The Complexity of Problems Defined by Subclasses of Boolean Functions

Steffen Reith Klaus W. Wagner
Lehrstuhl für Theoretische Informatik
Universität Würzburg
[streit,wagner]@informatik.uni-wuerzburg.de

January 8, 1999

Keywords: Computational complexity, boolean functions, Post, Boolean functions, dichotomy, superposition, satisfiability, tautology, counting problems.

Abstract

We study the complexities of boolean formulae based combinatorial problems like satisfiability, tautology, and some counting problems when restricted to the various closed sets of boolean functions. A set of boolean functions is called closed if it is closed under superposition (i.e. substitution, permutation of variables, identification of variables, and introduction of fictive variables). Already in the twenties E. L. POST gave a complete characterization of all closed sets of boolean functions which is heavily used in this paper. We give a complete solution to the initial question in the form of dichotomy theorems of the following kind: Given a finite set B of boolean functions. The satisfiability problem restricted to the boolean formulae which can be generated from the functions of B by superposition is easy to solve, if B is a subset of one of four specified closed sets of boolean functions. Otherwise it is NP-complete. There exists an algorithm which decides which of the cases takes place. Similar results are proved for the other problems. Related work has been done in [Sch78, CH96, CH97, Cre95, RV98, KST97] where only boolean formulae in conjunctive normal form were considered.

1 Introduction

In the twenties of this century E. L. POST proved very remarkable results on the structure of boolean functions (which were published only in 1942, see [Pos41]). For a set B of boolean functions let $[B]$ be the class of all boolean functions which can be generated from the functions of B by superposition (i.e. substitution, permutation of variables, identification of variables, and introduction of fictive variables). In other words, $[B]$ is the class of all boolean functions that can be computed by circuits with gates which realize functions from B. The set B is called a *base* of $[B]$, and a class F such that $[F] = F$ is called *closed*. In [Pos41] POST gave a complete classification of all closed classes of boolean functions (shown in Figure 1 with their inclusional relationships), and he presented for each of these closed classes a finite base. Moreover, there is an algorithm which, given a finite set $B \subseteq BF$, determines that closed class which coincides with $[B]$. Among the infinitely many closed classes F which are not equal to the class BF of all boolean functions there are five maximal classes which are called *Post classes*. An immediate consequence of POST's results is the following nice completeness criterion: For every set B of boolean functions, $[B] = BF$ if and only if B is not included in one of the five Post classes.

In this paper we study the complexity of boolean formulae based combinatorial problems restricted to the various closed classes of boolean functions. For a finite set B of boolean functions we define a set FORM(B) of *B-formulae* which describe exactly the functions from $[B]$. Let f_H be the function described by the B-formula H, and let $\#H =_{\mathrm{df}} \#\{x \mid f_H(x) = 1\}$. Now we are interested in the complexity of the following problems and function:

$$\begin{aligned}
\mathbf{SAT}(B) &=_{\mathrm{df}} \{H \mid H \in \text{FORM}(B) \text{ and there exists an } x \text{ such that } f_H(x) = 1\}\\
\mathbf{TAUT}(B) &=_{\mathrm{df}} \{H \mid H \in \text{FORM}(B) \text{ and } f_H(x) = 1 \text{ for all } x\}\\
\bigoplus(B) &=_{\mathrm{df}} \{H \mid H \in \text{FORM}(B) \text{ and } \#H \equiv 0(2)\}\\
\mathbf{P}(B) &=_{\mathrm{df}} \{(H,k) \mid H \in \text{FORM}(B),\, k \geq 0 \text{ and } \#H > k\}\\
\#(B) &=_{\mathrm{df}} \{\#H \mid H \in \text{FORM}(B)\}.
\end{aligned}$$

It is obvious that we have $\mathbf{SAT}(B) \in NP$, $\mathbf{TAUT}(B) \in$ *co-NP*, $\bigoplus(B) \in \oplus P$, $\mathbf{P}(B) \in PP$, and $\#(B) \in \#P$. On the other hand, it is well known (cf.[GJ79]) that for $B = \{\text{et}, \text{vel}, \text{non}\}$ these problems (this function) are complete for the classes NP, *co-NP*, $\oplus P$, PP, and $\#P$, resp., even for very restricted normal forms. What can we say about arbitrary bases B? In this paper we give a complete answer to this question by the following *dichotomy theorems*. Let M, L, D, R_0, R_1, E, and V be the classes of *monoton*, *linear*, *self-dual*, *0-reproducing*, *1-reproducing*, *et-*, and *vel*-functions (for a definition see Section 3). For any finite set B of boolean functions the following is true:

- If $[B] \subseteq D$ or $[B] \subseteq L$ or $[B] \subseteq E$ or $[B] \subseteq V$ then $\#(B)$ is polynomial time computable, and $\bigoplus(B)$ and $\mathbf{P}(B)$ are in P. Otherwise $\#(B)$ is complete for $\#P$, $\bigoplus(B)$ is complete for $\oplus P$, and $\mathbf{P}(B)$ is complete for PP .
- If $[B] \subseteq R_1$ or $[B] \subseteq M$ or $[B] \subseteq D$ or $[B] \subseteq L$ then $\mathbf{SAT}(B)$ is in P. Otherwise $\mathbf{SAT}(B)$ is complete for NP.
- If $[B] \subseteq R_0$ or $[B] \subseteq M$ or $[B] \subseteq D$ or $[B] \subseteq L$ then $\mathbf{TAUT}(B)$ is in P. Otherwise $\mathbf{TAUT}(B)$ is complete for *co-NP*.

For each of these statements there exists an algorithm which decides which of the cases takes place.

Finally let us mention the related work done previously in [Sch78, CH96, CH97, Cre95, RV98, KST97]. There the restricted case of formulae in conjunctive normal form is studied. All the restrictions made there concern only the boolean functions which build the clauses of the conjunctive normal forms.

