over the basis $(\oplus, 1, \wedge)$ is a directed acyclic graph with designated *output* nodes $y_1, \ldots y_m$. Initial nodes are labeled by input variables $\mathbf{x} = x_1, \ldots, x_n$ or the constant basis element 1. Non-initial nodes, called *gates*, have in-degree 2 and are labeled by basis elements $\oplus$ or $\wedge$. The number of $\wedge$-gates in $\mathcal{C}$ is denoted by $\#_\wedge(\mathcal{C})$. A node $u$ *depends* on a node $v$ if there is a directed path from $v$ to $u$ in $\mathcal{C}$.

Each node $v$ of $\mathcal{C}$ computes a Boolean predicate $f_v(\mathbf{x})$ in the usual way. We say that $\mathcal{C}$ *computes* the Boolean function $f_\mathcal{C}$, where $f_\mathcal{C}(\mathbf{x}) = (f_{y_1}(\mathbf{x}), \ldots, f_{y_m}(\mathbf{x}))$. The *conjunctive complexity* $C_\wedge(f)$ of $f$ is the least integer $k$ for which there is a circuit $\mathcal{C}$ with $f_\mathcal{C} = f$ and $\#_\wedge(\mathcal{C}) = k$.

For any natural number $k$, the set of predicates with conjunctive complexity $k$ can be partitioned into positive and negative predicates. Both subsets have the same cardinality since the output of a circuit can be negated using a $\oplus$-gate whose other input is fixed to 1. This proves:

**Theorem 1** *The number of predicates with conjunctive complexity $k$ is exactly twice the number of positive such predicates.*

A circuit $\mathcal{C}$ is *constant-free* if it has no nodes labeled by 1. Constant-free circuits obviously compute positive predicates. The following lemma shows that converting a circuit for a positive polynomial into an equivalent constant-free circuit does not increase the number of $\wedge$-gates.

**Lemma 2** *Let $\mathcal{C}$ be a circuit computing a positive predicate $f(\mathbf{x})$. Then there is a constant-free circuit $\mathcal{C}'$ that also computes $f$, and $\#_\wedge(\mathcal{C}') \leq \#_\wedge(\mathcal{C})$.*

*Proof.* We proceed by induction on $k$, the number of $\wedge$-gates in $\mathcal{C}$.

If $k = 0$, then $f$ is linear. Since $f$ is positive, it can be represented by a positive linear polynomial. It is straightforward to construct a tree of $\oplus$ gates that computes $f(\mathbf{x})$ without using constants.

Now let $\#_\wedge(\mathcal{C}) = k$, and suppose the lemma holds for all circuits with $k-1$ or fewer $\wedge$-gates. Let $\gamma$ be an initial $\wedge$-gate of $\mathcal{C}$, that is, an $\wedge$-gate which does not depend on any other $\wedge$-gate. We obtain two circuits $\mathcal{G}$ and $\mathcal{H}$ by "cutting" $\mathcal{C}$ at $\gamma$. $\mathcal{G}$ is the subgraph of $\mathcal{C}$ induced by $\gamma$ and all nodes of $\mathcal{C}$ upon which $\gamma$ depends. $\gamma$ is the only output node in $\mathcal{G}$. $\mathcal{H}$ is a copy of $\mathcal{C}$ except that the $\wedge$-gate $\gamma$ is changed to an input node $y$, and the incoming edges to $\gamma$ are deleted. The output node of $\mathcal{H}$ is the same as that of $\mathcal{C}$. Clearly, $\#_\wedge(\mathcal{G}) = 1$ and $\#_\wedge(\mathcal{H}) = k-1$. Furthermore, the output of $\mathcal{H}$ on $\mathbf{x}$ is the same as the output of $\mathcal{C}$ on $\mathbf{x}$ when the new input $y$ is supplied with the output of $\mathcal{G}$ on $\mathbf{x}$. Thus, $f_\mathcal{C}(\mathbf{x}) = f_\mathcal{H}(\mathbf{x}, f_\mathcal{G}(\mathbf{x}))$.

Let $\alpha_1$ and $\alpha_2$ be the nodes of $\mathcal{G}$ which feed the two inputs of $\wedge$-gate $\gamma$. We can write $f_{\alpha_i} = g_i + c_i$, $i = 1, 2$, where $g_i$ is positive linear and $c_i \in \{0, 1\}$. Then

$$f_\mathcal{G} = (g_1 + c_1)(g_2 + c_2) = g_1 g_2 + c_1 g_2 + c_2 g_1 + c_1 c_2 \tag{1}$$

We consider two cases, depending on whether $f_\mathcal{G}(\mathbf{0})$ is 0 or 1.

