INSTITUTE OF MATHEMATICS

THE CZECH ACADEMY OF SCIENCES

On the complexity of the clone membership problem

Emil Jeřábek

Preprint No. 55-2019

PRAHA 2019

On the complexity of the clone membership problem

Emil Jeřábek*

The Czech Academy of Sciences, Institute of Mathematics
Žitná 25, 115 67 Praha 1, Czech Republic, email: jerabek@math.cas.cz

September 26, 2019

Abstract

We investigate the complexity of the Boolean clone membership problem (CMP): given a set of Boolean functions F and a Boolean function f, determine if f is in the clone generated by F, i.e., if it can be expressed by a circuit with F-gates. Here, f and elements of F are given as circuits or formulas over the usual De Morgan basis. Böhler and Schnoor [3] proved that for any fixed F, the problem is **coNP**-complete, with a few exceptions where it is in **P**. Vollmer [17] incorrectly claimed that the full problem CMP is also **coNP**-complete. We prove that CMP is in fact $\mathbf{\Theta}_2^{\mathbf{P}}$-complete, and we complement Böhler and Schnoor's results by showing that for fixed f, the problem is **NP**-complete unless f is a projection.

More generally, we study the problem B-CMP where F and f are given by circuits using gates from B. For most choices of B, we classify the complexity of B-CMP as being $\mathbf{\Theta}_2^{\mathbf{P}}$-complete (possibly under randomized reductions), **coDP**-complete, or in **P**.

1 Introduction

The clone membership problem, asking whether a given function is expressible by means of a given list of initial functions, is a basic problem in universal algebra, where it can be phrased as whether a given operation is term-definable in a given algebra, in logic, where we ask if a given truth function (in a possibly multi-valued logic) is definable by a formula over a given set of connectives, and in computer science, where we ask if a given function is computable by a circuit over a given basis of gates.

From the point of view of computational complexity, several variants of the problem were studied in the literature. The most straightforward representation of the input functions is by tables of values. In this setting, Kozen [9] proved that the membership problem for clones of *unary* functions on arbitrary finite domains is **PSPACE**-complete. The general clone membership problem for arbitrary functions on finite domains is **EXP**-complete. This result is credited in [2] to an unpublished manuscript of H. Friedman, but the first published proof is due to Bergman, Juedes, and Slutzki [1]; a mistake in their paper was corrected by Mašulović [13]. As

*Supported by grant 19-05497S of GA ČR. The Institute of Mathematics of the Czech Academy of Sciences is supported by RVO: 67985840.

shown by Kozik [10], there even exists a *fixed* finitely generated clone on a finite domain whose membership problem is **EXP**-complete.

The high complexity of the problem on arbitrary finite domains is related to the complicated structure of clones on domains of size ≥ 3. In contrast, the lattice of clones on the *Boolean* (two-element) domain is quite simple, and it has been explicitly described by Post [14]. As a result, the Boolean clone membership problem is computationally much easier than the general case: it was shown to be in **NL** by Bergman and Slutzki [2], while Vollmer [17] proved that it was in quasipolynomial $\mathbf{AC}^0$, which implies it is not **NL**-hard (or even $\mathbf{AC}^0[2]$-hard).

The representation of functions by tables is quite inefficient, as it always has size exponential in the number of variables. A viable alternative, especially in the Boolean case, is to represent functions by expressions (circuits or formulas) over some canonically chosen functionally complete basis, say, the De Morgan basis $\{\wedge, \vee, \neg\}$. In this setting, Böhler and Schnoor [3] studied the complexity of membership problems for *fixed* Boolean clones C: they proved that all such problems are **coNP**-complete with a few exceptions that are in **P**. More generally, they studied variants of the problem where f is not given by a circuit over a functionally complete basis, but over an arbitrary (but fixed) basis. They classified most such problems as being **coNP**-complete or in **P**.

The full Boolean clone membership problem in the circuit representation (denoted CMP in this paper) was considered by Vollmer [17], who claimed it was also **coNP**-complete. However, he did not provide much in the way of proof for the **coNP** upper bound,[1] and as we will see shortly, this claim is wrong.