The paper is organized as follows: In Section 2 we present POST's results on closed classes of boolean functions. In Section 3 the problems are defined whose complexity we study here. In Sections 4, 5, and 6 we develop our results on the complexity of $\#(B)$, $\bigoplus(B)$ and $\mathbf{P}(B)$, and $\mathbf{SAT}(B)$ and $\mathbf{TAUT}(B)$, resp.

2 Closed Classes of Boolean Functions

Let BF be the class of all boolean functions. The arity of boolean functions will sometimes be added as a superscript, i.e. f^n stands for $f : \{0,1\}^n \to \{0,1\}$. Let c_0 and c_1 be the 0-ary constant functions having value 0 and 1, resp. The functions id^1, non^1, et^2, vel^2, and aut^2 are defined by $\text{id}(a) = a$, $\text{non}(a) = 1 \leftrightarrow a = 0$, $\text{et}(a,b) = 1 \leftrightarrow a = b = 1$, $\text{vel}(a,b) = 0 \leftrightarrow a = b = 0$, and $\text{aut}(a,b) = 1 \leftrightarrow a \neq b$. We also write $\overline{x}$ instead of $\text{non}(x)$, $x \wedge y$ or $x \cdot y$ instead of $\text{et}(x,y)$, $x \vee y$ instead of $\text{vel}(x,y)$, and $x \oplus y$ instead of $\text{aut}(x,y)$. For $i \in \{1, 2, \ldots, n\}$, the i-th variable of the boolean function f^n is said to be *fictive* iff $f(a_1, \ldots a_{i-1}, 0, a_{i+1}, \ldots, a_n) = f(a_1, \ldots a_{i-1}, 1, a_{i+1}, \ldots, a_n)$ for all $a_1, \ldots a_{i-1}, a_{i+1}, \ldots, a_n \in \{0,1\}$.

For a class F of boolean functions let $[F]$ be the smallest class which contains F and which is closed under superposition (i.e. substitution, permutation of variables and identification of variables, introduction of fictive variables). Note that $[F]$ is exactly the class of those boolean functions which can be computed by boolean circuits whose gates realize boolean functions from F. For a boolean function f let $\langle f \rangle$ be the class of boolean functions which can be generated from f by introducing fictive variables. A set B of boolean functions is called a *base* of class F of boolean functions if $[B] = F$ and it is called *complete* if $[B] = BF$. A class F of boolean functions is called *closed* if $[F] = F$, and it is called a *Post class* if it is a maximal noncomplete closed class, i.e. if $[F] = F \subset BF$ and there is no closed G such that $F \subset G \subset BF$.

A boolean function f^n is said to be

- *a-reproducing* iff $f(a, a, \dots, a) = a$ ($a \in \{0, 1\}$),
- *linear* iff there exist $a_0, a_1, \dots, a_n \in \{0, 1\}$ such that $f(b_1, \dots, b_n) = c_{a_0} \oplus (c_{a_1} \cdot b_1) \oplus (c_{a_2} \cdot b_2) \oplus \dots \oplus (c_{a_n} \cdot b_n)$ for all $b_0, b_1, \dots, b_n \in \{0, 1\}$,
- *self-dual* iff $f(a_1, \dots, a_n) = \overline{f(\overline{a_1}, \dots \overline{a_n})}$ for all $a_1, \dots, a_n \in \{0, 1\}$,
- *monotonic* iff $f_m(a_1, \dots, a_n) \le f_m(b_1, \dots, b_n)$ for all $a_1, \dots, a_n, b_1, \dots, b_n \in \{0, 1\}$ such that $a_1 \le b_1, a_2 \le b_2, \dots, a_n \le b_n$,
- *a-separating* iff there exists an $i \in \{1, 2, \dots, n\}$ such that $f^{-1}(a) \subseteq \{0, 1\}^{i-1} \times \{a\} \times \{0, 1\}^{n-i}$ ($a \in \{0, 1\}$),
- *a-separating of degree* m iff for every $U \subseteq f^{-1}(a)$ such that $|U| = m$ there exists an $i \in \{1, 2, \dots, n\}$ such that $U \subseteq \{0, 1\}^{i-1} \times \{a\} \times \{0, 1\}^{n-i}$ ($a \in \{0, 1\}$, $m \ge 2$).

The classes of all 0-reproducing, 1-reproducing, linear, self-dual, monotonic, 0-separating, 1-separating, 0-separating of degree m, and 1-separating of degree m boolean functions are denoted by BF, R_0, R_1, L, D, M, S_0, S_1, S_0^m, and S_1^m, resp.

The closed classes of boolean functions were intensively studied by E. L. POST already at the beginning of the twenties. He gave a complete characterization of these classes. His main results are presented in the following two theorems. For a detailed presentation see also [JGK70].

