The Complexity of Problems Defined by Boolean Circuits

- Preliminary Version -

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

21st March 2000

Keywords: Computational complexity, boolean circuits, circuit value problem, satisfiability, tautology, counting functions, threshold problem.

Abstract

We study the complexity of circuit-based combinatorial problems (e.g., the circuit value problem and the satisfiability problem) defined by boolean circuits with gates from an arbitrary finite base B of boolean functions. Special cases have been investigated in the literature. We give a complete characterization of their complexity depending on the base B. For example, for the satisfiability problem for boolean circuits with gates from B we present a complete collection of (decidable) criteria which tell us for which B this problem is in L, is complete for NL, is complete for $\oplus$L, is complete for P, or is complete for NP. Our proofs make substantial use of the characterization of all closed classes of boolean functions given by E.L. POST already in the twenties.

1 Introduction

The complexity of formula-based and circuit-based combinatorial problems was studied through the more than three decades of Complexity Theory. Already in 1971, S.A. COOK [Coo71] proved that the satisfiability problem for boolean formulae is NP-complete: This was the first NP-complete problem ever discovered. R.E. LADNER [Lad77] proved in 1977 that the circuit value problem is P-complete. In many cases when a new complexity class was introduced and investigated, a formula-based or circuit-based combinatorial problem was the first which was proved to be complete for this class (see [SM73, Gil77], for example). However, usually these problems were defined using formulae or circuits with a *complete base* of boolean operations or gates, mostly with the base $\{\wedge, \vee, \neg\}$. But what about the complexity of such problems when a different base is used? There are several special results of this kind (e.g. [Sim75, Gol77, Lew78, GP86]). In particular, there are very detailed investigations for the special case of boolean formulae in conjunctive normal form in [Sch78, Cre95, CH96, CH97, KST97, KSW97, RV98]. But there are no results answering this question for unresricted circuits in full generality. In this paper we will give complete characterizations of the complexity of some combinatorial problems defined by circuits with boolean gates from an arbitrary finite base.

The paper is organized as follows. In Section 2 we briefly introduce some notation and results needed from Complexity Theory. In Section 3 we define B-circuits as boolean circuits with gates from a finite set B of boolean functions, and we define $\mathrm{BF}[B]$ as the class of boolean functions computed by B-circuits.

The complexity of a problem defined by B-circuits only depends on $\mathrm{BF}[B]$. The classes of the form $\mathrm{BF}[B]$ are exactly those classes of boolean functions which contain the identity function and which are closed under superposition (i.e., substitution, permutation of variables, identification of variables, and introduction of fictive variables). Already in the twenties, E.L. POST [Pos41] gave a complete characterization of these classes. We make substantial use of his results which we present in Section 4. In Sections 5-9 we study the complexity of the circuit value problem, the satisfiability problem and the tautology problem, some quantified circuit problems, the counting function, and the threshold problem, resp., when defined by circuits with boolean gates from an arbitrary finite base B. We give complete characterizations of their complexity as in the following example. Let the satisfiability problem $\mathbf{SAT}(B)$ be the set of all B-circuits which evaluate to 1 for at least one input tuple. If B consists only of 1-reproducing functions, of self-dual functions, or of non-functions (for definitions see Section 4), then $\mathbf{SAT}(B)$ is in L. Otherwise, if B consists only of and-functions or of or-functions then $\mathbf{SAT}(B)$ is $\leq_{\mathrm{m}}^{\mathrm{log}}$-complete for NL. Otherwise, if B consists only of linear functions then $\mathbf{SAT}(B)$ is $\leq_{\mathrm{m}}^{\mathrm{log}}$-complete for $\oplus$L. Otherwise, if $\mathbf{SAT}(B)$ consists only of monotone functions then $\mathbf{SAT}(B)$ is $\leq_{\mathrm{m}}^{\mathrm{log}}$-complete for P. Otherwise $\mathbf{SAT}(B)$ is $\leq_{\mathrm{m}}^{\mathrm{log}}$-complete for NP.

2 Complexity Theoretic Preliminaries

In this section we will informally introduce some standard notation and results from complexity theory. For more details see [BDG90a, BDG90b, Pap94].

The complexity classes P, NP, L, NL, and PSPACE are defined as the classes of languages (problems) which can be accepted by deterministic polynomially time bounded Turing machines, nondeterministic polynomially time bounded Turing machines, deterministic logarithmically space bounded Turing machines, nondeterministic logarithmically space bounded Turing machines, and deterministic polynomially space bounded Turing machines, resp. Let PP be the class of languages which are accepted by a nondeterministic polynomially time bounded Turing machine M in the following manner: An input x is accepted by M iff more than half of M's computation paths on input x are accepting paths. Let $\oplus$L be the class of languages which are accepted by a nondeterministic logarithmically space bounded Turing machine M in the following manner: An input x is accepted by M iff the number of M's accepting paths on input x is odd.

For a class $\mathcal{K}$ of languages let $\text{co-}\mathcal{K} =_{\mathrm{df}} \{\overline{\mathrm{A}} \mid \mathrm{A} \in \mathcal{K}\}$. Furthermore let $\mathrm{P}^{\mathcal{K}}$, $\mathrm{NP}^{\mathcal{K}}$, $\mathrm{L}^{\mathcal{K}}$, and $\mathrm{NL}^{\mathcal{K}}$ be the classes of languages accepted by the type of machines used for the classes P, NP, L, and NL, resp., but which have the additional ability to ask free-of-cost queries to languages from $\mathcal{K}$ (such machines are called *oracle machines*). Define the classes of the polynomial time hierarchy by $\Sigma_1^{\mathrm{p}} =_{\mathrm{df}} \mathrm{NP}$, $\Sigma_{\mathrm{k}+1}^{\mathrm{p}} =_{\mathrm{df}} \mathrm{NP}^{\Sigma_{\mathrm{k}}^{\mathrm{p}}}$, and $\Pi_{\mathrm{k}}^{\mathrm{p}} =_{\mathrm{df}} \text{co-}\Sigma_{\mathrm{k}}^{\mathrm{p}}$ for $k \geq 1$.

Let FP be the class of functions which can be computed by a deterministic polynomially time bounded Turing machine. For a nondeterministic polynomially time bounded machine M let $\#M(x)$ be the number of accepting paths of M on input x. Let #P be the class of all such functions $\#M$. For a class $\mathcal{K}$ of sets and a function $s : \mathbb{N} \to \mathbb{N}$, let $\mathrm{FL}_{||}^{\mathcal{K}}[s]$ be the class of all functions which are computable by a logspace bounded Turing machine which, for inputs of length n, can make at most $s(n)$ parallel queries (i.e., the list of all queries is formed before any of them is asked) to a language from $\mathcal{K}$.

Here are some basic facts on the complexity classes defined above.

Theorem 1 *1.* $\mathrm{L} \subseteq \mathrm{NL} \cap \oplus\mathrm{L}$ *and* $\mathrm{NL} \cup \oplus\mathrm{L} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE}$.

2. All languages in $\mathrm{NL} \cup \oplus\mathrm{L}$ *can be accepted by deterministic* $\log^2 n$ *space bounded Turing machines.*

3. The classes L, NL, $\oplus$L, P, PP, *and* PSPACE *are closed under complementation.*

4. $\mathrm{L}^{\mathrm{NL}} = \mathrm{NL}$ *and* $\mathrm{L}^{\oplus\mathrm{L}} = \oplus\mathrm{L}$.
5. $\Sigma^{\mathrm{p}}_{\mathrm{k}} \cup \Pi^{\mathrm{p}}_{\mathrm{k}} \subseteq \Sigma^{\mathrm{p}}_{\mathrm{k+1}} \cap \Pi^{\mathrm{p}}_{\mathrm{k+1}}$ *for* $k \geq 1$ *and* $\bigcup_{k\geq 1}(\Sigma^{\mathrm{p}}_{\mathrm{k}} \cup \Pi^{\mathrm{p}}_{\mathrm{k}}) \subseteq \mathrm{PSPACE}$.
6. $\mathrm{FP} \subseteq \#\mathrm{P}$.
7. *For every* $f \in \#\mathrm{P}$ *there exist* $f' \in \#\mathrm{P}$ *and a logspace computable function* g *such that* $f(x) = g(x) - f'(x)$ *for all* x.

To compare the complexity of languages or functions we use reducibilities. For languages A and B we write $A \leq^{\log}_{\mathrm{m}} B$ if there exists a logspace computable function g such that $x \in A \Leftrightarrow g(x) \in B$ for all x. For functions f and f' we write $f \leq^{\log}_{\mathrm{m}} f'$ if there exists a logspace computable function g such that $f(x) = f'(g(x))$ for all x. We write $f \leq^{\log}_{1-\mathrm{T}} f'$ if there exist logspace computable functions g_1 and g_2 such that $f(x) = g_1(x, f'(g_2(x)))$. Obviously, $f \leq^{\log}_{\mathrm{m}} f'$ implies $f \leq^{\log}_{1-\mathrm{T}} f'$. These reductions are reflexive and transitive relations. A language A is called $\leq^{\log}_{\mathrm{m}}$*-hard* ($\leq^{\log}_{\mathrm{m}}$*-complete*) for a class $\mathcal{K}$ of languages iff $B \leq^{\log}_{\mathrm{m}} A$ for every $B \in \mathcal{K}$ ($A \in \mathcal{K}$ and $B \leq^{\log}_{\mathrm{m}} A$ for every $B \in \mathcal{K}$, resp.). In the same way one defines hardness und completeness for the reducibilities $\leq^{\log}_{\mathrm{m}}$and $\leq^{\log}_{1-\mathrm{T}}$ for functions.

In this paper we need the following problems which are complete for the classes NL and $\oplus$L, resp.

Graph Accessibility Problem GAP:
Input: A directed acyclic graph G whose vertices have outdegree 0 or 2,
a start vertex s, and a target vertex t.
Question: Is there a path in G which leads from s to t?

Graph Odd Accessibility Problem GOAP:
Input: A directed acyclic graph G whose vertices have outdegree 0 or 2,
a start vertex s, and a target vertex t.
Question: Is the number of paths in G, which lead from s to t, odd?

Theorem 2 1. GAP *and* $\overline{\mathrm{GAP}}$ *are* $\leq^{\log}_{\mathrm{m}}$*-complete for* NL.

2. GOAP *and* $\overline{\mathrm{GOAP}}$ *are* $\leq^{\log}_{\mathrm{m}}$*-complete for* $\oplus$L.

3 Problems Defined by B-Circuits

In this paper we will study problems which are defined by boolean circuits with gates from a finite set of boolean functions. Informally, for a finite set B of boolean functions, a *B-circuit C with input variables* $x_1, \ldots, x_{\alpha(C)}$ is a directed acyclic graph with a special output node which has the following properties: Every vertex (gate) with indegree 0 is labeled with an x_i or a 0-ary function from B. Every vertex (gate) with indegree $k > 0$ is labeled with a k-ary function from B. Given values $a_1, \ldots, a_{\alpha(C)} \in \{0, 1\}$ to $x_1, \ldots, x_{\alpha(C)}$, every gate computes a boolean value by applying the boolean function of this gate to the values of the incoming edges. The boolean value computed by the output gate is denoted by $f_C(a_1, \ldots, a_{\alpha(C)})$. In such a way the B-circuit C computes the $\alpha(C)$-ary boolean function f_C. For more formal definitions see [Vol99]. Furthermore, let $\mathrm{BF}[B] =_{\mathrm{df}} \{f_C \mid C \text{ is a } B\text{-circuit}\}$ be the class of all boolean functions which can be computed by B-circuits.

Now let $\mathbf{A}$ be a property related to boolean functions. For a finite set B of boolean functions define

$$\mathbf{A}(B) =_{\mathrm{df}} \{(C, a) \mid C \text{ is a } B\text{-circuit such that } (f_C, a) \text{ has property } \mathbf{A}\}.$$

In order to study the complexity of $\mathbf{A}(B)$ the following question is very important: How we can relate the complexity of $\mathbf{A}(B)$ and $\mathbf{A}(B')$ by relating the sets B and B' of boolean functions themselves? The following proposition gives a satisfactory answer.