*Case 1:* $f_\mathcal{G}(\mathbf{0}) = 0$. Then $c_1 c_2 = 0$ and $f_\mathcal{G}$ is positive linear, so we can easily construct a constant-free circuit $\mathcal{G}'$ with $f_{\mathcal{G}'} = f_\mathcal{G}$ and $\#_\wedge(\mathcal{G}') = 1$. Also, $f_\mathcal{H}(\mathbf{0}, 0) = f_\mathcal{H}(\mathbf{0}, f_\mathcal{G}(\mathbf{0})) = f_\mathcal{C}(\mathbf{0}) = 0$ since $f_\mathcal{C}$ is positive, so $f_\mathcal{H}(\mathbf{x}, y)$ is also positive. By induction, there exists an equivalent constant-free circuit $\mathcal{H}'$ with $\#_\wedge(\mathcal{H}') \leq k-1$. Composing $\mathcal{G}'$ with $\mathcal{H}'$ yields the desired constant-free circuit for $f_\mathcal{C}(\mathbf{x})$.[1]

*Case 2:* $f_\mathcal{G}(\mathbf{0}) = 1$. Then $c_1 c_2 = 1$ and $f_\mathcal{G} + 1$ is positive linear, so we can easily construct a constant-free circuit $\mathcal{G}'$ with $f_{\mathcal{G}'} = f_\mathcal{G} + 1$ and $\#_\wedge(\mathcal{G}') = 1$. Also, $f_\mathcal{H}(\mathbf{0}, 1) = f_\mathcal{H}(\mathbf{0}, f_\mathcal{G}(\mathbf{0})) = f_\mathcal{C}(\mathbf{0}) = 0$ since $f_\mathcal{C}$ is positive, so $f_\mathcal{H}(\mathbf{x}, y+1)$ is also positive. By induction, there exists a constant-free circuit $\mathcal{H}'$ such that $f_{\mathcal{H}'}(\mathbf{x}, y) = f_\mathcal{H}(\mathbf{x}, y+1)$ and $\#_\wedge(\mathcal{H}') \leq k-1$. Composing $\mathcal{G}'$ with $\mathcal{H}'$ yields the desired constant-free circuit for $f_\mathcal{C}(\mathbf{x})$. □

We now define a special class of polynomials. We call the pair $(\{a, b, c\}, d)$ a *standard 3-form* if $a, b, c, d$ are positive linear polynomials, $a, b, c$ are pairwise disjoint, and at most one of $a, b, c$ is 0. We identify a standard 3-form with the polynomial $p = ab + ac + bc + d$. Note that $p$ is square-free and positive.

The correspondence between predicates with conjunctive complexity 1 and standard 3-form polynomials is given by the following lemmas.

**Lemma 3** *Let $p = ab + ac + bc + d$ be a standard 3-form. Then $C_\wedge(f_p) = 1$.*

*Proof.* Assume without loss of generality that $a \neq 0$ and $b \neq 0$. Over $GF_2$, we have $p \equiv (a+c)(b+c) + c + d$, so $C_\wedge(f_p) \leq 1$. Let $x \in \nu(a)$ and $y \in \nu(b)$. By the disjointness

[1] To compose $\mathcal{G}'$ and $\mathcal{H}'$, take their disjoint union, merge together corresponding input nodes, and add an edge from the output of $\mathcal{G}'$ to input $y$ of $\mathcal{H}'$.

conditions, $x \neq y$, and neither $x$ nor $y$ occurs in $c$. Now, let $p'$ result from $p$ by setting all other variables of $p$ to 0. Clearly, $p' = xy$, so $C_\wedge(f_{p'}) = 1$. Furthermore, $C_\wedge(f_{p'}) \leq C_\wedge(f_p)$ since any circuit for $f_p$ can be turned into a circuit for $f_{p'}$ without increasing the number of $\wedge$-gates. We conclude that $C_\wedge(f_p) = 1$ as claimed. □

**Lemma 4** *Every positive predicate $f$ with $C_\wedge(f) = 1$ can be represented by a standard 3-form $p$.*

*Proof.* Let $f$ be a positive predicate over $n$ variables with $C_\wedge(f) = 1$. By Lemma 2, there is a constant-free circuit $\mathcal{C}$ such that $f_\mathcal{C} = f$ and $\#_\wedge(\mathcal{C}) = 1$. Let $\gamma$ be the one $\wedge$-gate in $\mathcal{C}$. Because $\mathcal{C}$ is constant-free, the two inputs to $\gamma$ can be represented by positive linear polynomials $u$ and $v$ respectively. Also, the computation of the final output from the inputs $\mathbf{x}$ and the output of the $\wedge$-gate can be represented by a positive linear polynomial $w'$ over $\mathbf{x}$ and a new variable $y$. Clearly, $u \neq v$, $u \neq 0$, $v \neq 0$, and $y \in \nu(w')$ or else the $\wedge$-gate could be eliminated from $\mathcal{C}$. Hence, $w' = w + y$ for some positive linear polynomial $w$. Let $q = uv + w$. Then by construction, $f = f_q$.