Theorem 1 ([Pos41]) *1. The complete list of closed classes of boolean functions is as follows:*
BF, R_0, R_1, $R =_{\mathrm{df}} R_0 \cap R_1$,
M, $M_0 =_{\mathrm{df}} M \cap R_0$, $M_1 =_{\mathrm{df}} M \cap R_1$, $M_2 =_{\mathrm{df}} M \cap R$,
D, $D_1 =_{\mathrm{df}} D \cap R$, $D_2 =_{\mathrm{df}} D \cap M$,
L, $L_0 =_{\mathrm{df}} L \cap R_0$, $L_1 =_{\mathrm{df}} L \cap R_1$, $L_2 =_{\mathrm{df}} L \cap R$, $L_3 =_{\mathrm{df}} L \cap D$,
S_0, $S_{02} =_{\mathrm{df}} S_0 \cap R$, $S_{01} =_{\mathrm{df}} S_0 \cap M$, $S_{00} =_{\mathrm{df}} S_0 \cap R \cap M$,
S_1, $S_{12} =_{\mathrm{df}} S_1 \cap R$, $S_{11} =_{\mathrm{df}} S_1 \cap M$, $S_{10} =_{\mathrm{df}} S_1 \cap R \cap M$,
S_0^m, $S_{02}^m =_{\mathrm{df}} S_0^m \cap R$, $S_{01}^m =_{\mathrm{df}} S_0^m \cap M$, $S_{00}^m =_{\mathrm{df}} S_0^m \cap R \cap M$ *for* $m \ge 2$,
S_1^m, $S_{12}^m =_{\mathrm{df}} S_1^m \cap R$, $S_{11}^m =_{\mathrm{df}} S_1^m \cap M$, $S_{10}^m =_{\mathrm{df}} S_1^m \cap R \cap M$ *for* $m \ge 2$,
$E =_{\mathrm{df}} \langle et \rangle \cup \langle c_0 \rangle \cup \langle c_1 \rangle$, $E_0 =_{\mathrm{df}} \langle et \rangle \cup \langle c_0 \rangle$, $E_1 =_{\mathrm{df}} \langle et \rangle \cup \langle c_1 \rangle$, $E_2 =_{\mathrm{df}} \langle et \rangle$,
$V =_{\mathrm{df}} \langle vel \rangle \cup \langle c_0 \rangle \cup \langle c_1 \rangle$, $V_0 =_{\mathrm{df}} \langle vel \rangle \cup \langle c_0 \rangle$, $V_1 =_{\mathrm{df}} \langle vel \rangle \cup \langle c_1 \rangle$, $V_2 =_{\mathrm{df}} \langle vel \rangle$,
$O_1 =_{\mathrm{df}} \langle id \rangle$, $O_2 =_{\mathrm{df}} \langle c_1 \rangle$, $O_3 =_{\mathrm{df}} \langle c_0 \rangle$, $O_4 =_{\mathrm{df}} \langle id \rangle \cup \langle non \rangle$, $O_5 =_{\mathrm{df}} \langle id \rangle \cup \langle c_1 \rangle$, $O_6 =_{\mathrm{df}} \langle id \rangle \cup \langle c_0 \rangle$,
$O_7 =_{\mathrm{df}} \langle c_0 \rangle \cup \langle c_1 \rangle$, $O_8 =_{\mathrm{df}} \langle id \rangle \cup \langle c_0 \rangle \cup \langle c_1 \rangle$, $O_9 =_{\mathrm{df}} \langle id \rangle \cup \langle non \rangle \cup \langle c_0 \rangle \cup \langle c_1 \rangle$.

2. *The inclusional relationships between the closed classes of boolean functions are completely characterized by Figure 1.*

3. *There exists an algorithm which, given a finite set* $B \subseteq \mathrm{BF}$, *determines the closed class of boolean functions from the list above which coincides with* $[B]$.

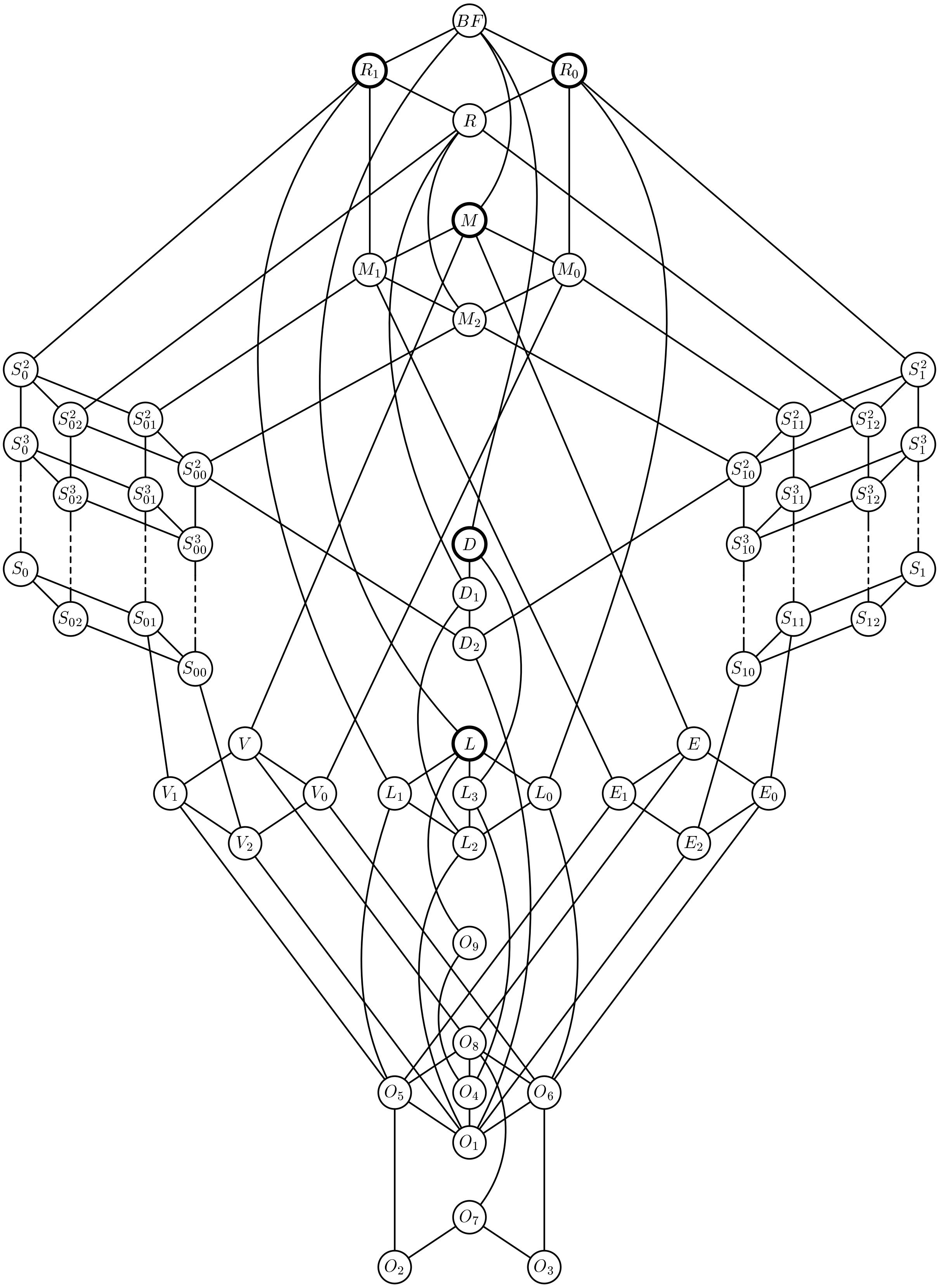

Figure 1: Graph of all classes of boolean function being closed under superposition