Proposition 3 *Let* $\mathbf{A}$ *be a property of boolean functions, and let* B *and* B' *be finite sets of boolean functions.*

1. *If* $B \subseteq \mathrm{BF}[B']$ *then* $\mathbf{A}(B) \leq_{\mathrm{m}}^{\log} \mathbf{A}(B')$.
2. *If* $\mathrm{BF}[B] = \mathrm{BF}[B']$ *then* $\mathbf{A}(B) \equiv_{\mathrm{m}}^{\log} \mathbf{A}(B')$.

Proof: Statement 2 is an immediate consequence of Statement 1. For the latter just mention the obvious fact that $B \subseteq \mathrm{BF}[B']$ implies that there exists a logspace computable function g converting every B-circuit C into a B'-circuit $g(C)$ such that $f_C = f_{g(C)}$. (An h-gate in C is simply replaced by a B'-circuit computing h.) ❑

In what follows we will make substantial use of the fact that the complexity of $\mathbf{A}(B)$ depends only on $\mathrm{BF}[B]$. To this end it is necessary to study the classes of boolean functions which have the form $\mathrm{BF}[B]$. It turns out that this are exactly those classes of boolean functions which contain the identity function and which are closed under superposition (i.e., substitution, permutation of variables, identification of variables, and introduction of fictive variables). In the following section we will report on important results which are known on these classes.

4 Closed Classes of Boolean Functions

A function $f : \{0,1\}^n \to \{0,1\}$ with $n \geq 0$ is called an *n-ary boolean function*. By BF we denote the class of all boolean functions. In particular, let c_0 and c_1 be the 0-ary constant functions having value 0 and 1, resp., let id and non be the unary functions defined by $\mathrm{id}(a) = a$ and $\mathrm{non}(a) = 1 \Leftrightarrow a = 0$, let et, vel, and aut be the binary functions defined by $\mathrm{et}(a,b) = 1 \Leftrightarrow a = b = 1$, $\mathrm{vel}(a,b) = 0 \Leftrightarrow a = b = 0$, and $\mathrm{aut}(a,b) = 1 \Leftrightarrow a \neq b$. We also write 0 instaed of c_0, 1 instaed of c_1, $\overline{x}$ or $\neg x$ instead of $\mathrm{non}(x)$, $x \wedge y$, $x \cdot y$, or xy instead of $\mathrm{et}(x,y)$, $x \vee y$ instead of $\mathrm{vel}(x,y)$, and $x \oplus y$ instead of $\mathrm{aut}(x,y)$. For $i \in \{1,2,\ldots,n\}$, the i-th variable of the n-ary boolean function f 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 set B of boolean functions let $[B]$ be the smallest class which contains B and which is closed under superposition (i.e. substitution, permutation of variables and identification of variables, introduction of fictive variables). A set B of boolean functions is called a *base* of the class F of boolean functions if $[B] = F$ and it is called *complete* if $[B] = \mathrm{BF}$. A class F of boolean functions is called *closed* if $[F] = F$.

The closed classes of boolean functions are closely related to the sets of boolean functions computed by B-circuits (cf. [Wag94]).

Proposition 4 *For any set* B *of boolean functions,* $\mathrm{BF}[B] = [B \cup \{\mathrm{id}\}]$.

Now consider some special properties of boolean functions. An n-ary boolean function f is said to be

- *a-reproducing* iff $f(a,a,\ldots,a) = a$ ($a \in \{0,1\}$),
- *linear* iff there exist $a_0,a_1,\ldots,a_n \in \{0,1\}$ such that $f(b_1,\ldots,b_n) = a_0 \oplus (a_1 \cdot b_1) \oplus (a_2 \cdot b_2) \oplus \ldots \oplus (a_n \cdot b_n)$ for all $b_0,b_1,\ldots,b_n \in \{0,1\}$,
- *self-dual* iff $f(a_1,\ldots,a_n) = \overline{f(\overline{a_1},\ldots\overline{a_n})}$ for all $a_1,\ldots,a_n \in \{0,1\}$,
- *monotone* iff $f_m(a_1,\ldots,a_n) \leq f_m(b_1,\ldots,b_n)$ for all $a_1,\ldots,a_n,b_1,\ldots,b_n \in \{0,1\}$ such that $a_1 \leq b_1, a_2 \leq b_2,\ldots,a_n \leq b_n$,
- *a-separating* iff there exists an $i \in \{1,2,\ldots,n\}$ 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 $i \in \{1, 2, \ldots, n\}$ such that $U \subseteq \{0,1\}^{i-1} \times \{a\} \times \{0,1\}^{n-i}$ ($a \in \{0,1\}, m \geq 2$).

The classes of all boolean functions which are 0-reproducing, 1-reproducing, linear, self-dual, monotone, 0-separating, 1-separating, 0-separating of degree m, and 1-separating of degree m, resp., 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. In this paper we will make substantial use of his main results which are presented in the following two theorems. For a detailed presentation see also [JGK70].

Theorem 5 ([Pos41]) *1. The complete list of closed classes of boolean functions is:*
BF, R_0, R_1, $\mathrm{R} =_{\mathrm{df}} \mathrm{R}_0 \cap \mathrm{R}_1$,
M, $\mathrm{M}_0 =_{\mathrm{df}} \mathrm{M} \cap \mathrm{R}_0$, $\mathrm{M}_1 =_{\mathrm{df}} \mathrm{M} \cap \mathrm{R}_1$, $\mathrm{M}_2 =_{\mathrm{df}} \mathrm{M} \cap \mathrm{R}$,
D, $\mathrm{D}_1 =_{\mathrm{df}} \mathrm{D} \cap \mathrm{R}$, $\mathrm{D}_2 =_{\mathrm{df}} \mathrm{D} \cap \mathrm{M}$,
L, $\mathrm{L}_0 =_{\mathrm{df}} \mathrm{L} \cap \mathrm{R}_0$, $\mathrm{L}_1 =_{\mathrm{df}} \mathrm{L} \cap \mathrm{R}_1$, $\mathrm{L}_2 =_{\mathrm{df}} \mathrm{L} \cap \mathrm{R}$, $\mathrm{L}_3 =_{\mathrm{df}} \mathrm{L} \cap \mathrm{D}$,
S_0, $\mathrm{S}_{02} =_{\mathrm{df}} \mathrm{S}_0 \cap \mathrm{R}$, $\mathrm{S}_{01} =_{\mathrm{df}} \mathrm{S}_0 \cap \mathrm{M}$, $\mathrm{S}_{00} =_{\mathrm{df}} \mathrm{S}_0 \cap \mathrm{R} \cap \mathrm{M}$,
S_1, $\mathrm{S}_{12} =_{\mathrm{df}} \mathrm{S}_1 \cap \mathrm{R}$, $\mathrm{S}_{11} =_{\mathrm{df}} \mathrm{S}_1 \cap \mathrm{M}$, $\mathrm{S}_{10} =_{\mathrm{df}} \mathrm{S}_1 \cap \mathrm{R} \cap \mathrm{M}$,
S_0^m, $\mathrm{S}_{02}^m =_{\mathrm{df}} \mathrm{S}_0^m \cap \mathrm{R}$, $\mathrm{S}_{01}^m =_{\mathrm{df}} \mathrm{S}_0^m \cap \mathrm{M}$, $\mathrm{S}_{00}^m =_{\mathrm{df}} \mathrm{S}_0^m \cap \mathrm{R} \cap \mathrm{M}$ *for* $m \geq 2$,
S_1^m, $\mathrm{S}_{12}^m =_{\mathrm{df}} \mathrm{S}_1^m \cap \mathrm{R}$, $\mathrm{S}_{11}^m =_{\mathrm{df}} \mathrm{S}_1^m \cap \mathrm{M}$, $\mathrm{S}_{10}^m =_{\mathrm{df}} \mathrm{S}_1^m \cap \mathrm{R} \cap \mathrm{M}$ *for* $m \geq 2$,
$\mathrm{E} =_{\mathrm{df}} [\mathrm{et}] \cup [\mathrm{c}_0] \cup [\mathrm{c}_1]$, $\mathrm{E}_0 =_{\mathrm{df}} [\mathrm{et}] \cup [\mathrm{c}_0]$, $\mathrm{E}_1 =_{\mathrm{df}} [\mathrm{et}] \cup [\mathrm{c}_1]$, $\mathrm{E}_2 =_{\mathrm{df}} [\mathrm{et}]$,
$\mathrm{V} =_{\mathrm{df}} [\mathrm{vel}] \cup [\mathrm{c}_0] \cup [\mathrm{c}_1]$, $\mathrm{V}_0 =_{\mathrm{df}} [\mathrm{vel}] \cup [\mathrm{c}_0]$, $\mathrm{V}_1 =_{\mathrm{df}} [\mathrm{vel}] \cup [\mathrm{c}_1]$, $\mathrm{V}_2 =_{\mathrm{df}} [\mathrm{vel}]$,
$\mathrm{N} =_{\mathrm{df}} [\mathrm{non}] \cup [\mathrm{c}_0] \cup [\mathrm{c}_1]$, $\mathrm{N}_2 =_{\mathrm{df}} [\mathrm{non}]$,
$\mathrm{I} =_{\mathrm{df}} [\mathrm{id}] \cup [\mathrm{c}_0] \cup [\mathrm{c}_1]$, $\mathrm{I}_0 =_{\mathrm{df}} [\mathrm{id}] \cup [\mathrm{c}_0]$, $\mathrm{I}_1 =_{\mathrm{df}} [\mathrm{id}] \cup [\mathrm{c}_1]$, $\mathrm{I}_2 =_{\mathrm{df}} [\mathrm{id}]$,
$\mathrm{C} =_{\mathrm{df}} [\mathrm{c}_0] \cup [\mathrm{c}_1]$, $\mathrm{C}_0 =_{\mathrm{df}} [\mathrm{c}_0]$, $\mathrm{C}_1 =_{\mathrm{df}} [\mathrm{c}_1]$, $\emptyset$.

2. *The inclusional relationships between the closed classes of boolean functions are presented in 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 $[B]$.*

4. *There exists an algorithm which, given $f \in \mathrm{BF}$ and a finite set $B \subseteq \mathrm{BF}$, decides whether $f \in [B]$ or not.*

Let us consider an example. Define the boolean function f^3 such that $f(x, y, z) = 1$ iff exactly one argument has the value 1. Clearly, f is 0-reproducing but not 1-reproducing. Now assume that f is a linear function, i.e., $f(x, y, z) = a_0 \oplus (a_1 \cdot x) \oplus (a_2 \cdot y) \oplus (a_3 \cdot z)$ for suitable $a_0, a_1, a_2, a_3 \in \{0,1\}$. Since $f(0,0,0) = 0$ we know $a_0 = 0$ and it follows that $a_1 = a_2 = a_3 = 1$, 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 monotone since $f(1,0,0) = 1$ and $f(1,1,1) = 0$, and it is not self-dual because of $f(0,0,0) = f(1,1,1)$. Finally, f cannot be 1-separating of degree 2 because of $f(0,0,1) = f(0,1,0) = 1$. 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 $[\{f\}] = \mathrm{R}_0$.