A characterization of clone membership in terms of preservation of relational invariants easily implies that CMP is computable in $\mathbf{P}^{\mathbf{NP}}$—more precisely, in the class $\mathbf{\Theta}_2^{\mathbf{P}} = \mathbf{P}^{\mathbf{NP}[\log]} = \mathbf{P}^{\|\mathbf{NP}}$. The main goal of this paper is to prove that CMP is in fact $\mathbf{\Theta}_2^{\mathbf{P}}$-complete. As a warm-up, we consider a restriction of CMP dual to Böhler and Schnoor's results: we prove that for a fixed target Boolean function f, the clone membership problem is **NP**-complete in all nontrivial cases (i.e., unless f is a projection function, or a nullary function if we allow them). This already shows that CMP cannot be **coNP**-complete unless $\mathbf{NP} = \mathbf{coNP}$. We then go on to prove that CMP is $\mathbf{\Theta}_2^{\mathbf{P}}$-complete; our main technical tool is a characterization of clones generated by threshold functions. We also discuss some variants of our results, such as using formulas instead of circuits for representation of functions, or allowing nullary functions.

In the second part of the paper, we investigate the complexity of restricted versions of CMP, denoted B-CMP, where the input functions are given by circuits or formulas over an arbitrary (but fixed) finite basis B instead of the De Morgan basis. We show that B-CMP remains $\mathbf{\Theta}_2^{\mathbf{P}}$-complete, albeit using randomized reductions, when the clone $[B]$ generated by B has infinitely many subclones, and includes some non-monotone functions; we rely on a randomized construction of formulas for threshold functions using fixed threshold functions as gates, following the method of Valiant [16]. On the other hand, if $[B]$ has only finitely many subclones, we classify the complexity of B-CMP as either **coDP**-complete or in **P**. The complexity of B-CMP remains open when $[B]$ has infinitely many subclones, but consists of monotone functions only.

[1] Essentially just stating that it follows from a criterion similar to our Corollary 3.2 below, despite that it involves both positive and negative occurrences of the **coNP** preservation relation.

2 Preliminaries

We will assume basic familiarity with the theory of Boolean clones; this is described in many places, for example Lau [11]. We will summarize the most important points below to fix our terminology and notation.

Let $\mathbf{2} = \{0, 1\}$. An *n-ary Boolean function* (or *operation*) is a mapping $f\colon \mathbf{2}^n \to \mathbf{2}$. We denote the set of n-ary Boolean functions by Op_n, and the set of all Boolean functions by $\mathrm{Op} = \bigcup_{n\geq 1} \mathrm{Op}_n$. (Following the tradition in literature on Boolean clones, we disallow nullary functions; we will comment later on how this affects our results.)

We will use common connectives such as $\wedge, \vee, \to, \neg$ to denote specific Boolean functions; by abuse of notation, 0 and 1 denote the constant functions of arbitrary arity. If $0 \leq i < n$, the *projection function* $\pi_i^n \in \mathrm{Op}_n$ is defined by $\pi_i^n(x_0, \ldots, x_{n-1}) = x_i$. For any $n > 0$ and $0 \leq t \leq n+1$, the *threshold function* $\theta_t^n \in \mathrm{Op}_n$ is defined by

$$\theta_k^n(x_0, \ldots, x_{n-1}) = 1 \iff \big|\{i < n : x_i = 1\}\big| \geq t.$$

Notice that $\theta_0^n = 1$, $\theta_{n+1}^n = 0$, $\theta_1^2 = \vee$, $\theta_2^2 = \wedge$, and $\theta_1^1 = \pi_1^1$ is the identity function. We recall that threshold functions have uniformly constructible polynomial-size circuits, and indeed, uniformly constructible $O(\log n)$-depth formulas.

Given functions $f \in \mathrm{Op}_n$ and $g_0, \ldots, g_{n-1} \in \mathrm{Op}_m$, their *composition* is the function $h \in \mathrm{Op}_m$ defined by

$$h(\vec{x}) = f\big(g_0(\vec{x}), \ldots, g_{n-1}(\vec{x})\big).$$

A set of Boolean functions $C \subseteq \mathrm{Op}$ is a *clone* if it contains all projections and is closed under composition. The intersection of an arbitrary collection of clones is again a clone (where the empty intersection is understood as Op), thus the poset of clones under inclusion forms a complete lattice, and it yields an (algebraic) closure operator $[-]\colon \mathcal{P}(\mathrm{Op}) \to \mathcal{P}(\mathrm{Op})$; that is, for any $F \subseteq \mathrm{Op}$, $[F]$ denotes the clone generated by F.