Let us consider an example. Define the boolean function f^3 such that $f(x, y, z) = 1$ iff exactly one argument is assigned by 1. Clearly, f is 0-reproducing but not 1-reproducing. Now assume that f is a linear function. So we know that f has a ring-sum expansion and $f(x, y, z) = c_0 \oplus (c_1 \cdot x) \oplus (c_2 \cdot y) \oplus (c_3 \cdot z)$. Since $f(0, 0, 0) = 0$ we know $c_0 = 0$ and it follows that $c_1 = c_2 = c_3 = 1$, because of $f(1, 0, 0) = f(0, 1, 0) = f(0, 0, 1) = 1$. But this is a contradiction because $f(1, 1, 1) = 0$, showing that f is not linear. Furthermore f is not monotonic since $f(1, 0, 0) = 1$ and $f(1, 1, 1) = 0$, and it is not selfdual because of $f(0, 0, 0) = f(1, 1, 1)$. Finally, f cannot be 1-separating of degree 2 because of $f(0, 0, 1) = f(0, 1, 0) = 1$ which shows that f is a base for R_0. Summarizing the above, f is not in R_1, L, M, D, and S_1^2 but it is in R_0. A short look at Figure 1 shows that $[\{f\}]$ must be equal to R_0.

Corollary 2 *1. There are five Post classes, namely R_0, R_1, L, D, and M.*

2. A class B of boolean functions is complete if and only if $B \not\subseteq R_0$, $B \not\subseteq R_1$, $B \not\subseteq L$, $B \not\subseteq D$, and $B \not\subseteq M$.

The following theorem gives us some information about bases of closed classes of boolean functions.

Theorem 3 ([Pos41]) *Every closed class of boolean functions has a finite base. In particular:*

1. *$\{et, vel, non\}$ is a base of BF.*
2. *$\{et, aut\}$ is a base of R_0.*
3. *$\{vel, x \oplus y \oplus 1\}$ is a base of R_1.*
4. *$\{et, vel, c_0, c_1\}$ is a base of M.*
5. *$\{(x \cdot \overline{y}) \vee (x \cdot \overline{z}) \vee (\overline{y} \cdot \overline{z})\}$ is a base of D.*
6. *$\{aut, c_1\}$ is a base of L.*
7. *$\{x \vee \overline{y}\}$ is a base of S_0.*
8. *$\{x \wedge \overline{y}\}$ is a base of S_1.*
9. *$\{x \vee (y \wedge \overline{z})\}$ is a base of S_{02}.*
10. *$\{x \wedge (y \vee \overline{z})\}$ is a base of S_{12}.*
11. *$\{x \vee (y \wedge z)\}$ is a base of S_{00}.*
12. *$\{x \wedge (y \vee z)\}$ is a base of S_{10}.*
13. *$\{et, c_0, c_1\}$ is a base of E.*
14. *$\{vel, c_0, c_1\}$ is a base of V.*

For a boolean function f define the boolean function $\mathrm{dual}(f)$ by $\mathrm{dual}(f)(x_1, \ldots, x_n) =_{\mathrm{df}} \overline{f(\overline{x_1}, \ldots, \overline{x_n})}$. The functions f and $\mathrm{dual}(f)$ are said to be *dual*. Obviously, $\mathrm{dual}(\mathrm{dual}(f)) = f$. Furthermore, f is self-dual iff $\mathrm{dual}(f) = f$. For a class F of boolean functions define $\mathrm{dual}(F) =_{\mathrm{df}} \{\mathrm{dual}(f) \mid f \in F\}$. The classes F and $\mathrm{dual}(F)$ are called *dual*.

Proposition 4 *1. If B is a set of boolean functions then $[\mathrm{dual}(B)] = \mathrm{dual}([B])$.*

2. Every closed class is dual to its "mirror class" (via the symmetry axis in Figure 1). In particular, $S_0 = \mathrm{dual}(S_1)$, $S_{02} = \mathrm{dual}(S_{12})$, $S_{01} = \mathrm{dual}(S_{11})$, and $S_{00} = \mathrm{dual}(S_{10})$.

3 Problems Defined by Closed Classes of Boolean Functions

In this paper we will restrict problems like satisfiability and counting problems, which are usually considered for the class of all boolean functions, to the various closed subclasses of the class of all boolean functions. Since we are interested in the complexity of these restricted problems, we have to speak about how we will present the boolean functions involved. There are different possibilities to present functions from $[B]$ where B is an arbitrary set of boolean functions. For example, such boolean functions can be presented by a circuit with B-gates, by a formula using symbols for functions from B, or by its truth table. In this paper we will use the second of these representations.

For every set B of boolean functions we will use a set FORM(B) of B-*formulae* which represent exactly the functions from $[B]$. We will not give a formal definition of B-formulae. Informally a B-formula H representing the boolean function $h(x_1, \ldots, x_n)$ describes (as usually done) the substitution structure using for every boolean function $f \in B$ a symbol $\mathbf{f}$. Additionally H gives information about where the variables $x_1, \ldots, x_n$ appear in H. The latter makes precise which permutation of the variables in H and which new fictive variables are used to represent h. In such a way every B-formula H describes a boolean function f_H. We obtain:

Proposition 5 *For every set B of boolean functions, $[B] = \{f_H \mid H \in FORM(B)\}$.*

By definition, for infinite sets $B \subseteq$ BH the set FORM(B) is a set of strings over an infinite alphabet which rises problems in the complexity considerations below. So we restrict ourselves to the case of finite sets B. Note that Theorem 3 ensures that every closed class has a finite base B.