Corollary 6 *A class B of boolean functions is complete if and only if $B \not\subseteq \mathrm{R}_0$, $B \not\subseteq \mathrm{R}_1$, $B \not\subseteq \mathrm{L}$, $B \not\subseteq \mathrm{D}$, and $B \not\subseteq \mathrm{M}$.*

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

Theorem 7 ([Pos41]) *Every closed class of boolean functions has a finite base. In particular:*

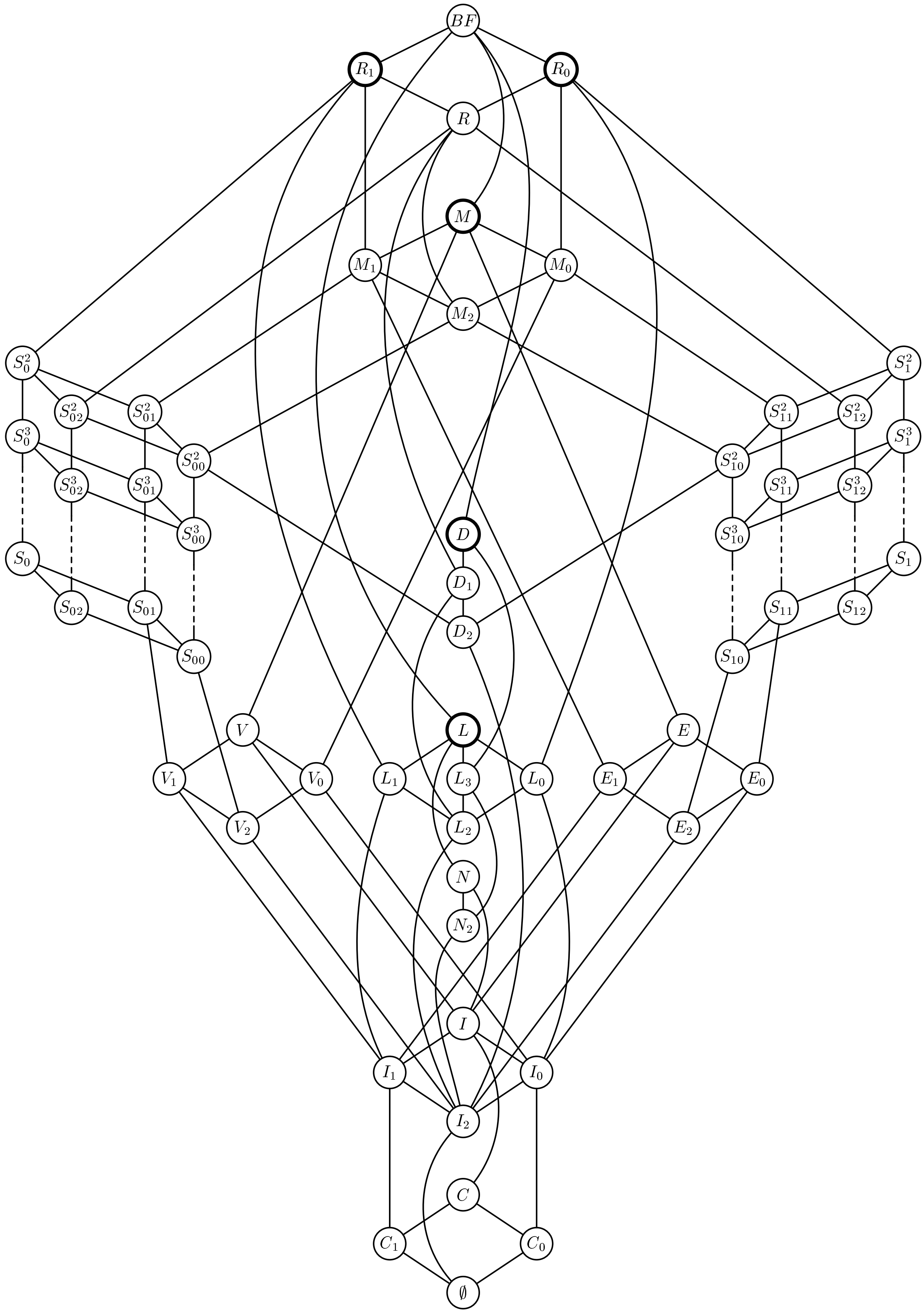

Figure 1: Graph of all classes of boolean function being closed under superposition

1. $\{\mathrm{et}, \mathrm{vel}, \mathrm{non}\}$ *is a base of* BF.
2. $\{\mathrm{et}, \mathrm{aut}\}$ *is a base of* R_0.
3. $\{\mathrm{vel}, \mathrm{x} \oplus \mathrm{y} \oplus 1\}$ *is a base of* R_1.
4. $\{\mathrm{et}, \mathrm{vel}, \mathrm{c}_0, \mathrm{c}_1\}$ *is a base of* M.
5. $\{x\overline{y} \vee x\overline{z} \vee \overline{y}\,\overline{z}\}$ *is a base of* D.
6. $\{\mathrm{aut}, \mathrm{c}_1\}$ *is a base of* L.
7. $\{\mathrm{aut}\}$ *is a base of* L_0.
8. $\{x \oplus y \oplus 1\}$ *is a base of* L_1.
9. $\{x \oplus y \oplus z\}$ *is a base of* L_2.
10. $\{x \vee \overline{y}\}$ *is a base of* S_0.
11. $\{x \wedge \overline{y}\}$ *is a base of* S_1.
12. $\{x \vee (y \wedge z)\}$ *is a base of* S_{00}.
13. $\{x \wedge (y \vee z)\}$ *is a base of* S_{10}.
14. $\{\mathrm{et}, \mathrm{c}_0, \mathrm{c}_1\}$ *is a base of* E.
15. $\{\mathrm{et}\}$ *is a base of* E_2.
16. $\{\mathrm{vel}, \mathrm{c}_0, \mathrm{c}_1\}$ *is a base of* V.
17. $\{\mathrm{vel}\}$ *is a base of* V_2.
18. $\{\mathrm{non}, \mathrm{c}_1\}$ *is a base of* N.

For a n-ary boolean function f define the n-ary 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 8 *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).

5 Circuit Value

Let B be a finite set of boolean functions. The *circuit value problem for B-circuits* is defined as

$$\mathbf{VAL}(B) =_{\mathrm{df}} \{(C, a) \mid C \text{ is a } B\text{-circuit, } a \in \{0,1\}^{\alpha(C)}, \text{ and } f_C(a) = 1\}$$

It is obvious that $\mathbf{VAL}(B) \in \mathrm{P}$ for every finite set B of boolean functions. The following facts on the complexity of $\mathbf{VAL}(B)$ can be found in or easily be derived from the literature.

Theorem 9 *Let B be a finite set of boolean functions.*

1. *[Lad77] If* $[B] = \mathrm{BF}$ *then* $\mathbf{VAL}(B)$ *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* P.
2. *[Gol77] If* $\{\mathrm{et}, \mathrm{vel}\} \subseteq [\mathrm{B}]$ *then* $\mathbf{VAL}(B)$ *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* P.
3. *[GP86] Let B be a set of binary boolean functions. If* $\{\mathrm{et}, \mathrm{vel}\} \subseteq [\mathrm{B}]$ *or (*$B \not\subseteq \mathrm{L}$ *and* $B \not\subseteq \mathrm{M}$*) then* $\mathbf{VAL}(B)$ *is* $\leq_{\mathrm{m}}^{\log}$*-complete for* P*, otherwise* $\mathbf{VAL}(B)$ *is acceptable in* $\log^2 n$ *space.*

We start the investigation of the complexity of $\mathbf{VAL}(B)$ by strengthening Proposition 3 for this particular case.

Proposition 10 *Let B and B' be finite sets of boolean functions.*

1. *If* $B \subseteq [B' \cup \{\mathrm{id}, \mathrm{c}_0, \mathrm{c}_1\}]$ *then* $\mathbf{VAL}(B) \leq_{\mathrm{m}}^{\log} \mathbf{VAL}(B')$.
2. *If* $[B \cup \{\mathrm{id}, \mathrm{c}_0, \mathrm{c}_1\}] = [\mathrm{B}' \cup \{\mathrm{id}, \mathrm{c}_0, \mathrm{c}_1\}]$ *then* $\mathbf{VAL}(B) \equiv_{\mathrm{m}}^{\log} \mathbf{VAL}(B')$.

Proof: As for Proposition 3, but additionally a c_0-gate (c_1-gate, resp.) is replaced by an input gate with boolean value 0 (1, resp.). ❑

Hence, for the study of the complexity of $\mathbf{VAL}(B)$, only those closed classes of boolean functions are of importance which contain id, c_0, and c_1. A close inspection of Figure 1 shows:

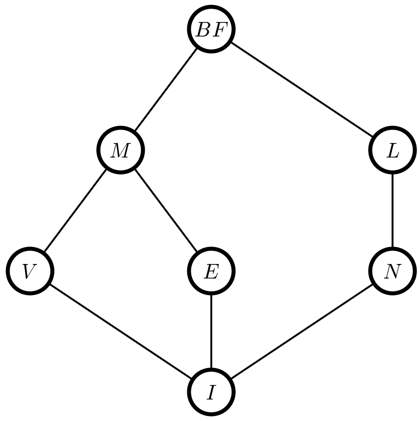

Figure 2: All closed classes of boolean functions containing id, c_0, and c_1.

Proposition 11 *The closed classes of boolean functions containing* id, c_0, *and* c_1 *are* BF, M, V, E, L, N, *and* I. *The inclusional relationships between these classes are presented in Figure 2.*

Now we are ready to prove our main theorem on the complexity of $\mathbf{VAL}(B)$.

Theorem 12 *Let B be a finite set of boolean functions.*

$\boxed{\textit{Is } B \subseteq \mathrm{N}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\mathbf{VAL}(B) \in \mathrm{L}}$

$\Downarrow$ *no*

$\boxed{\textit{is } B \subseteq \mathrm{V} \textit{ or } \mathrm{B} \subseteq \mathrm{E}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\mathbf{VAL}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \mathrm{NL}}$

$\Downarrow$ *no*

$\boxed{\textit{is } B \subseteq \mathrm{L}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\mathbf{VAL}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \oplus\mathrm{L}}$

$\Downarrow$ *no*

$\boxed{\mathbf{VAL}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \mathrm{P}}$

There exists an algorithm which decides which of the cases above takes place.

Proof: 1. If $B \subseteq \mathrm{N} = [\{\mathrm{non}, \mathrm{id}, \mathrm{c}_0, \mathrm{c}_1\}]$ then, by Proposition 10, we get $\mathbf{VAL}(B) \leq_{\mathrm{m}}^{\log} \mathbf{VAL}(\{\mathrm{non}\})$. For a $\{\mathrm{non}\}$-circuit C, there is $f_C(a) = 1$ iff the "backward path" from the output gate either has even length and ends in an input with value 1 or has odd length and ends in an input with value 0. This can be checked in logarithmic space.

2. Let $B \not\subseteq \mathrm{N}$ and $B \subseteq \mathrm{V}$. By Proposition 11 and Theorem 7 we obtain $[B \cup \{\mathrm{id}, \mathrm{c}_0, \mathrm{c}_1\}] = \mathrm{V} = [\{\mathrm{vel}, \mathrm{id}, \mathrm{c}_0, \mathrm{c}_1\}]$, and by Proposition 10 we get $\mathbf{VAL}(B) \equiv_{\mathrm{m}}^{\log} \mathbf{VAL}(\{\mathrm{vel}\})$.
Upper bound. For a $\{\mathrm{vel}\}$-circuit C there is $f_C(a) = 1$ iff at least one of the "backward paths" from the output gate ends in an input gate with value 1. This is a graph accessibility problem which can be solved in nondeterministic logarithmic space.
Lower bound. We prove $\mathrm{GAP} \leq_{\mathrm{m}}^{\log} \mathbf{VAL}(\{\mathrm{vel}\})$ (cf. Theorem 2.1). Given a directed acyclic graph G whose vertices have outdegree 0 or 2, a start vertex s, and a target vertex t with outdegree 0. We construct in logarithmic space a $\{\mathrm{vel}\}$-circuit C as follows: Every outdegree 2 vertex becomes a vel-gate, the vertex t becomes an input gate with input value 1, and every other outdegree 0 vertex becomes an input gate with

input value 0. The vertex s becomes the output gate of C. Let a be the input string constructed in this way. Obviously, there exists a path in G from s to t iff $(C, a) \in \mathbf{VAL}(B)$.

3. The case $B \not\subseteq \mathrm{N}$ and $B \subseteq \mathrm{E}$ is treated in the same way; just replace V by E, vel by et, GAP by $\overline{\mathrm{GAP}}$, etc.

4. Let $B \not\subseteq \mathrm{N}$ and $B \subseteq \mathrm{L}$. By Proposition 11 and Theorem 7 we obtain $[B \cup \{\mathrm{id}, \mathrm{c}_0, \mathrm{c}_1\}] = \mathrm{L} = [\{\mathrm{aut}, \mathrm{id}, \mathrm{c}_0, \mathrm{c}_1\}]$, and by Proposition 10 we get $\mathbf{VAL}(B) \equiv_{\mathrm{m}}^{\log} \mathbf{VAL}(\{\mathrm{aut}\})$.
Upper bound. For a $\{\mathrm{aut}\}$-circuit C there is $f_C(a) = 1$ iff the number of "backward paths" from the output gate to an input gate with value 1 is odd. This can be checked by a $\oplus$L-computation.
Lower bound. We prove GOAP $\leq_{\mathrm{m}}^{\log}$ $\mathbf{VAL}(\{\mathrm{aut}\})$ (cf. Theorem 2.2). This can be done in the same way as GAP $\leq_{\mathrm{m}}^{\log}$ $\mathbf{VAL}(\{\mathrm{vel}\})$ in Item 2 of this proof.