Now, let $a, b, c$ be positive linear polynomials, where $\nu(a) = \nu(u) - \nu(v)$, $\nu(b) = \nu(v) - \nu(u)$, and $\nu(c) = \nu(u) \cap \nu(v)$. Thus, $u = a + c$ and $v = b + c$. Let $d = w + c$ and let $p = ab + ac + bc + d$. Then $p$ is a standard 3-form polynomial, and

$$q = uv + w = (a + c)(b + c) + (c + d) = ab + ac + bc + c^2 + c + d \equiv p.$$

Hence, $f = f_q = f_p$ as desired. □

**Lemma 5** *Let $(\{a, b, c\}, d)$ and $(\{a', b', c'\}, d')$ be distinct standard 3-forms. Then the polynomials $f = (ab + ac + bc) + d$ and $g = (a'b' + a'c' + b'c') + d'$ are distinct.*

*Proof.* Suppose, for a contradiction, that $f$ and $g$ are equal. Then $d = d'$ and $ab+ac+bc = a'b'+a'c'+b'c'$. Now, any homogeneous polynomial of total degree 2 is uniquely represented by a graph in which the nodes are the variables and there is an edge between $x$ and $y$ iff the term $xy$ is present in the polynomial. Since $a, b, c$ are pairwise disjoint, the graph for $ab+ac+bc$ is a complete tripartite graph (or complete bipartite if one of the linear terms is 0). The same holds for $a'b' + a'c' + b'c'$. Since the polynomials are the same, the graphs are identical. Since the graphs are complete tripartite (bipartite), the partition into three (two) independent sets is unique, i.e., $\{a, b, c\} = \{a', b', c'\}$. Thus $(\{a, b, c\}, d) = (\{a', b', c'\}, d')$, contradiction. □

From Lemmas 3–5, we immediately have:

**Theorem 6** *There is a one-to-one correspondence between standard 3-forms and predicates of conjunctive complexity 1.*

## 4 The number of predicates with conjunctive complexity 1

Theorem 6 allows us to count the predicates of conjunctive complexity 1 by counting the number of standard 3-forms:

**Theorem 7** *The number of positive predicates over $GF_2^n$ with conjunctive complexity 1 is $\binom{2^n}{3}$.*

*Proof.* Let $s$ be the number of distinct standard 3-forms $(\{a, b, c\}, d)$. We count the number of ways of picking $a, b, c, d$. There are $2^n$ ways of picking $d$. If $t$ is the number of ways of picking $a, b, c$, then $s = 2^n \cdot \frac{t}{3!}$.

To calculate $t$, we count the number of ways of assigning $n$ distinct variables to four sets $a, b, c, \lambda$ ($\lambda$ is the set of variables not in $a, b$, or $c$) in such a way that at most one of $a, b, c$ is empty. There are $4^n$ assignments in all. We must subtract from this the number of assignments in which two or more of $a, b, c$ are empty. Let $A$ be the set of assignments in which $a = \emptyset$. Similarly define $B$ and $C$. The set of interest is $S = (A \cap B) \cup (A \cap C) \cup (B \cap C)$. Inclusion-exclusion and the symmetry of $A, B, C$ gives $|S| = 3|A \cap B| - 2|A \cap B \cap C| = 3 \cdot 2^n - 2$ since $|A \cap B| = 2^n$ and $|A \cap B \cap C| = 1$. Thus, $t = 4^n - |S| = 4^n - 3 \cdot 2^n + 2 = (2^n - 1)(2^n - 2)$. Substituting back, we get

$$s = \frac{2^n(2^n - 1)(2^n - 2)}{3!} = \binom{2^n}{3}$$

as claimed. □

As mentioned earlier, the total number of predicates with conjunctive complexity 1 is twice the number of positive such predicates, yielding

**Theorem 8** *The number of predicates over $GF_2^n$ with conjunctive complexity 1 is $2\binom{2^n}{3}$.*

# 5 Counting predicates with higher conjunctive complexity

It might seem to the reader that the techniques used in counting the number of predicates with conjunctive complexity one should extend to the general case of conjunctive complexity $k$. The following observations, based on a group-theoretic characterization of the predicate space,[2] suggest this is likely to be hard.