Now we define some satisfiability, tautology, and counting problems restricted to arbitrary closed classes of boolean functions. For any formula H, let $\alpha(H)$ be the arity of f_H, and we define $\#H =_{\mathrm{df}} \#\{x \in \{0,1\}^{\alpha(H)} \mid f_H(x) = 1\}$.

$$
\begin{array}{rcl}
\mathbf{SAT}(B) & =_{\mathrm{df}} & \{H \mid H \in \text{FORM}(B) \text{ and there exists an } x \in \{0,1\}^{\alpha(H)} \text{ such that } f_H(x) = 1\} \\
\mathbf{TAUT}(B) & =_{\mathrm{df}} & \{H \mid H \in \text{FORM}(B) \text{ and } f_H(x) = 1 \text{ for all } x \in \{0,1\}^{\alpha(H)}\} \\
\bigoplus(B) & =_{\mathrm{df}} & \{H \mid H \in \text{FORM}(B) \text{ and } \#H \equiv 0(2)\} \\
\mathbf{P}(B) & =_{\mathrm{df}} & \{(H,k) \mid H \in \text{FORM}(B),\, k \geq 0 \text{ and } \#H > k\} \\
\#(B) & =_{\mathrm{df}} & \{\#H \mid H \in \text{FORM}(B)\}.
\end{array}
$$

Note that from the equality $[B_1] = [B_2]$ we cannot automatically conclude that the problems $\mathbf{SAT}(B_1)$ and $\mathbf{SAT}(B_2)$ have the same complexities. The same is true for the other problems and the #-function.

4 Counting Satisfying Assignments

In this section we study the complexity of the functions $\#(B)$. It is obvious that $\#(B) \in \#P$ for every finite set B of boolean functions. Furthermore $\#(\{\text{et}, \text{vel}, \text{non}\})$ is just the problem to compute the number of satisfying assignments to an ordinary (et, vel, non)-formula which is known to be complete for $\#P$. But what can we say about other bases B? We start with a statement on the number of satisfying assignments of dual functions.

Proposition 6

If f is an n-ary boolean function then $\#\mathrm{dual}(f) = 2^n - \#f$.

Lemma 7 *Let B be a finite subset of BF.*

1. *If $[B] \supseteq S_1$ then for every function $f \in \#P$ there exists a polynomial time computable function $h : \Sigma^* \mapsto FORM(B)$ such that $f(x) = \#h(x)$ for all $x \in \Sigma^*$.*
2. *If $[B] \supseteq S_{10}$ or $[B] \supseteq S_{00}$ then for every function $f \in \#P$ there exist polynomial time computable functions $h, h' : \Sigma^* \mapsto FORM(B)$ such that $f(x) = \#h(x) - \#h'(x)$ for all $x \in \Sigma^*$.*

Proof: It is well known (cf. [GJ79]) that counting the satisfying assignments of a 3-CNF formula (which is a special $(\text{et}, \text{vel}, \text{non})$-formula) is $\leq_m^p$-complete for $\#P$, i.e. for every function $f \in \#P$ there exists a polynomial time computable function g which maps into the set of all 3-CNF formulae such that $f(x) = \#g(x)$ for all $x \in \Sigma^*$. In all proofs below we can therefore start with a $\#H$ formula where H is a 3-CNF formula.

1. First note that the function $g(x,y) =_{\text{df}} \text{et}(x, \text{non}(y))$ is a base of S_1 and that $\text{et}(x,y) = g(x, g(x,y))$ and $\text{non}(x) = g(1,x)$. Because of $[B] \supseteq S_1$ there exist B-formulae E and N such that $E(x,y) = et(x,y)$ and $N(x,1) = \text{non}(x)$.

Let $H = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ be a 3-CNF-formula. Obviously H is satisfiable iff $(H \wedge u)$ is satisfiable, where any satisfying assignment of $H \wedge u$ ensures $u = 1$ for the new variable u. Because of $\text{vel} \in [\{\text{et}, \text{non}\}]$, any clause $C_i(x,y,z)$ can be transformed to a B-formula $H_i(x,y,z,u)$ such that $H_i(x,y,z,1) = C_i(x,y,z)$ for all $x,y,z \in \{0,1\}$, where the length of every $H_i(x,y,z,u)$ can be bounded by the same constant c.

Next we need a B-formula $F(u_1, \ldots, u_m, u)$ such that $F(u_1, \ldots, u_m, u) = u_1 \wedge \ldots \wedge u_m \wedge u$. Using the B-formula E such an F is easily constructed. However, the variables x and y can occurs several times in $E(x,y)$ which can result in an exponential blow-up of F. The trick is to introduce in $u_1 \wedge \ldots \wedge u_m \wedge u$ parentheses in such a way that we get a tree of depth $log(m+1)$. Now a blow-up which is exponential in $log(m+1)$ results in a B-formula F whose length is polynomially bounded in the length of H. Finally we obtain $\#F(H_1, H_2, \ldots, H_m, u) = \#H$ where $\#F(H_1, H_2, \ldots, H_m, u)$ is a B-formula.

2. S_{10}: First note that the function $g(x,y,z) =_{\text{df}} \text{et}(x, \text{vel}(y,z))$ is a base of S_{10} and that $\text{et} \in S_{10}$ since $\text{et}(x,y) = g(x,y,y)$ and $\text{vel}(x,y) = g(1,x,y)$. Because of $[B] \supseteq S_{10}$ there exist B-formulae E and V such that $E(x,y) = et(x,y)$ and $V(x,y,1) = \text{vel}(x,y)$.