5. Let $B \not\subseteq \mathrm{V}$, $B \not\subseteq \mathrm{E}$, and $B \not\subseteq \mathrm{L}$. By Proposition 11 we have $\{\mathrm{et}, \mathrm{vel}\} \subseteq \mathrm{M} \subseteq [\mathrm{B} \cup \{\mathrm{id}, \mathrm{c}_0, \mathrm{c}_1\}]$, and by Proposition 10 we get $\mathbf{VAL}(\{\mathrm{et}, \mathrm{vel}\}) \leq_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$. However, $\mathbf{VAL}(\{\mathrm{et}, \mathrm{vel}\})$ is $\leq_{\mathrm{m}}^{\log}$-complete for P by Theorem 9, and $\mathbf{VAL}(B)$ is obviously in P. ❏

Notice that Theorem 12 improves the GOLDSCHLAGER-PARBERRY result (Theorem 9.3) in two ways:

- Theorem 9.3 applies only to finite sets B of *binary* boolean functions whereas Theorem 12 applies to all finite sets B of boolean functions.
- Theorem 9.3 distinguishes only between "P-complete" and "acceptable in $\log^2 n$ space", whereas Theorem 12 splits the latter case into the cases "NL-complete", "$\oplus$L-complete", and "acceptable in $\log n$ space" (cf. Theorem 1).

Finally let us mention that the GOLDSCHLAGER-PARBERRY criterion for sets B of binary boolean functions is not valid for arbitrary finite sets B of boolean functions. Take for example $B = \{xy \vee xz\}$. By Theorem 7 we know that $[B] = \mathrm{S}_{10}$ and $[\{\mathrm{vel}\}] = \mathrm{V}_2$. A short look at Figure 1 shows that vel $\notin$ [B]. Because of $B \subseteq \mathrm{M}$ this would correspond to the "otherwise" case of Theorem 9.3. However, Theorem 12 shows that $\mathbf{VAL}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for P.

6 Satisfiability and Tautology

In this section we study the complexity of the satisfiability problem and the tautology problem for B-circuits. For a finite set B of boolean functions we define

$$\mathbf{SAT}(B) =_{\mathrm{df}} \{C \mid C \text{ is a } B\text{-circuit and } f_C(a) = 1 \text{ for at least one } a \in \{0,1\}^{\alpha(C)}\}$$

and

$$\mathbf{TAUT}(B) =_{\mathrm{df}} \{C \mid C \text{ is a } B\text{-circuit and } f_C(a) = 1 \text{ for all } a \in \{0,1\}^{\alpha(C)}\}.$$

It is obvious that $\mathbf{SAT}(B) \in \mathrm{NP}$ and $\mathbf{TAUT}(B) \in \text{co-NP}$ for every finite set B of boolean functions. The following facts about the complexity of $\mathbf{SAT}(B)$ can easily be derived from the literature where the special case of B-formulae (i.e., "tree-like" B-circuits) is considered.

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

1. *[Coo71] If $[B] = \mathrm{BF}$ then $\mathbf{SAT}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for* NP *and $\mathbf{TAUT}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for* co-NP.
2. *[Lew78] Let B be a set of binary boolean functions. If $\{x \wedge \overline{y}\} \subseteq B$, $\{\oplus, \mathrm{et}\} \subseteq \mathrm{B}$, or $\{\oplus, \mathrm{vel}\} \subseteq \mathrm{B}$ then $\mathbf{SAT}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for* NP.

In some special cases the complexity of $\mathbf{SAT}(B)$ can be related to the complexity of $\mathbf{VAL}(B)$.

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

1. *If $B \subseteq \mathrm{M}$ then $\mathbf{SAT}(B) \leq_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$.*
2. *If $\mathrm{c}_0 \in [B] \subseteq \mathrm{M}$ then $\mathbf{VAL}(B) \leq_{\mathrm{m}}^{\log} \mathbf{SAT}(B)$.*
3. *If $\mathrm{c}_0, \mathrm{c}_1 \in [B]$ then $\mathbf{VAL}(B) \leq_{\mathrm{m}}^{\log} \mathbf{SAT}(B)$.*

Proof: 1. The reduction is given by $C \in \mathbf{SAT}(B) \Leftrightarrow (C, 1^{\alpha(C)}) \in \mathbf{VAL}(B)$.
2. Convert a B-circuit C with an input string $a \in \{0,1\}^{\alpha(C)}$ into a B-circuit C' as follows: Replace every input gate with value 0 by a B-circuit computing c_0 and replace every input gate with value 1 by an input gate with a new input variable. Obviously, $(C, a) \in \mathbf{VAL}(B) \Leftrightarrow C' \in \mathbf{SAT}(B)$.
3. Convert a B-circuit C with an input string $a \in \{0,1\}^{\alpha(C)}$ into a B-circuit C' as follows: Replace every input gate with value 0 by a B-circuit computing c_0 and replace every input gate with value 1 by a B-circuit computing c_1. Obviously, $(C, a) \in \mathbf{VAL}(B) \Leftrightarrow C' \in \mathbf{SAT}(B)$. ❑

Proposition 15 *If B is a finite set of boolean functions containing* et *then* $\mathbf{SAT}(B \cup \{\mathrm{c}_1\}) \leq_{\mathrm{m}}^{\log} \mathbf{SAT}(B)$.

Proof: A $(B \cup \{\mathrm{c}_1\})$-circuit C with input variables $x_1, \ldots, x_n$ is converted into a B-circuit C' with a new input variable z by replacing every c_1-gate by a z-gate. Hence $f_{C'}(x_1, \ldots, x_n, 1) = f_C(x_1, \ldots, x_n)$. Now construct a B-circuit C'' such that $f_{C''}(x_1, \ldots, x_n, z) = f_{C'}(x_1, \ldots, x_n, z) \wedge z$. We obtain: C is satisfiable iff C'' is satisfiable. ❑

Now we are ready to prove the main theorem on the complexity of $\mathbf{SAT}(B)$.

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

$$\boxed{\textit{Is } B \subseteq \mathrm{R}_1 \textit{ or } \mathrm{B} \subseteq \mathrm{D} \textit{ or } \mathrm{B} \subseteq \mathrm{N}?} \overset{\textit{yes}}{\Longrightarrow} \boxed{\mathbf{SAT}(B) \in \mathrm{L}}$$

$$\Downarrow \textit{no}$$

$$\boxed{\textit{is } B \subseteq \mathrm{V} \textit{ or } \mathrm{B} \subseteq \mathrm{E}?} \overset{\textit{yes}}{\Longrightarrow} \boxed{\mathbf{SAT}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \mathrm{NL}}$$

$$\Downarrow \textit{no}$$

$$\boxed{\textit{is } B \subseteq \mathrm{L}?} \overset{\textit{yes}}{\Longrightarrow} \boxed{\mathbf{SAT}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \oplus\mathrm{L}}$$

$$\Downarrow \textit{no}$$

$$\boxed{\textit{is } B \subseteq \mathrm{M}?} \overset{\textit{yes}}{\Longrightarrow} \boxed{\mathbf{SAT}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \mathrm{P}}$$

$$\Downarrow \textit{no}$$

$$\boxed{\mathbf{SAT}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \mathrm{NP}}$$

There exists an algorithm which decides which of the cases above takes place.

Proof: 1. Let $B \subseteq \mathrm{R}_1$, and let C be a B-circuit. By Proposition 4 we obtain $f_C \in [B \cup \{id\}] \subseteq \mathrm{R}_1$. Hence $f_C(1, \ldots 1) = 1$, and $C \in \mathbf{SAT}(B)$.
2. Let $B \subseteq D$, and let C be a B-circuit. By Proposition 4 we obtain $f_C \in [B \cup \{id\}] \subseteq \mathrm{D}$. Hence $f_C(0, \ldots, 0) = 1$ or $f_C(1, \ldots, 1) = 1$, and $C \in \mathbf{SAT}(B)$.
3. If $B \subseteq \mathrm{N} = [\{\mathrm{non}, \mathrm{c}_1\}]$ then, by Proposition 3, we get $\mathbf{SAT}(B) \leq_{\mathrm{m}}^{\log} \mathbf{SAT}(\{\mathrm{non}, \mathrm{c}_1\})$. For a $\{\mathrm{non}, \mathrm{c}_1\}$-circuit C, there exists an $a \in \{0,1\}^{\alpha(C)}$ such that $f_C(a) = 1$ iff the “backward path” from the

output gate either ends in an input gate or it ends in a c_1-gate and has even length. This can be checked in logarithmic space.

4. Let $B \not\subseteq \mathrm{R}_1$, $B \not\subseteq \mathrm{N}$, and $B \subseteq \mathrm{V}$. An inspection of Figure 1 yields $[B] = \mathrm{V}_0$ or $[B] = \mathrm{V}$. Consequently $\mathrm{c}_0 \in \mathrm{V}_0 \subseteq [\mathrm{B}] \subseteq \mathrm{M}$, and by Proposition 14 we obtain $\mathbf{SAT}(B) \equiv_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$. Now Theorem 12 yields that $\mathbf{SAT}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for NL.

5. The case $B \not\subseteq \mathrm{R}_1$, $B \not\subseteq \mathrm{N}$, and $B \subseteq \mathrm{E}$ is treated in the same way; just replace V by E.

6. Let $B \not\subseteq \mathrm{R}_1$, $B \not\subseteq \mathrm{N}$, and $B \subseteq \mathrm{L}$. An inspection of Figure 1 yields $[B] = \mathrm{L}_0$ or $[B] = \mathrm{L}$. Consequently, $\mathrm{aut} \in \mathrm{L}_0 \subseteq [\mathrm{B} \cup \{\mathrm{id}\}]$ and, by Theorem 7, we have $B \subseteq \mathrm{L} = [\{\mathrm{aut}, \mathrm{c}_1, \mathrm{id}\}]$. By Proposition 3 we obtain $\mathbf{SAT}(\{\mathrm{aut}\}) \leq_{\mathrm{m}}^{\log} \mathbf{SAT}(\mathrm{B}) \leq_{\mathrm{m}}^{\log} \mathbf{SAT}(\{\mathrm{aut}, \mathrm{c}_1\})$.

Upper bound. For an $\{\mathrm{aut}, \mathrm{c}_1\}$-circuit C with input variables $x_1, \ldots, x_n$ let m_0 be the number of paths from the output gate to a constant c_1-gate, and for $i = 1, \ldots, k$, let m_i be the number of paths from the output gate to an x_i-gate. Obviously, x_i is fictive in f_C iff m_i is even. Consequently, C is satisfiable iff at least one of the numbers $m_0, m_1, \ldots, m_k$ is odd. This can be checked by a $\mathrm{L}^{\oplus\mathrm{L}}$ computation. By Theorem 1 we have $\mathrm{L}^{\oplus\mathrm{L}} = \oplus\mathrm{L}$.

Lower Bound. We prove GOAP $\leq_{\mathrm{m}}^{\log}$ $\mathbf{SAT}(\{\mathrm{aut}\})$ (cf. Theorem 2). Given a directed acyclic graph G whose vertices have outdegree 0 or 2, a start vertex s, and a target vertex t with outdegree 0. We construct in logarithmic space a $\{\mathrm{aut}\}$-circuit C with only one input variable x as follows: Every outdegree 2 vertex becomes a aut-gate, the vertex t becomes an x-gate, and every other outdegree 0 vertex becomes an aut-gate with both incoming edges from an x-gate. The vertex s becomes the output gate of C. If there is an odd number of paths in G from s to t, then $f_C = \mathrm{id}$, otherwise f_C is the unary constant 0 function.

7. Let $B \not\subseteq \mathrm{R}_1$, $B \not\subseteq \mathrm{D}$, $B \not\subseteq \mathrm{V}$, $B \not\subseteq \mathrm{E}$, and $B \subseteq \mathrm{M}$. An inspection of Figure 1 yields $\mathrm{c}_0 \in \mathrm{S}_{11} \subseteq [B] \subseteq \mathrm{M}$. By Proposition 14 we obtain $\mathbf{SAT}(B) \equiv_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$. Now Theorem 12 yields that $\mathbf{SAT}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for P.