The *Boolean clone membership problem* CMP is the following decision problem:

> **Input:** A finite set of functions $F \subseteq \mathrm{Op}$ and a function $f \in \mathrm{Op}$, all given by Boolean circuits over the De Morgan basis $\{\wedge, \vee, \neg\}$.
>
> **Output:** YES if $f \in [F]$, otherwise NO.

For any clone $C = [C]$, the membership problem CMP_C is the special case of CMP where F is fixed:

> **Input:** A function $f \in \mathrm{Op}$, given by a Boolean circuit over the De Morgan basis.
>
> **Output:** YES if $f \in C$, otherwise NO.

Dually, for a fixed function $f \in \mathrm{Op}$, CMP^f denotes the following special case of CMP:

> **Input:** A finite set of functions $F \subseteq \mathrm{Op}$, given by Boolean circuits over the De Morgan basis.
>
> **Output:** YES if $f \in [F]$, otherwise NO.

Notice that $\mathrm{CMP}^f = \mathrm{CMP}^g$ whenever $[f] = [g]$, as the output condition can be stated as $[f] \subseteq [F]$.

A *k-ary Boolean relation* is $r \subseteq \mathbf{2}^k$. The set of k-ary Boolean relations is denoted by Rel_k, and the set of all relations by $\mathrm{Rel} = \bigcup_k \mathrm{Rel}_k$. (Here, it will not make any difference if we allow nullary relations or not; the reader is welcome to make the choice.) A function $f \in \mathrm{Op}_n$ *preserves* a relation $r \in \mathrm{Rel}_k$, written as $f \triangleright r$, if f, considered as a mapping of the relational structures $\langle \mathbf{2}, r\rangle \times \cdots \times \langle \mathbf{2}, r\rangle \to \langle \mathbf{2}, r\rangle$, is a homomorphism. Explicitly, $f \triangleright r$ iff the following implication holds for every matrix $(a_i^j)_{i<n}^{j<k} \in \mathbf{2}^{k\times n}$:

$$\big\{\langle a_i^0, \dots, a_i^{k-1}\rangle : i < n\big\} \subseteq r \implies \langle f(a_0^0, \dots, a_{n-1}^0), \dots, f(a_0^{k-1}, \dots, a_{n-1}^{k-1})\rangle \in r.$$

If $F \subseteq \mathrm{Op}$ and $R \subseteq \mathrm{Rel}$, we write $F \triangleright R$ if $f \triangleright r$ for all $f \in F$ and $r \in R$. The set of *invariants* of $F \subseteq \mathrm{Op}$ and the set of *polymorphisms* of $R \subseteq \mathrm{Rel}$ are defined by

$$\mathrm{Inv}(F) = \{r \in \mathrm{Rel} : F \triangleright r\},$$
$$\mathrm{Pol}(R) = \{f \in \mathrm{Op} : f \triangleright R\}.$$

The mappings $\mathrm{Inv}\colon \mathcal{P}(\mathrm{Op}) \to \mathcal{P}(\mathrm{Rel})$ and $\mathrm{Pol}\colon \mathcal{P}(\mathrm{Rel}) \to \mathcal{P}(\mathrm{Op})$ form an (antitone) Galois connection. The Galois-closed subsets of Op are exactly the clones: that is, $\mathrm{Pol}(R)$ is a clone for any $R \subseteq \mathrm{Rel}$, each clone can be described as $\mathrm{Pol}(R)$ for some $R \subseteq \mathrm{Rel}$, and $[F] = \mathrm{Pol}(\mathrm{Inv}(F))$ for any $F \subseteq \mathrm{Op}$.