Let $H(x_1, \ldots, x_n) = C_1 \wedge C_2 \wedge \ldots \wedge C_m$ be a 3-CNF-formula. Choose new variables $y_1, \ldots, y_n$ and let $H_1(x_1, \ldots, x_n, y_1, \ldots, y_n)$ be that formula which build from H when every $\overline{x_i}$ is replaced with y_i. Defining $H_2 =_{\text{df}} \bigwedge_{i=1}^n (\overline{x_i} \leftrightarrow y_i) \wedge H_1$ we obtain $H_2 = \overline{H_3} \wedge H_4$ where $H_3 =_{\text{df}} \bigvee_{i=1}^n (x_i \wedge y_i)$ and $H_4 =_{\text{df}} \bigwedge_{i=1}^n (x_i \vee y_i) \wedge H_1$. We conclude $\#H = \#H_2 = \#(\overline{H_3} \wedge H_4) = \#H_4 - \#(H_3 \wedge H_4)$, since $\#H_4 = \#(H_3 \wedge H_4) + \#(\overline{H_3} \wedge H_4)$.

Now we proceed similar to the proof of Statement 1. We define $H_5 =_{\text{df}} (H_4 \wedge u)$ and $H_6 =_{\text{df}} (H_3 \wedge H_4 \wedge u)$, and we obtain $\#H = \#H_5 - \#H_6$. Note that every satisfying assignment of H_5 or H_6 ensures $u = 1$ and that in the formulae H_5 and H_6 only the boolean functions et and vel occur (i.e. no negation). Using the B-formulae E and $V(\cdot,\cdot,u)$ we can construct a B-formulae G_1 and G_2 such that $\#G_1 = \#H_5$ and $\#G_2 = \#H_6$.

S_{00}: Since $[B] \supseteq S_{00}$ and Proposition 4 we obtain $[\text{dual}(B)] = \text{dual}([B]) \supseteq \text{dual}(S_{00}) = S_{10}$. By the proof of this Statement for S_{10}, for every 3-CNF formula H there exist two $\text{dual}(B)$-formulae G_1 and G_2 of arity $2n+1$ such that $\#H = \#G_1 - \#G_2$. Replacing in G_1 (G_2) every occurrence of a function $f \in \text{dual}(B)$ with the function $\text{dual}(f) \in B$ we obtain a B-formula G_3 (G_4) such that $f_{G_3} = \text{dual}(f_{G_1})$ and $f_{G_4} = \text{dual}(f_{G_2})$. By Proposition 6 we obtain $\#G_3 = 2^{2n+1} - \#G_1$ and $\#G_4 = 2^{2n+1} - \#G_2$ and hence $\#H = \#G_3 - \#G_4$. ❏

By this lemma we get immediately the following theorem. For functions f, f' we write $f \leq_m^p f'$ if there exists a polynomial time computable function g such that $f(x) = f'(g(x))$ for all x. For $k \geq 1$ we write $f \leq_{||k}^p f'$ if there exists a polynomial time algorithm which, for all x, computes $f(x)$ making at most k parallel queries for values of f'. Furthermore, for a function f (a set $\mathcal{F}$ of functions) let $P^f[O(1)]$ ($P^{\mathcal{F}}[O(1)]$, resp.) be the class of sets which can be decided by a polynomial time algorithm making , for every x, a constant number of queries for values of f (of a function f from $\mathcal{F}$, resp.).

Theorem 8 *Let B be a finite set of boolean functions.*

1. *If $[B] \supseteq S_1$ then $\#(B)$ is $\leq_m^p$-complete for $\#P$.*

2. *If $[B] \supseteq S_{00}$ or $[B] \supseteq S_{10}$ then #(B) is $\leq^p_{||2}$-complete for #P and hence $P^{\#P}[O(1)] = P^{\#(B)}[O(1)]$.*

Theorem 9 *For a finite set $B \subseteq BF$, if $[B] \subseteq D$ or $[B] \subseteq L$ or $[B] \subseteq E$ or $[B] \subseteq V$ then #(B) is polynomial time computable.*

Proof: A self-dual n-ary boolean function has obviously 2^{n-1} satisfying assignments.

A linear function $f(x_1, \dots, x_n)$ having exactly m nonzero coefficients in its ring-sum expansion has exactly 2^{m-1} satisfying assignments. This coefficient of x_i is equal to 0 if and only if $f(0, \dots, 0, 0, 0, \dots, 0) = f(0, \dots, 0, 1, 0, \dots, 0)$ where the 1 is at the i-th place in f.

For a function $f(x_1, \dots, x_n) \in E$ we have either $f(x_1, \dots, x_n) = 0$ or there exists a set $I \subseteq \{1, \dots, n\}$ such that $f(x_1, \dots, x_n) = \bigwedge_{i \in I} x_i$ where $i \in I$ if and only if $f(1, \dots, 1, 0, 1, \dots, 1) = 0$. The number of satisfying assignments is 0 in the first case and $2^{n-|I|}$ in the second case.

For a function $f(x_1, \dots, x_n) \in V$ we have either $f(x_1, \dots, x_n) = 1$ or there exists a set $I \subseteq \{1, \dots, n\}$ such that $f(x_1, \dots, x_n) = \bigvee_{i \in I} x_i$ where $i \in I$ if and only if $f(0, \dots, 0, 1, 0, \dots, 0) = 1$. The number of satisfying assignments is 2^n in the first case and $2^n - 2^{n-|I|}$ in the second case.