8. Let $B \not\subseteq \mathrm{R}_1$, $B \not\subseteq \mathrm{D}$, $B \not\subseteq \mathrm{L}$, and $B \not\subseteq \mathrm{M}$. An inspection of Figure 1 yields $\mathrm{S}_1 \subseteq [B]$. Observe $x \wedge \overline{y} \in \mathrm{S}_1$. By Theorem 13 we get that $\mathbf{SAT}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for NP. ❑

Notice that Theorem 16 improves the LEWIS result (Theorem 13.2) because the latter applies only to finite sets B of *binary* boolean functions whereas Theorem 12 applies to all finite sets B of boolean functions.

Now turn to the tautology problem. For a B-circuit C let $\mathrm{dual}(C)$ be that $\mathrm{dual}(B)$-circuit which arises from C by replacing every f-gate by a $\mathrm{dual}(f)$-gate. Clearly, $f_{\mathrm{dual}(C)} = \mathrm{dual}(f_C)$. We will use the following duality principle.

Proposition 17 *For any finite set B of boolean functions,* $\mathbf{TAUT}(B) \equiv_{\mathrm{m}}^{\log} \overline{\mathbf{SAT}(\mathrm{dual}(B))}$.

Proof: For a B-circuit C there holds $C \in \mathbf{TAUT}(B) \Leftrightarrow f_C \equiv 1 \Leftrightarrow f_{\mathrm{dual}(C)} \equiv 0 \Leftrightarrow \mathrm{dual}(C) \in \overline{\mathbf{SAT}(\mathrm{dual}(B))}$. ❑

Now we are ready to prove the main theorem on the complexity of $\mathbf{TAUT}(B)$.

Theorem 18 *Let be B a finite set of boolean functions.*

$$\boxed{\textit{Is } B \subseteq \mathrm{R}_0 \textit{ or } \mathrm{B} \subseteq \mathrm{D} \textit{ or } \mathrm{B} \subseteq \mathrm{N}?} \overset{\textit{yes}}{\Longrightarrow} \boxed{\boxed{\mathbf{TAUT}(B) \in \mathrm{L}}}$$

$$\Downarrow \textit{no}$$

$$\boxed{\textit{is } B \subseteq \mathrm{V} \textit{ or } \mathrm{B} \subseteq \mathrm{E}?} \overset{\textit{yes}}{\Longrightarrow} \boxed{\boxed{\mathbf{TAUT}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \mathrm{NL}}}$$

$$\Downarrow \textit{no}$$

$$\boxed{\textit{is } B \subseteq \mathrm{L}?} \overset{\textit{yes}}{\Longrightarrow} \boxed{\boxed{\mathbf{TAUT}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \oplus\mathrm{L}}}$$

$$\Downarrow \textit{no}$$

$$\boxed{\textit{is } B \subseteq \mathrm{M}?} \overset{\textit{yes}}{\Longrightarrow} \boxed{\boxed{\mathbf{TAUT}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \mathrm{P}}}$$

$$\Downarrow \textit{no}$$

$$\boxed{\boxed{\mathbf{TAUT}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \text{co-NP}}}$$

There exists an algorithm which decides which of the cases above takes place.

Proof: The theorem follows from Theorem 16 and Proposition 17 by using the facts $\mathrm{dual}(\mathrm{N}) = \mathrm{N}$, $\mathrm{dual}(\mathrm{D}) = \mathrm{D}$, $\mathrm{dual}(\mathrm{R}_0) = \mathrm{R}_1$, $\mathrm{dual}(\mathrm{V}) = \mathrm{E}$, $\mathrm{dual}(\mathrm{E}) = \mathrm{V}$, $\mathrm{dual}(\mathrm{L}) = \mathrm{L}$, and $\mathrm{dual}(\mathrm{M}) = \mathrm{M}$ (cf. Proposition 8.2) as well as co-L $=$ L, co-NL $=$ NL, co- $\oplus$ L $= \oplus$L, and co-P $=$ P (cf. Theorem 1). For example: If $B \not\subseteq \mathrm{R}_0$, $B \not\subseteq \mathrm{D}$, $B \not\subseteq \mathrm{N}$, and $B \subseteq \mathrm{V}$ then $\mathrm{dual}(B) \not\subseteq \mathrm{R}_1$, $\mathrm{dual}(B) \not\subseteq \mathrm{D}$, $\mathrm{dual}(B) \not\subseteq \mathrm{N}$, and $\mathrm{dual}(B) \subseteq \mathrm{E}$. By Theorem 16 the problem $\mathbf{SAT}(\mathrm{dual}(B))$ is $\leq_{\mathrm{m}}^{\log}$-complete for NL, and by Proposition 17 the problem $\mathbf{TAUT}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for co-NL $=$ NL. ❑

7 Quantifiers

In this section we study the complexity of problems defined by B-circuits with quantified input variables. For $Q_1, Q_2, \ldots, Q_m \in \{\exists, \forall\}$ and $k \geq 1$, we call $Q_1 Q_2 \ldots Q_m$ a Σ_k*-string* (a Π_k*-string*) if $Q_1 = \exists$ ($Q_1 = \forall$, resp.) and there are at most $k-1$ alternations between $\exists$ and $\forall$ in $Q_1 Q_2 \ldots Q_m$. For a finite set B of boolean functions define:

$$\begin{array}{rcll}
\mathbf{\Sigma}_k(B) & =_{\mathrm{df}} & \{Q_1 x_1 Q_2 x_2 \ldots Q_m x_m C \mid & C \text{ is a } B\text{-circuit with variables } x_1, x_2, \ldots, x_m, \\
 & & & Q_1 Q_2 \ldots Q_m \text{ is a } \Sigma_k\text{-string and} \\
 & & & Q_1 x_1 Q_2 x_2 \ldots Q_m x_m f_C(x_1, x_2, \ldots, x_m) = 1\} \\
\mathbf{\Pi}_k(B) & =_{\mathrm{df}} & \{Q_1 x_1 Q_2 x_2 \ldots Q_m x_m C \mid & C \text{ is a } B\text{-circuit with variables } x_1, x_2, \ldots, x_m, \\
 & & & Q_1 Q_2 \ldots Q_m \text{ is a } \Pi_k\text{-string and} \\
 & & & Q_1 x_1 Q_2 x_2 \ldots Q_m x_m f_C(x_1, x_2, \ldots, x_m) = 1\} \\
\mathbf{QBC}(B) & =_{\mathrm{df}} & \{Q_1 x_1 Q_2 x_2 \ldots Q_m x_m C \mid & C \text{ is a } B\text{-circuit with variables } x_1, x_2, \ldots, x_m, \\
 & & & Q_1, Q_2, \ldots, Q_m \in \{\exists, \forall\} \text{ and} \\
 & & & Q_1 x_1 Q_2 x_2 \ldots Q_m x_m f_C(x_1, x_2, \ldots, x_m) = 1\}
\end{array}$$

Notice that $\mathbf{\Sigma}_1(B) = \mathbf{SAT}(B)$ and $\mathbf{\Pi}_1(B) = \mathbf{TAUT}(B)$ have already been treated in the previous section. Here we concentrate on the case $k \geq 2$.

It is obvious that $\mathbf{\Sigma}_k(B) \in \Sigma_{\mathrm{k}}^{\mathrm{p}}$, $\mathbf{\Pi}_k(B) \in \Pi_{\mathrm{k}}^{\mathrm{p}}$, and $\mathbf{QBC}(B) \in \mathrm{PSPACE}$ for every finite set B of boolean functions. The following results can easily be derived from the literature where these results are proved for the special case of B-formulae.

Theorem 19 *[SM73] Let B be a finite set of boolean functions such that $[B] = \mathrm{BF}$, and let $k \geq 1$. Then $\mathbf{\Sigma}_k(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for $\Sigma_{\mathrm{k}}^{\mathrm{p}}$, $\mathbf{\Pi}_k(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for $\Pi_{\mathrm{k}}^{\mathrm{p}}$, and $\mathbf{QBC}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for* PSPACE.

For sets B of monotone boolean functions the complexity of $\mathbf{\Sigma}_k(B)$, $\mathbf{\Pi}_k(B)$ and $\mathbf{QBC}(B)$ can be related to the complexity of $\mathbf{VAL}(B)$.

Proposition 20 *If $B \subseteq \mathrm{M}$ is a finite set of boolean functions then $\mathbf{\Sigma}_k(B) \equiv_{\mathrm{m}}^{\log} \mathbf{\Pi}_k(B) \equiv_{\mathrm{m}}^{\log} \mathbf{QBC}(B) \equiv_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$ for every $k \geq 2$.*

Proof: $\mathbf{\Sigma}_k(B) \leq_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$, $\mathbf{\Pi}_k(B) \leq_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$, and $\mathbf{QBC}(B) \leq_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$. Let C be a B-circuit with input variables $x_1, \ldots, x_n$ and let $Q_1, \ldots, Q_n \in \{\exists, \forall\}$. Defining $a_i = 1$ iff $Q_i = \exists$ for $1 \leq i \leq n$ we obtain $Q_1x_1Q_2x_2 \ldots Q_nx_nf_C(x_1, \ldots, x_n) = 1 \Leftrightarrow f(a_1, \ldots, a_n) = 1$.

$\mathbf{VAL}(B) \leq_{\mathrm{m}}^{\log} \mathbf{\Sigma}_k(B)$, $\mathbf{VAL}(B) \leq_{\mathrm{m}}^{\log} \mathbf{\Pi}_k(B)$, and $\mathbf{VAL}(B) \leq_{\mathrm{m}}^{\log} \mathbf{QBC}(B)$. Convert a B-circuit C with an input string $a \in \{0,1\}^{\alpha(C)}$ into a B-circuit C' with input variables x and z as follows: Every input gate with value 0 becomes an z-gate, and every input gate with value 1 becomes a x-gate. Obviously, $(C, a) \in \mathbf{VAL}(B) \Leftrightarrow \exists x \forall z C' \in \mathbf{\Sigma}_k(B) \Leftrightarrow \forall z \exists x C' \in \mathbf{\Pi}_k(B) \Leftrightarrow \forall z \exists x C' \in \mathbf{QBC}(B)$. ❑

Theorem 21 *Let be B a finite set of boolean functions, and let $k \geq 2$.*

Is $B \subseteq \mathrm{N}$? $\overset{yes}{\Longrightarrow}$ $\mathbf{\Sigma}_k(B), \mathbf{\Pi}_k(B), \mathbf{QBC}(B) \in \mathrm{L}$

$\Downarrow$ *no*

is $B \subseteq \mathrm{V}$ or $\mathrm{B} \subseteq \mathrm{E}$? $\overset{yes}{\Longrightarrow}$ *$\mathbf{\Sigma}_k(B), \mathbf{\Pi}_k(B)$, and $\mathbf{QBC}(B)$ are complete for* NL

$\Downarrow$ *no*

is $B \subseteq \mathrm{L}$? $\overset{yes}{\Longrightarrow}$ *$\mathbf{\Sigma}_k(B), \mathbf{\Pi}_k(B)$, and $\mathbf{QBC}(B)$ are complete for* $\oplus$L

$\Downarrow$ *no*

is $B \subseteq \mathrm{M}$? $\overset{yes}{\Longrightarrow}$ *$\mathbf{\Sigma}_k(B), \mathbf{\Pi}_k(B)$, and $\mathbf{QBC}(B)$ are complete for* P

$\Downarrow$ *no*

$\mathbf{\Sigma}_k(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for $\Sigma_{\mathrm{k}}^{\mathrm{p}}$
$\mathbf{\Pi}_k(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for $\Pi_{\mathrm{k}}^{\mathrm{p}}$
$\mathbf{QBC}(B)$ is $\leq_{\mathrm{m}}^{\log}$-complete for PSPACE

There exists an algorithm which decides which of the cases above takes place.