If we allowed nullary functions, then Galois-closed subsets of Rel would be exactly the *coclones*: subsets $R \subseteq \mathrm{Rel}$ that are closed under definitions by *primitive positive* formulas, i.e., first-order formulas $\varphi(\vec{x})$ of the form $\exists \vec{y} \bigwedge_{i<k} \psi_i(\vec{x}, \vec{y})$, where each ψ_i is an atomic formula (an instance of a relation $r \in R$, or of equality). Under our restriction to non-nullary functions, Galois-closed subsets of Rel are only the coclones that contain the empty relation $\varnothing \in \mathrm{Rel}_1$.

The lattice of Boolean clones was completely described by Post [14]. In particular, we will make use of the following characterization, fixing our naming of basic clones and their invariants along the way. Here, for any $f \in \mathrm{Op}_n$, $\mathrm{gr}(f) = \{\langle \vec{x}, y\rangle \in \mathbf{2}^{n+1} : y = f(\vec{x})\} \in \mathrm{Rel}_{n+1}$ denotes the graph of f.

Fact 2.1 *Every Boolean clone is an intersection of a family of completely meet-irreducible clones, which are:*

- *The clone* $\mathrm{M} = [\wedge, \vee, 0, 1] = \mathrm{Pol}(\leq)$ *of* monotone *functions, where* $\leq$ *denotes the relation* $\{\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,1\rangle\} \in \mathrm{Rel}_2$.
- *The clone* $\mathrm{A} = [+, 1] = \mathrm{Pol}(r_A)$ *of* affine *functions, where* $+$ *denotes addition in* $\mathbb{F}_2$*, and* $r_A = \{\langle x, y, z, w\rangle : x + y + z + w = 0\} \in \mathrm{Rel}_4$.
- *The clone* $\mathrm{D} = [\theta_2^3, \neg] = \mathrm{Pol}\big(\mathrm{gr}(\neg)\big)$ *of* self-dual *functions.*
- *The clone* $\bigwedge = [\wedge, 0, 1] = \mathrm{Pol}\big(\mathrm{gr}(\wedge)\big)$ *of* conjunctive *functions.*
- *The clone* $\bigvee = [\vee, 0, 1] = \mathrm{Pol}\big(\mathrm{gr}(\vee)\big)$ *of* disjunctive *functions.*

- *The clone* $\mathrm{U} = [\neg, 0] = \mathrm{Pol}(r_U)$ *of* essentially unary *functions, where* $r_U = \big\{\langle x, y, z\rangle : z \in \{x, y\}\big\} \in \mathrm{Rel}_3$.
- *For each* $m \geq 1$, *the clones* $\mathrm{T}_1^m = [\theta_2^{m+1}, \to] = \mathrm{Pol}(r_1^m)$ *and* $\mathrm{T}_0^m = [\theta_m^{m+1}, \nrightarrow] = \mathrm{Pol}(r_0^m)$, *where* $x \nrightarrow y = \neg(x \to y) = x \wedge \neg y$, *and the relations* $r_\alpha^m \in \mathrm{Rel}_m$ *are defined by*

$$r_1^m = \{\vec{x} \in \mathbf{2}^m : x_0 \vee \cdots \vee x_{m-1} = 1\},$$
$$r_0^m = \{\vec{x} \in \mathbf{2}^m : x_0 \wedge \cdots \wedge x_{m-1} = 0\}.$$

  Since $r_\alpha^1 = \{\alpha\}$, *this includes as a special case the clones* $\mathrm{P}_1 = [\wedge, \to] = \mathrm{T}_1^1$ *and* $\mathrm{P}_0 = [\vee, \nrightarrow] = \mathrm{T}_0^1$ *of* 1-preserving *and* 0-preserving *functions* (*respectively*).