❑

Using Figure 1 it is easy to see that the condition "$[B] \not\supseteq S_{00}$ and $[B] \not\supseteq S_{10}$" is equivalent to the condition "$[B] \subseteq D$ or $[B] \subseteq L$ or $[B] \subseteq E$ or $[B] \subseteq V$". Hence we can summarize the preceding two theorems as follows:

Theorem 10 *(Dichotomy Theorem) For a finite set $B \subseteq BF$, if $[B] \subseteq D$ or $[B] \subseteq L$ or $[B] \subseteq E$ or $[B] \subseteq V$ then #(B) is polynomial time computable. Otherwise #(B) is $\leq^p_{||2}$-complete for #P. There exists an algorithm which decides which of these two cases takes place.*

5 Counting Problems

In this section we study the complexity of the problems $\bigoplus(B)$ and $\mathbf{P}(B)$. As in the case of #(B) it is obvious that $\bigoplus(B) \in \oplus P$ and $\mathbf{P}(B) \in PP$ for every finite set B of boolean functions and that $\bigoplus(\{\text{et}, \text{vel}, \text{non}\})$ and $\mathbf{P}(\{\text{et}, \text{vel}, \text{non}\})$ are complete for $\oplus P$ and PP, resp. Now we consider the case of an arbitrary base B. For sets L' and L we write $L' \leq^p_m L$ if there exists a polynomial time computable function g such that $x \in L' \leftrightarrow g(x) \in L$ for all x. For $k \geq 1$, we write $L' \leq^p_{k-tt} L$ if there exists a polynomial time algorithm which decides L' making for every x at most k parallel queries to the oracle L. For $k \geq 1$ and a set L (a class $\mathcal{K}$ of sets), let $P^L_{||}[k]$ ($P^{\mathcal{K}}_{||}[k]$, resp.) be the class of all sets L' such that $L' \leq^p_{k-tt} L$ ($L' \leq^p_{k-tt} L$ for some $L \in \mathcal{K}$, resp.). Furthermore, for a set L (a class $\mathcal{K}$ of sets), let P^L ($P^{\mathcal{K}}$, resp.) be the class of all sets which can be decided by an polynomial time algorithm using the oracle L (an oracle $L \in \mathcal{K}$, resp.).

Theorem 11 *Let B be a finite set of boolean functions.*

1. *If $[B] \supseteq S_1$ then $\bigoplus(B)$ and $\mathbf{P}(B)$ are $\leq^p_m$-complete for $\oplus P$ and PP, resp.*

2. *If $[B] \supseteq S_{00}$ or $[B] \supseteq S_{10}$ then $\bigoplus(B)$ is $\leq^p_{2-tt}$-complete for $\oplus P$, and hence $\oplus P = P^{\bigoplus(B)}_{||}[2]$.*

3. *If $[B] \supseteq S_{00}$ or $[B] \supseteq S_{10}$ then $\mathbf{P}(B)$ is $\leq^p_T$-complete for PP, and hence $P^{PP} = P^{\mathbf{P}(B)}$.*

Proof: 1. This is an immediate consequence of Lemma 7.1.

2. A language L is in $\oplus P$ iff there exists a function $f \in \#P$ such that $x \in L \leftrightarrow f(x) \equiv 0(2)$. By Lemma 7.2 we get functions $h_1, h_2 \in FP$ producing B-formulae such that $f(x) = \#h_1(x) - \#h_2(x)$. Now a polynomial time algorithm with oracle $\bigoplus(B)$ can decide whether $f(x) \equiv 0(2)$ by querying $h_1(x)$

and $h_2(x)$ to the oracle. From this we can conclude $\oplus P \subseteq P_{||}^{\oplus(B)}[2] \subseteq P_{||}^{\oplus P}[2]$. However, it is known that $P_{||}^{\oplus P}[2] = \oplus P$ (see [PZ83]).

3. A language L is in PP iff there exist functions $f \in \#P$ and $g \in FP$ such that $x \in L \leftrightarrow f(x) \geq g(x)$. By Lemma 7.2 we get functions $h_1, h_2 \in FP$ producing B-formulae such that $f(x) = \#h_1(x) - \#h_2(x)$. Now a polynomial time algorithm with oracle $\mathbf{P}(B)$ can decide $x \in L$ as follows: a) Compute $\#h_1(x)$ and $\#h_2(x)$ with a binary search using oracle queries of the form "$\#h_i(x) \geq k$". b) Compute $g(x)$ and check whether $\#h_1(x) - \#h_2(x) \geq g(x)$. ❑

Theorem 12 *For a finite set B of boolean functions, if $[B] \subseteq D$ or $[B] \subseteq L$ or $[B] \subseteq E$ or $[B] \subseteq V$ then $\mathbf{MOD}_k(B)$ and $\mathbf{P}(B)$ are in P.*

Proof: The theorem follows directly by the fact that, by Theorem 9, the corresponding counting function $\#(B)$ can be computed in polynomial time. ❑

As in the previous section we can summarize the preceding two theorems as follows:

Theorem 13 *(Dichotomy Theorem) Let B be a finite set of boolean functions.*

1. *For $k \geq 2$, if $[B] \subseteq D$ or $[B] \subseteq L$ or $[B] \subseteq E$ or $[B] \subseteq V$ then $\oplus(B)$ is in P. Otherwise $\oplus(B)$ is $\leq^p_{2-tt}$-complete for $\oplus P$. There exists an algorithm which decides which of these two cases takes place.*

2. *If $[B] \subseteq D$ or $[B] \subseteq L$ or $[B] \subseteq E$ or $[B] \subseteq V$ then $\mathbf{P}(B)$ is in P. Otherwise $\mathbf{P}(B)$ is $\leq^p_T$-complete for PP. There exists an algorithm which decides which of these two cases takes place.*

6 Satisfiability and Tautology

In this section we study the complexity of the problems $\mathbf{SAT}(B)$ and $\mathbf{TAUT}(B)$. As in the case of $\#(B)$ it is obvious that $\mathbf{SAT}(B) \in NP$ and $\mathbf{TAUT}(B) \in co\text{-}NP$ for every finite set B of boolean functions and that $\mathbf{SAT}(\{\mathrm{et}, \mathrm{vel}, \mathrm{non}\})$ and $\mathbf{TAUT}(\{\mathrm{et}, \mathrm{vel}, \mathrm{non}\})$ are complete for NP and $co\text{-}NP$, resp. Now we consider the case of an arbitrary base B.