Proof: 1. If $B \subseteq \mathrm{N}$ then, by Proposition 3 and Theorem 7, we get $\mathbf{\Sigma}_k(B) \leq_{\mathrm{m}}^{\log} \mathbf{\Sigma}_k(\{\mathrm{non}, \mathrm{c}_1\})$, $\mathbf{\Pi}_k(B) \leq_{\mathrm{m}}^{\log} \mathbf{\Pi}_k(\{\mathrm{non}, \mathrm{c}_1\})$, and $\mathbf{QBC}(B) \leq_{\mathrm{m}}^{\log} \mathbf{QBC}(\{\mathrm{non}, \mathrm{c}_1\})$. For a $\{\mathrm{non}, \mathrm{c}_1\}$-circuit C we have $Q_1x_1Q_2x_2 \ldots Q_nx_nf_C(x_1, \ldots, x_n) = 1$ iff the "backward path" from the output gate either ends in an input gate with an existentially quantified variable or it ends in a c_1-gate and has even length. This can be checked in logarithmic space.

2. Let $B \not\subseteq \mathrm{N}$ and ($B \subseteq \mathrm{V}$ or $B \subseteq \mathrm{E}$). By Proposition 20 we have $\mathbf{\Sigma}_k(B) \equiv_{\mathrm{m}}^{\log} \mathbf{\Pi}_k(B) \equiv_{\mathrm{m}}^{\log} \mathbf{QBC}(B) \equiv_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$, and by Theorem 12 we obtain that these problems are $\leq_{\mathrm{m}}^{\log}$-complete for NL.

3. Let $B \not\subseteq \mathrm{N}$ and $B \subseteq \mathrm{L}$. An inspection of Figure 1 yields $\mathrm{L}_2 \subseteq [\mathrm{B}] \subseteq \mathrm{L}$. Define $h(x,y,z) =_{\mathrm{df}} x \oplus y \oplus z$. By Theorem 7 we get $h \in \mathrm{L}_2$ and $\mathrm{L} = [\{\mathrm{aut}, \mathrm{c}_1\}]$, and by Proposition 3 we obtain $\mathbf{\Sigma}_k(\{h\}) \leq_{\mathrm{m}}^{\log} \mathbf{\Sigma}_k(B) \leq_{\mathrm{m}}^{\log} \mathbf{\Sigma}_k(\{\mathrm{aut}, \mathrm{c}_1\})$, $\mathbf{\Pi}_k(\{h\}) \leq_{\mathrm{m}}^{\log} \mathbf{\Pi}_k(B) \leq_{\mathrm{m}}^{\log} \mathbf{\Pi}_k(\{\mathrm{aut}, \mathrm{c}_1\})$, and $\mathbf{QBC}(\{h\}) \leq_{\mathrm{m}}^{\log} \mathbf{QBC}(B)$

$\leq_{\mathrm{m}}^{\log}$ $\mathbf{QBC}(\{\mathrm{aut}, \mathrm{c}_1\})$.

Upper bound. For a $\{\mathrm{aut}, \mathrm{c}_1\}$-circuit C with input variables $x_1, \ldots, x_n$ let m_0 be the number of paths from the output gate to a constant c_1-gate, and for $i = 1, \ldots, k$, let m_i be the number of paths from the output gate to an x_i-gate. Obviously, x_i is fictive in f_C iff m_i is even. We conclude

$Q_1 x_1 \ldots Q_n x_n f_C(x_1, \ldots, x_n) = 1 \Leftrightarrow$

$\Leftrightarrow$ either $f_C(0^n) = 1$ and $x_1, \ldots, x_n$ are fictive
or there exists a non-fictive x_i such that $Q_i = \exists$ and $x_{i+1}, \ldots, x_n$ are fictive.

$\Leftrightarrow$ either m_0 is odd and $m_1, \ldots, m_n$ are even
or there exists an i such that $Q_i = \exists$, m_i is odd and $m_{i+1}, \ldots, m_n$ are even.

The latter can be checked by a $\mathrm{L}^{\oplus\mathrm{L}}$ computation. By Theorem 1 we have $\mathrm{L}^{\oplus\mathrm{L}} = \oplus\mathrm{L}$.

Lower Bound. By Theorem 2 it is sufficient to prove $\mathrm{GOAP} \leq_{\mathrm{m}}^{\log} \mathbf{\Pi}_2(\{h\})$ and $\overline{\mathrm{GOAP}} \leq_{\mathrm{m}}^{\log} \mathbf{\Sigma}_2(\{h\})$. Given a directed acyclic graph G whose vertices have outdegree 0 or 2, a start vertex s, and a target vertex t with outdegree 0. We construct in logarithmic space a $\{h\}$-circuit C with input variables x, y and z as follows: Every outdegree 2 vertex becomes a h-gate where the additional incoming edge comes from a z-gate. The vertex t becomes an x-gate, and every other outdegree 0 vertex becomes a z-gate. It is not hard to see that $f_C(x,z) = g(z)$ for some boolean function g if the number of paths in G from s to t is even and $f_C(x,z) = x \oplus g(z)$ otherwise. Construct a $\{h\}$-circuit C' such that $f_{C'}(x,y,z) = f_C(x,z) \oplus f_C(y,z) \oplus z$. Now, if the number of paths in G from s to t is even then $f_{C'}(x,y,z) = z$ and consequently $\exists z \forall x \forall y C' \in \mathbf{\Sigma}_2(\{h\})$ and $\forall z \exists x \exists y C' \notin \mathbf{\Pi}_2(\{h\})$. On the other hand, if the number of paths in G from s to t is odd then $f_{C'}(x,y,z) = x \oplus y \oplus z$ and consequently $\forall z \exists x \exists y C' \in \mathbf{\Pi}_2(\{h\})$ and $\exists z \forall x \forall y C' \notin \mathbf{\Sigma}_2(\{h\})$.

4. Let $B \not\subseteq \mathrm{V}$, $B \not\subseteq \mathrm{E}$, and $B \subseteq \mathrm{M}$. By Proposition 20 we have $\mathbf{\Sigma}_k(B) \equiv_{\mathrm{m}}^{\log} \mathbf{\Pi}_k(B) \equiv_{\mathrm{m}}^{\log} \mathbf{QBC}(B)$ $\equiv_{\mathrm{m}}^{\log} \mathbf{VAL}(B)$, and by Theorem 12 we obtain that these problems are $\leq_{\mathrm{m}}^{\log}$-complete for P.

5. Let $B \not\subseteq \mathrm{M}$ and $B \not\subseteq \mathrm{L}$. An inspection of Figure 1 yields $\mathrm{S}_{02} \subseteq [B]$, $\mathrm{S}_{12} \subseteq [B]$, or $\mathrm{D}_1 \subseteq [\mathrm{B}]$. Define the functions $h_1(x,u,v,z) =_{\mathrm{df}} (x \vee u) \cdot \overline{v} \cdot z \vee u$, $h_2(x,u,v,z) =_{\mathrm{df}} (x \vee u) \cdot \overline{v} \cdot z \vee u \cdot z$, and $h_3(x,u,v,z) =_{\mathrm{df}} (x \vee u)\overline{v}z \vee u(x \vee \overline{v} \vee z)$. Observe that h_1, h_2, and h_3 are 0-reproducing and 1-reproducing, that h_1 is 0-separating, that h_2 is 1-separating, and that h_3 is self-dual. Hence $h_1 \in \mathrm{R}_0 \cap \mathrm{R}_1 \cap \mathrm{S}_0 = \mathrm{S}_{02}$, $h_2 \in \mathrm{R}_0 \cap \mathrm{R}_1 \cap \mathrm{S}_1 = \mathrm{S}_{12}$, and $h_3 \in \mathrm{R}_0 \cap \mathrm{R}_1 \cap \mathrm{D} = \mathrm{D}_1$. Thus there exists an $h \in [B]$ such that $h(x,u,0,1) = x \vee u$ and $h(x,u,v,z) \leq u$ for all $(v,z) \neq (0,1)$.

Observe that $B \cup \{\mathrm{c}_0, \mathrm{c}_1\}$ is complete. By Theorem 19 it is sufficient to prove $\mathbf{\Sigma}_k(B \cup \{\mathrm{c}_0, \mathrm{c}_1\})$ $\leq_{\mathrm{m}}^{\log}$ $\mathbf{\Sigma}_k(B)$ and $\mathbf{\Pi}_k(B \cup \{\mathrm{c}_0, \mathrm{c}_1\}) \leq_{\mathrm{m}}^{\log} \mathbf{\Pi}_k(B)$ for $k \geq 2$ as well as $\mathbf{QBC}(B \cup \{\mathrm{c}_0, \mathrm{c}_1\}) \leq_{\mathrm{m}}^{\log}$ $\mathbf{QBC}(B)$. For a $(B \cup \{\mathrm{c}_0, \mathrm{c}_1\})$-circuit C we construct a B-circuit C_1 with two new input variables such that $f_C(w,x,y) = f_{C_1}(w,x,y,0,1)$ where w, x, y stand for couples of variables. Now we construct a B-circuit C_2 such that $f_{C_2}(w,x,y,u,v,z) = h(f_{C_1}(w,x,y,v,z), u, v, z)$. Consequently we have $f_{C_2}(w,x,y,u,0,1) = f_C(w,x,y) \vee u$ and $f_{C_2}(w,x,y,u,v,z) \leq u$ for $(v,z) \neq (0,1)$. We observe that for all w,

$$\exists x \forall y f_C(w,x,y) \Leftrightarrow \exists x \forall y \forall u (f_C(w,x,y) \vee u) \Leftrightarrow \exists v \exists z \exists x \forall y \forall u f_{C_2}(w,x,y,u,v,z)$$

and

$$\forall x \exists y f_C(w,x,y) \Leftrightarrow \forall u \forall x \exists y (f_C(w,x,y) \vee u) \Leftrightarrow \forall u \forall x \exists y \exists v \exists z f_{C_2}(w,x,y,u,v,z).$$

This gives the desired reductions. ❑

8 Counting Functions

In this section we study the complexity of functions which count the number of satisfying inputs to a B-circuit. For a B-circuit C define $\#C =_{\mathrm{df}} \#\{a \in \{0,1\}^{\alpha(C)} \mid f_C(a) = 1\}$, and for any finite set B of boolean functions define the *counting function* #(B) by #$(B)(C) =_{\mathrm{df}}$ #C where C is a B-circuit.

It is obvious that #$(B) \in$ #P for every finite set B of boolean functions. The following results on the complexity of #(B) can be found in or easily be derived from the literature.

Theorem 22 *1. [Sim75] If $[B] = \mathrm{BF}$ then #(B) is $\leq_{\mathrm{m}}^{\log}$-complete for #P.*

2. [Sim75] #$(\{\mathrm{et}, \mathrm{vel}, \mathrm{non}\})$ restricted to formulas in conjunctive normal form is $\leq_{\mathrm{m}}^{\log}$-complete for #P.

To study the complexity of #(B) in the general case we will use the following propositions which are analogues to Proposition 3 and Proposition 15 and which are proved in the same way.

Proposition 23 *Let B and B' be finite sets of boolean functions.*

1. *If $B \subseteq \mathrm{BF}[B']$ then #$(B) \leq_{\mathrm{m}}^{\log}$ #(B').*
2. *If $\mathrm{BF}[B'] = \mathrm{BF}[B']$ then #$(B) \equiv_{\mathrm{m}}^{\log}$ #(B').*

Proposition 24 *If B is a finite set of boolean functions containing* et *then #$(B \cup \{\mathrm{c}_1\}) \leq_{\mathrm{m}}^{\log}$ #(B).*

We will use the following obvious duality principle.

Proposition 25 *For any circuit C there holds #$\mathrm{dual}(C) = 2^{\alpha(C)} -$ #C.*

Next we prove a lemma about the representation of #P functions by #(B) functions.

Lemma 26 *Let B be a finite set of boolean functions.*

1. *If $[B] \supseteq \mathrm{S}_1$ then #(B) is $\leq_{\mathrm{m}}^{\log}$-complete for #P, i.e., for every function $f \in$ #P there exists a logspace computable function h generating B-circuits such that $f(x) =$ #$h(x)$.*
2. *If $[B] \supseteq \mathrm{S}_{10}$ or $[B] \supseteq \mathrm{S}_{00}$ then #(B) is $\leq_{1-\mathrm{T}}^{\log}$-complete for #P. More specifically, for every function $f \in$ #P there exist logspace computable functions h_1, h_2 generating B-circuits and logspace computable functions g_1, g_2 such that $f(x) =$ #$h_1(x) - g_1(x) = g_2(x) -$ #$h_2(x)$.*

Proof: 1. Because of $\mathrm{et} \in \mathrm{S}_1$ and by Proposition 24 we obtain #$(B \cup \{\mathrm{c}_1\}) \leq_{\mathrm{m}}^{\log}$ #(B). Observe $[B \cup \{\mathrm{c}_1\}] = \mathrm{BF}$. Now Theorem 22 yields that #(B) is $\leq_{\mathrm{m}}^{\log}$-complete for #P.