We will denote intersection of named clones by juxtaposition, so that, e.g., $\mathrm{AD} = \mathrm{A} \cap \mathrm{D}$. For convenience, we also put $\mathrm{P} = \mathrm{P}_0\mathrm{P}_1$ and $\mathrm{T}_\alpha^\infty = \bigcap_m \mathrm{T}_\alpha^m$. We have $\mathrm{T}_1^\infty = [\to]$ and $\mathrm{T}_0^\infty = [\nrightarrow]$. We will denote the top and bottom of the lattice of clones by $\top$ and $\bot$, i.e., $\top = \mathrm{Op}$, and $\bot = \{\pi_i^n : i < n\} = \mathrm{UP}$. We define

$$R_n = \{\leq, r_A, \mathrm{gr}(\neg), \mathrm{gr}(\wedge), \mathrm{gr}(\vee), r_U\} \cup \{r_\alpha^m : 0 < m \leq n, \alpha \in \mathbf{2}\} \subseteq \mathrm{Rel}$$

for each $n \geq 0$, and $R_\infty = \bigcup_n R_n$.

The Hasse diagram of the lattice of Boolean clones (called *Post's lattice*) is depicted in Fig. 1. (In fact, Post [14] did not work with the modern definition of clones, but with a slightly weaker concept of *iterative classes*, which do not necessarily contain all projections. Thus, his original lattice has four more classes.)

We assume the reader is familiar with basic notions of complexity theory, including the classes **P**, **NP**, and **coNP**. The class $\mathbf{\Theta}_2^\mathbf{P}$, introduced by Wagner [18], is defined as $\mathbf{P}^{\mathbf{NP}[\log]}$: the class of languages computable in polynomial time using $O(\log n)$ queries to an **NP** oracle. It has several other equivalent characterizations, see Buss and Hay [5]. In particular, $\mathbf{\Theta}_2^\mathbf{P} = \mathbf{P}^{\|\mathbf{NP}}$ (languages computable in polynomial time with non-adaptive access to an **NP** oracle).

If C is a complexity class, we say that a language L is C*-hard* if for every $L' \in C$, there exists a many-one (uniform) $\mathbf{TC}^0$ reduction from L' to L; if, moreover, $L \in C$, then L is C*-complete*. We chose $\mathbf{TC}^0$ reductions because they strike the right balance for our purposes. On the one hand, they are fairly restrictive: not only they are stricter than log-space or poly-time reductions that are often used to define **NP**-completeness, but they also have less power than our other classes of our interest, $\mathbf{NC}^1$ and **P**, hence they give rise to a sensible notion of $\mathbf{NC}^1$-completeness. On the other hand, $\mathbf{TC}^0$ is powerful enough to support the kind of syntactic manipulations that we will use to define our reductions, such as substituting one formula into another. We believe that the bulk of our completeness results actually hold under more restricted notions of reductions, such as dlogtime reductions, but the extra effort needed to get there would distract us from the main point of this paper.