The relation

$$f \sim g \quad \text{iff} \quad f \oplus g \text{ is linear}$$

is an equivalence relation on the space of predicates. The members of any equivalence class have the same conjunctive complexity. The set of equivalence classes is a group under the operation $\oplus$ defined as follows: Let $f_1, f_2$ be predicates in equivalence classes $C_1, C_2$, respectively. Let $C_3$ be the equivalence class of $f_1 \oplus f_2$, then we define $C_1 \oplus C_2 = C_3$. It is easy to verify that this operation is well-defined. The identity element of the group is the class of linear functions. Each equivalence class contains $2^{n+1}$ predicates. Since there are $2^{2^n}$ predicates altogether, the number of equivalence classes (i.e., the cardinality of the group) is $2^{2^n}/(2^{n+1}) = 2^m$, where $m = 2^n - n - 1$. Since each element is it's own inverse, and the group is finite Abelian, the group must be $(GF_2)^m$. Standard group theory then says that every subgroup must have cardinality a power of two.

Thus, our results imply there are $2\binom{2^n}{3}/(2^{n+1})$ equivalence classes with conjunctive complexity one. For $n = 3$, this evaluates to seven. Call two predicates *independent* if they belong to different equivalence classes (i.e., if their sum is non-linear). It can be verified that the predicates shown in Fig. 1 are pairwise independent. Clearly, they all have conjunctive complexity one.

One can also verify that these predicates, plus the $\mathbf{0}$ predicate, form a subgroup $S$ of the group of equivalence classes. This means one cannot, in the case $n = 3$, construct a

[2]We are not aware of this characterization having appeared in the literature.

$$x_1x_2 \qquad x_1x_3 \qquad x_2x_3 \qquad x_1(x_2{\oplus}x_3) \qquad x_2(x_1{\oplus}x_3)$$

$$x_3(x_1{\oplus}x_2) \qquad (x_1{\oplus}x_2)(x_1{\oplus}x_3)$$

Figure 1: Representatives of the seven complexity-1 equivalence classes on 3 inputs.

predicate with complexity two by taking the $\oplus$ of two predicates with complexity one. This is quite unexpected, and the same does not hold for the space of predicates over four bits. In that space, the number of equivalence classes with complexity one is $2^{\binom{2^4}{3}}/(2^{4+1}) = 35$. Since 36 is not a power of 2, the set of predicates with complexity $\leq 1$ is not closed under $\oplus$.

Returning to the case of $n = 3$, we note that $T = S{\oplus}\{x_1x_2x_3\}$ is a set of 8 pairwise independent predicates of conjunctive complexity more than one. Thus $T{+}S$ contain 16 pairwise independent predicates. Since the total number of equivalence classes is $2^{2^3}/(2^{3+1}) = 16$, $T$ and $S$ account for all equivalence classes. It is not hard to verify that all elements of $T$ have conjunctive complexity two. For example, $x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3$ turns out to be equal to $((x_1 + x_2)(x_1 + x_3) + x_1)(x_1 + x_2 + x_3 + 1)$. We conclude that all predicates on three inputs have conjunctive complexity two or less. More specifically there are 16 linear predicates, 112 predicates with complexity one, and 128 predicates with complexity two. Obtaining such a detailed map of the complexity of predicates on more than three bits is likely to be much more difficult.

## 6 Acknowledgments

Dan Sharfman coded an algorithm to count the number of predicates over $n$ variables of conjunctive complexity $k$. In practice, we can run this algorithm only on very small $n$ and $k$. We ran Sharfman's code for $k = 1$ and $2 \leq n \leq 5$. The experiment led to the hypothesis proven in this note. Conversations on this and related topics with Shigenori Uchiyama and Zoë Diamadi are also gratefully acknowledged.

## References

[1] J. Boyar, I. Damgård, and R. Peralta. Short non-interactive cryptographic proofs. *Journal of Cryptology*, 13:449–472, 2000.

[2] J. Boyar and R. Peralta. Short discreet proofs. In *Advances in Cryptology - Proceedings of EUROCRYPT 96*, volume 1070 of *Lecture Notes in Computer Science*, pages 131–142. Springer-Verlag, 1996.

[3] J. Boyar, R. Peralta, and D. Pochuev. On the multiplicative complexity of boolean functions over the basis $(\wedge, \oplus, 1)$. *Theoretical Computer Science*, 235:43–57, 2000.