2. Let $[B] \supseteq \mathrm{S}_{10}$. Because of $\mathrm{et} \in \mathrm{S}_{10}$ and by Proposition 24 we obtain #$(B \cup \{\mathrm{c}_1\}) \leq_{\mathrm{m}}^{\log}$ #(B). By Figure 1 we have $\{\mathrm{et}, \mathrm{vel}\} \subseteq \mathrm{M}_1 = [\mathrm{S}_{10} \cup \{\mathrm{c}_1\}] \subseteq [\mathrm{B} \cup \{\mathrm{c}_1\}]$. By Proposition 23 it is sufficient to prove the statement for $B = \{\mathrm{et}, \mathrm{vel}\}$, and by Theorem 22.2 it is sufficient to prove #$(\{\mathrm{et}, \mathrm{vel}, \mathrm{non}\})(\mathrm{H}) =$ #$(\{\mathrm{et}, \mathrm{vel}\})(\mathrm{h}_1(\mathrm{H})) - \mathrm{g}_1(\mathrm{H}) = \mathrm{g}_2(\mathrm{H}) -$ #$(\{\mathrm{et}, \mathrm{vel}\})(\mathrm{h}_2(\mathrm{H}))$ for suitable logspace computable functions g_1, g_2, h_1, h_2 and all formulas H in conjunctive normal form. Let $H(x_1, \ldots, x_n)$ be such a 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 arises from H when every $\overline{x_i}$ is replaced with y_i. Defining $H_2 =_{\mathrm{df}} \bigwedge_{i=1}^{n} (\overline{x_i} \leftrightarrow y_i) \wedge H_1$ we obtain $H_2 \equiv \overline{H_3} \wedge H_4$ where $H_3 =_{\mathrm{df}} \bigvee_{i=1}^{n} (x_i \wedge y_i)$ and $H_4 =_{\mathrm{df}} \bigwedge_{i=1}^{n} (x_i \vee y_i) \wedge H_1$. We conclude #$H =$ #$H_2 =$ #$(\overline{H_3} \wedge H_4) =$ #$(H_3 \vee H_4) -$ #$H_3 =$ #$(H_3 \vee H_4) - 2 \cdot (4^n - 3^n)$.

On the other hand, by Theorem 1 we know that for every function $f \in$ #P there exist a function $f' \in$ #P and a logspace computable function g such that $f(x) = g(x) - f'(x)$ for all x. By the above we get a logspace computable function h generating B-circuits and a logspace computable function g' such that $f'(x) =$ #$h(x) - g'(x)$ for all x. Hence $f(x) = (g(x) + g'(x)) -$ #$h(x)$ for all x.

Now let $[B] \supseteq \mathrm{S}_{00}$. By Proposition 8 we obtain $[\mathrm{dual}(B)] = \mathrm{dual}([B]) \supseteq \mathrm{dual}(\mathrm{S}_{00}) = \mathrm{S}_{10}$. By the S_{10}-case we get for every $f \in$ #P logspace computable functions g_1, g_2, h_1, h_2 such that $f(x) =$

$\#h_1(x) - g_1(x) = g_2(x) - \#h_2(x)$ where $h_1(x)$ and $h_2(x)$ are $\mathrm{dual}(B)$-circuits. Now by Propsosition 25 we get $f(x) = (2^{\alpha(h_1(x))} - \#\mathrm{dual}(h_1(x))) - g_1(x) = (2^{\alpha(h_1(x))} - g_1(x)) - \#\mathrm{dual}(h_1(x))$ and $f(x) = g_2(x) - (2^{\alpha(h_2(x))} - \#\mathrm{dual}(h_2(x))) = \#\mathrm{dual}(h_2(x)) - (2^{\alpha(h_2(x))} - g_2(x))$ where $\mathrm{dual}(h_1(x))$ and $\mathrm{dual}(h_2(x))$ are B-circuits. ❑

Now we can prove our main result on the complexity of $\#(B)$.

Theorem 27 *Let be B a finite set of boolean functions.*

$\boxed{\textit{Is } B \subseteq \mathrm{N} \textit{ or } \mathrm{B} \subseteq \mathrm{D}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\#(B) \in \mathrm{FL}}$

$\Downarrow$ *no*

$\boxed{\textit{is } B \subseteq \mathrm{V} \textit{ or } \mathrm{B} \subseteq \mathrm{E}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\#(B) \textit{ is } \leq^{\log}_{1-\mathrm{T}} \textit{-complete for } \mathrm{FL}^{\mathrm{NL}}_{||}[O(\log n)]}$

$\Downarrow$ *no*

$\boxed{\textit{is } B \subseteq \mathrm{L}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\#(B) \textit{ is } \leq^{\log}_{1-\mathrm{T}} \textit{-hard for } \mathrm{FL}^{\oplus\mathrm{L}}_{||}[1], \#(B) \in \mathrm{FL}^{\oplus\mathrm{L}}_{||}[2]}$

$\Downarrow$ *no*

$\boxed{\textit{is } B \subseteq \mathrm{R}_1?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\#(B) \textit{ is } \leq^{\log}_{1-\mathrm{T}} \textit{-complete for } \#\mathrm{P}, \textit{ but not } \leq^{\log}_{\mathrm{m}}\textit{-complete for } \#\mathrm{P}}$

$\Downarrow$ *no*

$\boxed{\textit{is } B \subseteq \mathrm{M}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\#(B) \textit{ is } \leq^{\log}_{1-\mathrm{T}} \textit{-complete for } \#\mathrm{P}. \textit{ If } \#(B) \textit{ is } \leq^{\log}_{\mathrm{m}}\textit{-complete for } \#\mathrm{P} \textit{ then } \mathrm{P} = \mathrm{NP}}$

$\Downarrow$ *no*

$\boxed{\#(B) \textit{ is } \leq^{\log}_{\mathrm{m}}\textit{-complete for } \#\mathrm{P}}$

There exists an algorithm which decides which of the cases above takes place.

Proof: 1. If $B \subseteq \mathrm{N}$ then, by Theorem 7, we obtain $B \subseteq [\{\mathrm{non}, \mathrm{c}_1\}]$ and hence $\#(B) \leq^{\log}_{\mathrm{m}} \#(\{\mathrm{non}, \mathrm{c}_1\})$. Let C be a $\{\mathrm{non}, \mathrm{c}_1\}$-circuit with n input variables. Follow the "backward path" from the output gate as long as an indegree 0 gate is reached. If this is a c_1-gate and the path has even length then $\#C = 2^n$. If this is a c_1-gate and the path has odd length then $\#C = 0$. If this is a input gate then $\#C = 2^{n-1}$. This algorithm works in logspace.

2. If $B \subseteq \mathrm{D}$ and C is a B-circuit with n input variables then $\#C = 2^{n-1}$.

3. Let $B \not\subseteq \mathrm{N}$ and $B \subseteq \mathrm{E}$. By Figure 1 and Theorem 7 we obtain $[\{\mathrm{et}\}] = \mathrm{E}_2 \subseteq [\mathrm{B}] \subseteq \mathrm{E} = [\{\mathrm{et}, \mathrm{c}_0, \mathrm{c}_1\}]$. By Proposition 23 we obtain $\#(\{\mathrm{et}\}) \leq^{\log}_{\mathrm{m}} \#(\mathrm{B}) \leq^{\log}_{\mathrm{m}} \#(\{\mathrm{et}, \mathrm{c}_0, \mathrm{c}_1\})$.

Upper bound. For a $\{\mathrm{et}, \mathrm{c}_0, \mathrm{c}_1\}$-circuit C with input variables $x_1, \ldots, x_n$ we have $f_C(x_1, \ldots, x_n) = \bigwedge_{i \in I_C} x_i$, where $I_C =_{\mathrm{df}} \{i \mid x_i \text{ is non-fictive in } f_C\}$, and hence $\#C = 2^{n-|I_C|}$. It is sufficient to prove that the function $C \to |I_C|$ is in $\mathrm{FL}^{\mathrm{NL}}_{||}[O(\log n)]$. Observe that x_i is non-fictive if $f_C(1^n) = 1$ and $f_C(1^{i-1}01^{n-i}) = 0$. By Theorem 12 the problem of evaluating C is in NL and hence the set $S_C =_{\mathrm{df}} \{(C, i) \mid \text{ the } i\text{-th bit of } |I_C| \text{ is } 1\}$ is in $\mathrm{L}^{\mathrm{NL}} = \mathrm{NL}$ (cf. Theorem 1). Consequently, $|I_C| = c_{S_C}(C, 0) c_{S_C}(C, 1) \ldots\ c_{S_C}(C, m)$ where m is the least number greater or equal to $\log n$.

Lower bound. For a function $f \in FL^{\mathrm{NL}}_{||}[O(\log n)]$ there exist logspace computable functions g_1,g_2 such that $g_2(x) = (z_0, \ldots, z_m)$ and $f(x) = g_1(x, c_{\mathrm{GAP}}(z_m) \ldots c_{\mathrm{GAP}}(z_1) c_{\mathrm{GAP}}(z_0))$ where $m =_{\mathrm{df}} c{\cdot}\log|x|$ for suitable $c > 0$. Consequently, we have $f(x) = g_1(x, \sum_{i=0}^{m} \sum_{j=1}^{2^i} c_{\mathrm{GAP}}(z_i))$, and hence there exists a logspace computable function g_3 such that $g_3(x) = (G_1, G_2, \ldots, G_k)$ and $f(x) = g_1(x, \sum_{i=1}^{k} c_{\mathrm{GAP}}(G_i))$.

So it is sufficient to prove that the function which maps $(G_1, G_2, \ldots, G_k)$ to $\sum_{i=1}^{k} c_{\mathrm{GAP}}(G_i)$ is $\leq^{\log}_{1-\mathrm{T}}$-reducible to #$(\{\mathrm{et}\})$. From every G_i construct a $\{\mathrm{et}\}$-circuit C_i with input variables u_i and z as follows. Every outdegree 2 vertex of G_i becomes an et-gate, the start vertex of G_i becomes the output gate of C_i, the target vertex of G_i becomes a u_i-gate, and every other outdegree 0 vertex of G_i becomes an z-gate. Obviously, if $G_i \in \mathrm{GAP}$ then $f_{C_i}(u_i, z) = u_i \wedge z$, otherwise $f_{C_i}(u_i, z) = z$. Now construct a $\{\mathrm{et}\}$-circuit C in such a way that $f_C(u_1, \ldots, u_k, z) = \bigwedge_{i=1}^{k} f_{C_i}(u_i, z)$. Consequently, f_C has $\sum_{i=1}^{k} c_{\mathrm{GAP}}(G_i) + 1$ non-fictive variables and hence $\#(\{\mathrm{et}\})(\mathrm{C}) = 2^{\mathrm{k} - \sum_{\mathrm{i}=1}^{\mathrm{k}} c_{\mathrm{GAP}}(\mathrm{G_i})}$, i.e., $\sum_{i=1}^{k} c_{\mathrm{GAP}}(G_i) = k - \log_2 \#(\{\mathrm{et}\})(\mathrm{C})$.