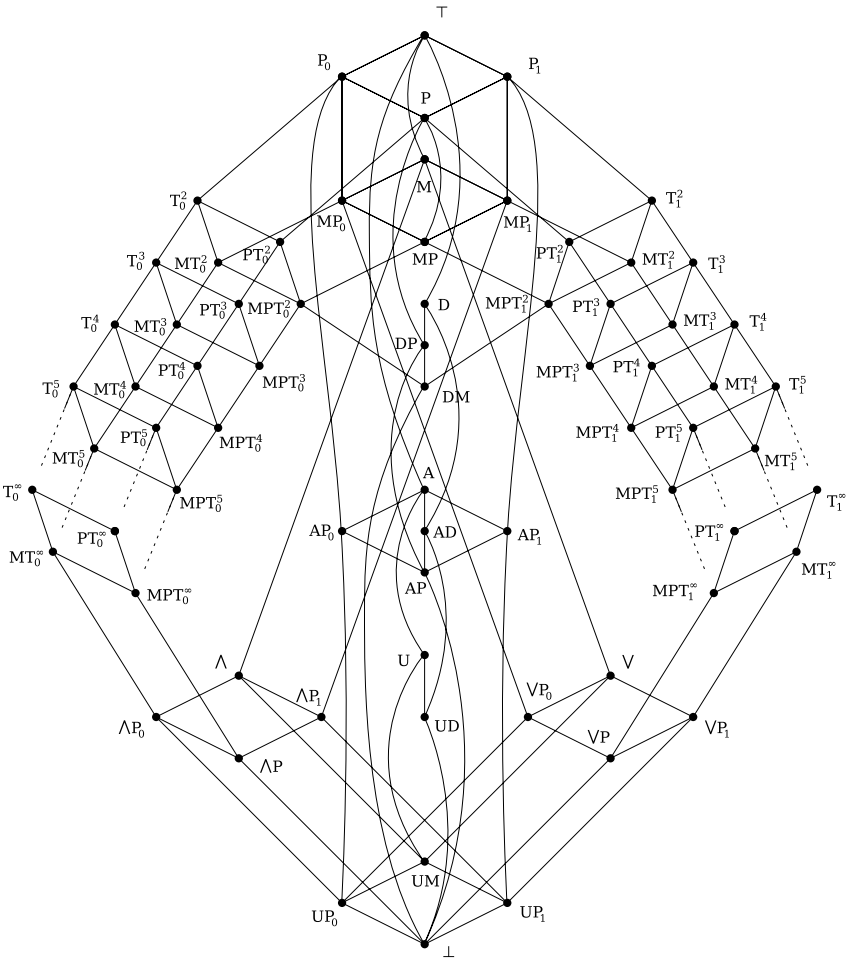

Figure 1: The lattice of Boolean clones (Post's lattice)

3 The complexity of CMP

We start with upper bounds. Similarly to [2, 3, 17], we will extract an algorithm for CMP from the characterization of clones in terms of the clone–coclone Galois connection, which implies

$$f \in [F] \iff \mathrm{Inv}(F) \subseteq \mathrm{Inv}(f) \iff \forall r \in \mathrm{Rel}\,(F \not\rhd r \text{ or } f \rhd r).$$

Using Fact 2.1, we may restrict attention to $r \in R_\infty$, but this is still an infinite number of invariants, hence we need an efficient bound on how far up in the R_n hierarchy we need to go.

Lemma 3.1 *Let $n \geq 1$, $f \in \mathrm{Op}_n$, and $\alpha \in \mathbf{2}$. The following are equivalent:*

(*i*) $f \in \mathrm{T}^\infty_\alpha$.

(*ii*) $f \in \mathrm{T}^n_\alpha$.

(*iii*) *There is $i < n$ such that $x_i \leq^\alpha f(\vec{x})$ for all $\vec{x} \in \mathbf{2}^n$, where $\leq^1 = \leq$, $\leq^0 = \geq$.*

Proof: (i) $\to$ (ii) is trivial, and (iii) $\to$ (i) follows easily from the definition.

(ii) $\to$ (iii): If (iii) does not hold, let us fix for each $i < n$ a vector $a^i = \langle a^i_0, \dots, a^i_{n-1}\rangle \in \mathbf{2}^n$ such that $a^i_i = \alpha$ and $f(a^i) = \neg\alpha$. Then the matrix $(a^j_i)^{j<n}_{i<n}$ witnesses that $f \not\triangleright r^n_\alpha$: if, say, $\alpha = 1$, we have $a^0_i \vee \dots \vee a^{n-1}_i \geq a^i_i = 1$ for each $i < n$, but $f(a^0) \vee \dots \vee f(a^{n-1}) = 0$. □

Corollary 3.2 *If $F \subseteq \mathrm{Op}$ and $f \in \mathrm{Op}_n$, then*

$$f \in [F] \iff \forall r \in R_n\, (F \not\triangleright r \text{ or } f \triangleright r). \tag{1}$$

Proof: The left-to-right implication is clear. For the right-to-left implication, if $f \notin [F]$, then one of the completely meet-irreducible clones C as given in Fact 2.1 satisfies $F \subseteq C$ and $f \notin C$. Moreover, if $C = \mathrm{T}^m_\alpha$ for some $\alpha \in \mathbf{2}$ and $m > n$, then also $f \notin \mathrm{T}^n_\alpha \supseteq C \supseteq F$ by Lemma 3.1. Thus, we may assume $C = \mathrm{Pol}(r)$ for some $r \in R_n$, i.e., $F \triangleright r$ and $f \not\triangleright r$. □

Lemma 3.3 *If $f \in \mathrm{Op}$ and $r \in \mathrm{Rel}$ are given by Boolean circuits, we can test if $f \triangleright r$ in* **coNP**.