Theorem 14 *Let B be a finite set of boolean functions.*

1. *If $[B] \supseteq S_1$ then $\mathbf{SAT}(B)$ is $\leq^p_m$-complete for NP.*

2. *If $[B] \supseteq S_0$ then $\mathbf{TAUT}(B)$ is $\leq^p_m$-complete for $co\text{-}NP$.*

Proof: 1. This is an immediate consequence of Lemma 7.1.

2. By $[B] \supseteq S_0$ and Proposition 4 we obtain $[\mathrm{dual}(B)] = \mathrm{dual}([B]) \supseteq \mathrm{dual}(S_0) = S_1$. Let $L \in co\text{-}NP$. By Statement 1 we get a function $h \in FP$ producing $\mathrm{dual}(B)$-formulae such that $x \in L \leftrightarrow h(x) \notin \mathbf{SAT}(\mathrm{dual}(B)) \leftrightarrow \#h(x) = 0$. Let $h'(x)$ be that B-formula which is obtained from the $\mathrm{dual}(B)$-formula $h(x)$ by replacing every occurrence of a function $f \in \mathrm{dual}(B)$ with the function $\mathrm{dual}(f) \in B$. Obviously, $h' \in FP$ and $h'(x) = \mathrm{dual}(h(x))$. By Proposition 6 we obtain $\#h'(x) = 2^{\alpha(h(x))} - \#h(x)$ and consequently $x \in L \leftrightarrow \#h(x) = 0 \leftrightarrow \#h'(x) = 2^{\alpha(h(x))} \leftrightarrow h'(x) \in \mathbf{TAUT}(B)$. ❑

Theorem 15 *Let B be a finite set of boolean functions.*

1. *If $[B] \subseteq R_1$ or $[B] \subseteq M$ or $[B] \subseteq D$ or $[B] \subseteq L$ then $\mathbf{SAT}(B)$ is in P.*

2. *If $[B] \subseteq R_0$ or $[B] \subseteq M$ or $[B] \subseteq D$ or $[B] \subseteq L$ then $\mathbf{TAUT}(B)$ is in P.*

Proof: If $[B] \subseteq R_1$ then every B-formula H is in $\mathbf{SAT}(B)$ because of $H(1,1,\ldots,1) = 1$.

If $[B] \subseteq R_0$ then no B-formula H is in $\mathbf{TAUT}(B)$ because of $H(0,0,\ldots,0) = 0$.

If $[B] \subseteq M$ then $\mathbf{SAT}(B)$ and $\mathbf{TAUT}(B)$ are in P because the following is true for B-formulae H: $H \in \mathbf{SAT}(B) \leftrightarrow H(1,1,\ldots,1) = 1$ and $H \in \mathbf{TAUT}(B) \leftrightarrow H(0,0,\ldots,0) = 1$

If $[B] \subseteq D$ or $[B] \subseteq L$ then $\mathbf{SAT}(B)$ and $\mathbf{TAUT}(B)$ are in P because of the fact that, by Theorem 9, the corresponding counting function $\#(B)$ can be computed in polynomial time. ❑

Using Figure 1 it is easy to see that the condition "$[B] \not\supseteq S_1$" is equivalent to the condition "$[B] \subseteq R_1$ or $[B] \subseteq M$ or $[B] \subseteq D$ or $[B] \subseteq L$" and that the condition "$[B] \not\supseteq S_0$" is equivalent to the condition "$[B] \subseteq R_0$ or $[B] \subseteq M$ or $[B] \subseteq D$ or $[B] \subseteq L$". Hence we can summarize the preceding two theorems as follows:

Theorem 16 *(Dichotomy Theorem) Let B be a finite set of boolean functions.*

1. *If $[B] \subseteq R_1$ or $[B] \subseteq M$ or $[B] \subseteq D$ or $[B] \subseteq L$ then $\mathbf{SAT}(B)$ is in P. Otherwise $\mathbf{SAT}(B)$ is $\leq_m^p$-complete for NP. There exists an algorithm which decides which of these two cases takes place.*

2. *If $[B] \subseteq R_0$ or $[B] \subseteq M$ or $[B] \subseteq D$ or $[B] \subseteq L$ then $\mathbf{TAUT}(B)$ is in P. Otherwise $\mathbf{TAUT}(B)$ is $\leq_m^p$-complete for co-NP. There exists an algorithm which decides which of these two cases takes place.*

Recall the example after Theorem 1 where we got $[\{f\}] = R_0$ for the exact-one-of-three function f. Theorem 16 now yields that $\mathbf{SAT}(\{f\})$ is $\leq_m^p$-complete for NP, but $\mathbf{TAUT}(\{f\})$ is in P.

References

[CH96] N. Creignou and M. Hermann. Complexity of generalized satisfiability counting problems. *Information and Computation*, 125:1–12, 1996.

[CH97] N. Creignou and J.-J. Hebrard. On generating all solutions of generalized satisfiability problems, 1997.

[Cre95] N. Creignou. A dichotomy theorem for maximum generalized satisfiability problems. *Journal of Computer and System Sciences*, 51:511–522, 1995.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.

[JGK70] S. W. Jablonski, G. P. Gawrilow, and W. B. Kudrajawzew. *Boolesche Funktionen und Postsche Klassen*. Akademie-Verlag, 1970.

[KST97] S. Khanna, M. Sudan, and L. Trevisan. Constraint satisfaction: The approximability of minimization problems. In *Proceedings 12th Computational Complexity Conference*, pages 282–296. IEEE Computer Society Press, 1997.

[Pos41] E. L. Post. Two-value iterative systems, 1941.

[PZ83] C. H. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings 6th GI Conference on Theoretical Computer Science*, volume 145 of *Lecture Notes in Computer Science*, pages 269–275. Springer Verlag, 1983.

[RV98] S. Reith and H. Vollmer. The complexity of computing optimal assignments of generalized propositional formulae. ECCC Report TR98-022, 1998.

[Sch78] T. J. Schaefer. The complexity of satisfiability problems. In *Proccedings 10th STOC, San Diego (CA, USA)*, pages 216–226. Association for Computing Machinery, 1978.