4. The case $B \not\subseteq \mathrm{N}$ and $B \subseteq \mathrm{V}$ is done in an anologous manner.

5. Let $B \not\subseteq \mathrm{N}$, $B \not\subseteq \mathrm{D}$, and $B \subseteq \mathrm{L}$. A look at Figure 1 shows that $\mathrm{L}_0 \subseteq [\mathrm{B}] \subseteq \mathrm{L}$ or $\mathrm{L}_1 \subseteq [\mathrm{B}] \subseteq \mathrm{L}$.
Let $\mathrm{L}_0 \subseteq [\mathrm{B}] \subseteq \mathrm{L}$. By Theorem 7 we obtain $[\{\mathrm{aut}\}] \subseteq [\mathrm{B}] \subseteq [\{\mathrm{aut}, \mathrm{c}_1\}]$, and by Proposition 23 we obtain $\#(\{\mathrm{aut}\}) \leq^{\log}_{\mathrm{m}} \#(\mathrm{B}) \leq^{\log}_{\mathrm{m}} \#(\{\mathrm{aut}, \mathrm{c}_1\})$.
Upper bound. Let C be an $\{\mathrm{aut}, \mathrm{c}_1\}$-circuit with n input variables. There are three different cases. If f_C has a non-fictive variable then $\#C = 2^{n-1}$. Otherwise f_C is a constant. If $f_C = 0$ ($f_C = 1$) then $\#C = 0$ ($\#C = 2^n$, resp.) Which of these cases takes place can be found out by the two queries "do there exists an i such that $f_C(0^n) \neq f_C(0^{i-1}10^{n-i})$?" and "$f_C(0^n) = 1$?". However, by Theorem 12 these are $\oplus\mathrm{L}$ queries (notice that $\mathrm{L}^{\oplus\mathrm{L}} = \oplus\mathrm{L}$ by Theorem 1).
Lower bound. It is sufficient to prove that there exist logspace computable functions g_1, g_2 such that $c_{\mathrm{GOAP}}(G) = g_1(\#(\{\mathrm{aut}\})(\mathrm{g}_2(\mathrm{G})))$ where c_{GOAP} is the characteristic function of GOAP (cf. Theorem 2). Given a directed acyclic graph G whose vertices have outdegree 0 or 2, a start vertex s, and a target vertex$t$ with outdegree 0. We construct a $\{\mathrm{aut}\}$-circuit C with two input variables x and z as follows: Every outdegree 2 vertex becomes an aut-gate, the vertex t becomes an aut-gate with incoming edges from an x-gate and a z-gate, and every outdegree 0 vertex becomes an aut-gate with both incoming edges from a z-gate. The vertex s becomes the output gate of C. If there is an odd number of paths from s to t in G then $f(x, z) = x \oplus z$ and hence $\#C = 2$. Otherwise $f(x, z) = 0$ and hence $\#C = 0$. Consequently, $c_{\mathrm{GOAP}}(G) = \frac{1}{2} \cdot \#(\{\mathrm{aut}\})(\mathrm{C})$.
Now let $\mathrm{L}_1 \subseteq [\mathrm{B}] \subseteq \mathrm{L}$. By Proposition 8 we obtain $\mathrm{L}_0 = \mathrm{dual}(\mathrm{L}_1) \subseteq \mathrm{dual}([\mathrm{B}]) = [\mathrm{dual}(\mathrm{B})] \subseteq \mathrm{dual}(\mathrm{L}) = \mathrm{L}$, and by the above we get that $\#(\mathrm{dual}(B))$ is $\leq^{\log}_{1-\mathrm{T}}$-hard for $\mathrm{FL}^{\oplus\mathrm{L}}_{||}[1]$. With Proposition 25 we conclude that $\#(B)$ is $\leq^{\log}_{1-\mathrm{T}}$-hard for $\mathrm{FL}^{\oplus\mathrm{L}}_{||}[1]$.

6. Let $B \not\subseteq \mathrm{D}$, $B \not\subseteq \mathrm{V}$, $B \not\subseteq \mathrm{E}$, $B \not\subseteq \mathrm{L}$, and $B \subseteq \mathrm{R}_1$. An inspection of Figure 1 shows that $[B] \supseteq \mathrm{S}_{10}$ or $[B] \supseteq \mathrm{S}_{00}$. By Lemma 26 we obtain that $\#(B)$ is $\leq^{\log}_{1-\mathrm{T}}$-complete for #P. Because of $B \subseteq \mathrm{R}_1$ we have $\#(B)(C) > 0$ for every B-circuit C. Hence $\#(B)$ cannot be $\leq^{\log}_{\mathrm{m}}$-complete for #P.

7. Let $B \not\subseteq \mathrm{D}$, $B \not\subseteq \mathrm{V}$, $B \not\subseteq \mathrm{E}$, $B \not\subseteq \mathrm{L}$, and $B \subseteq \mathrm{M}$. As above we obtain that $\#(B)$ is $\leq^{\log}_{1-\mathrm{T}}$-complete for #P. Assume that $\#(B)$ is $\leq^{\log}_{\mathrm{m}}$-complete for #P. For an arbitrary $A \in \mathrm{NP}$ there exists an $f \in \#\mathrm{P}$ such that $x \in A \Leftrightarrow f(x) > 0$ for every x. Since $\#(B)$ is $\leq^{\log}_{\mathrm{m}}$-complete for #P there exists a logspace computable function h such that $f(x) = \#(B)(h(x))$ for every x. Consequently, $x \in A \Leftrightarrow \#(B)(h(x)) > 0 \Leftrightarrow h(x) \in \mathbf{SAT}(B)$. Because of $B \subseteq \mathrm{M}$ we obtain $A \in \mathrm{P}$ by Theorem 16.

8. Let $B \not\subseteq \mathrm{D}$, $B \not\subseteq \mathrm{L}$, $B \not\subseteq \mathrm{R}_1$, and $B \not\subseteq \mathrm{M}$. An inspection of Figure 1 shows that $[B] \supseteq \mathrm{S}_1$. By Lemma 26 we obtain that $\#(B)$ is $\leq^{\log}_{\mathrm{m}}$-complete for #P. ❑

9 The Threshold Problem

In this section we will study the complexity of the *threshold problem*

$$\mathbf{THR}(B) =_{\mathrm{df}} \{(C, k) \mid C \text{ is a } B\text{-circuit, } k \geq 0, \text{ and } \#C \geq k\}$$

for every finite set B of boolean functions. Obviously, $\mathbf{THR}(B) \in \mathrm{PP}$.

The following result on the complexity of $\mathbf{THR}(B)$ can easily be derived from the literature where it is proved for the special case of $\{\mathrm{et}, \mathrm{vel}, \mathrm{non}\}$-formulae.

Theorem 28 *[Gil77] For a finite set B of boolean functions, if $[B] = \mathrm{BF}$ then $\mathbf{THR}(B)$ is even $\leq_{\mathrm{m}}^{\log}$-complete for* PP.

Now we are ready to prove the main theorem on the complexity of $\mathbf{THR}(B)$.

Theorem 29 *Let be B a finite set of boolean functions.*

$\boxed{\textit{Is } B \subseteq \mathrm{N} \textit{ or } \mathrm{B} \subseteq \mathrm{D}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\boxed{\mathbf{THR}(B) \in \mathrm{L}}}$

$\Downarrow$ *no*

$\boxed{\textit{is } B \subseteq \mathrm{V} \textit{ or } \mathrm{B} \subseteq \mathrm{E}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\boxed{\mathbf{THR}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \mathrm{NL}}}$

$\Downarrow$ *no*

$\boxed{\textit{is } B \subseteq \mathrm{L}?}$ $\overset{\textit{yes}}{\Longrightarrow}$ $\boxed{\boxed{\mathbf{THR}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \oplus\mathrm{L}}}$

$\Downarrow$ *no*

$\boxed{\boxed{\mathbf{THR}(B) \textit{ is } \leq_{\mathrm{m}}^{\log}\textit{-complete for } \mathrm{PP}}}$

There exists an algorithm which decides which of the cases above takes place.

Proof: 1. If $B \subseteq \mathrm{N}$ or $B \subseteq \mathrm{D}$ then, by Theorem 27, we have $\#(B) \in \mathrm{FL}$ and consequently $\mathbf{THR}(B) \in \mathrm{L}$.

2. Let $B \not\subseteq \mathrm{N}$ and $B \subseteq \mathrm{E}$. By Theorem 27 we obtain $\#(B) \in \mathrm{FL}_{||}^{\mathrm{NL}}$ and hence $\mathbf{THR}(B) \in \mathrm{L}^{\mathrm{NL}} = \mathrm{NL}$ (cf. Theorem 1).

For the lower bound it is sufficient to prove $\overline{\mathrm{GAP}} \leq_{\mathrm{m}}^{\log} \mathbf{THR}(\{\mathrm{et}\})$ (cf. Theorem 2). As in Item 3 of the proof of Theorem 27 we construct for a graph G an $\{\mathrm{et}\}$-circuit C with the following property: If $G \in \mathrm{GAP}$ then $f_C(u, z) = u \wedge z$, and $f_C(u, z) = z$ otherwise. Hence, if $G \in \overline{\mathrm{GAP}}$ then $\#C = 2$ and $(C, 2) \in \mathbf{THR}(\{\mathrm{et}\})$. Otherwise $\#C = 1$ and $(C, 2) \notin \mathbf{THR}(\{\mathrm{et}\})$

3. The case $B \not\subseteq \mathrm{N}$ and $B \subseteq \mathrm{V}$ is done in an analogous manner.

4. The case $B \not\subseteq \mathrm{N}$, $B \not\subseteq \mathrm{D}$, and $B \subseteq \mathrm{L}$ is also done analogously using the construction in Item 5 of the proof of Theorem 27 and the problen GOAP.

5. Let $B \not\subseteq \mathrm{D}$, $B \not\subseteq \mathrm{V}$, $B \not\subseteq \mathrm{E}$, and $B \not\subseteq \mathrm{L}$. An inspection of Figure 1 shows that $[B] \supseteq \mathrm{S}_{10}$ or $[B] \supseteq \mathrm{S}_{00}$ For a set $A \in \mathrm{PP}$ there exists a $f \in \#\mathrm{P}$ such that $(x, k) \in A \Leftrightarrow f(x) \geq k$. By Lemma 26 there exist logspace computable functions g, h such that $f(x) = \#(B)(h(x)) - g(x)$. Consequently, $(x, k) \in A \Leftrightarrow \#(B)(h(x)) \geq g(x) + k \Leftrightarrow (h(x), g(x) + k) \in \mathbf{THR}(B)$. ❑

10 Conclusions

We have investigated the complexity of circuit-based combinatorial problems defined by circuits with gates from an arbitrary finite base of boolean functions. What about the special case of boolean formulae, i.e., "tree-like" boolean circuits? Informally, all completeness results for complexity classes beyond P are the same as in the general case of circuits. Hopwever, different proofs are needed, because Proposion 3 does not hold in the formula case. The completeness results for complexity classes inside P do not remain valid in many cases because evaluating a formula is much easier than evaluating a circuit.

References

[BDG90a] J.L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*. Springer-Verlag, first edition, 1990.

[BDG90b] J.L. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity II*. Springer-Verlag, first edition, 1990.

[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.

[Coo71] S. A. Cook. The complexity if theorem proving procedures. In *Proc. 3rd Ann. ACM Symp. on Theory of Computation*, pages 151–158, 1971.

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

[Gil77] J.T. Gill. Computational complexity of probabilistic turing machines. *SIAM Journal of Computation*, 6:675–695, 1977.

[Gol77] L. M. Goldschlager. The monotone and planar circuit value problems are log-space complete for P. *SIGACT News*, 9:25–29, 1977.

[GP86] L.M. Goldschlager and I. Parberry. On the construction of parallel computers from various bases of boolean functions. *Theoretical Computer Science*, 43:43–58, 1986.

[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.

[KSW97] S. Khanna, M. Sudan, and D. Williamson. A complete classification of the approximability of maximization problems derived from boolean constraint satisfaction. In *Proceedings 29th Symposium on Theory of Computing*, pages 11–20. ACM Press, 1997.

[Lad77] R. E. Ladner. The circuit value problem is logspace complete for P. *SIGACT News*, pages 18–20, 1977.

[Lew78] H. R. Lewis. Satisfiability problems for propositional calculi. unpublished manuscript, 1978.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.

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

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

[Sch78] Th.J. Schaefer. The complexity of satisfiability problems. In *Proc. 10th ACM STOC*, pages 216–226. Association for Computing Machinery, 1978.

[Sim75] J. Simon. On some central problems in computational complexity. Doctoral Thesis, Dept. of Computer Science, Cornell University, 1975.

[SM73] L. J. Stockmeyer and A. R. Meyer. Word problems requiring exponential time. In *Proc. 35th Ann. ACM Symp. on Theory of Computation*, pages 1–9, 1973.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity*. Springer, 1999.

[Wag94] K.W. Wagner. *Einführung in die Theoretische Informatik*. Springer Verlag, 1994.