Proof: Straight from the definition, we have that $f \not\triangleright r$ iff there exists a matrix $A = (a^j_i)^{j<k}_{i<n}$ (which is a polynomial-size object) such that $\langle a^0_i, \dots, a^{k-1}_i\rangle \in r$ for each $i < n$, and

$$\langle f(a^0_0, \dots, a^0_{n-1}), \dots, f(a^{k-1}_0, \dots, a^{k-1}_{n-1})\rangle \notin r.$$

These properties of A can be checked in polynomial time. □

Theorem 3.4

(*i*) $\mathrm{CMP} \in \mathbf{\Theta}^{\mathbf{P}}_2$.

(*ii*) $\mathrm{CMP}_C \in$ **coNP** *for each clone $C \subseteq \mathrm{Op}$.*

(*iii*) $\mathrm{CMP}^f \in$ **NP** *for each $f \in \mathrm{Op}$.*

Proof:

(i): Given $F \subseteq \mathrm{Op}$ and $f \in \mathrm{Op}$ of arity n, we can determine if $f \in [F]$ in $\mathbf{P}^{\|\mathbf{NP}}$ by evaluating (1): there are $2n + O(1)$ relations in R_n, and they can be described by efficiently constructible Boolean circuits. Thus, in view of Lemma 3.3, we can ask the **NP** oracle if $F \triangleright r$ and if $f \triangleright r$ for each $r \in R_n$ in parallel, and read the answer off of the oracle responses.

(ii): We use Corollary 3.2 again, but since C is fixed, we can test $C \triangleright r$ in deterministic polynomial time: $\{r \in R_0 : C \triangleright r\}$ is a finite set, and for each $\alpha \in \mathbf{2}$, $\{m \in \mathbb{N}_{>0} : C \triangleright r^m_\alpha\}$ is a downward-closed subset of $\mathbb{N}_{>0}$, i.e., either all of $\mathbb{N}_{>0}$, or a finite set. Thus, (1) can be evaluated in **coNP**.

(iii) is even simpler: since f (hence n) is fixed, R_n is a fixed finite set, and so is $\{r \in R_n : f \not\triangleright r\}$. Thus, we can test if $F \not\triangleright r$ for each r from this finite set in **NP** by Lemma 3.3. □

We now turn to lower bounds which will show that Theorem 3.4 is mostly optimal, with a few exception in (ii) and (iii). We start with CMP_C and CMP^f, where we can prove (**co**)**NP**-hardness by simple reductions from Boolean satisfiability.

Theorem 3.5 (Böhler and Schnoor [3]) *Let C be a Boolean clone.*

(*i*) *If $C \supseteq \mathrm{P}$, then $\mathrm{CMP}_C \in \mathbf{P}$. More precisely, $\mathrm{CMP}_\top$ is trivial, and CMP_C is* **P***-complete for $C = \mathrm{P}_0, \mathrm{P}_1, \mathrm{P}$.*

(*ii*) *Otherwise (i.e., if $C \subseteq \mathrm{M}$, D, A, T_0^2, or T_1^2), CMP_C is* **coNP***-complete.*

Proof:

(i): $f \in \mathrm{P}_\alpha$ iff $f(\alpha, \ldots, \alpha) = \alpha$, which can be checked in polynomial time. In fact, it is easy to see that testing membership in P_0, P_1, or P is equivalent to the **P**-complete problem of evaluation of Boolean circuits.

(ii): That C is included in M, D, A, T_0^2, or T_1^2 follows by inspection of Post's lattice (Fig. 1). We have $\mathrm{CMP}_C \in \mathbf{coNP}$ by Theorem 3.4. In order to show **coNP**-hardness, we will provide a reduction from UnSat; it will even be independent of C.

Given a formula φ in variables $\vec{u}$, let

$$f_\varphi(x, y, z, \vec{u}) = \big((x \wedge \varphi) \wedge y\big) \vee \big(\neg(x \wedge \varphi) \wedge z\big).$$

Then

$$\varphi \in \textsc{UnSat} \implies f_\varphi \equiv z \implies f_\varphi \in \bot \subseteq C, \tag{2}$$

$$\varphi \in \textsc{Sat} \implies f_\varphi \notin \mathrm{M}, \mathrm{A}, \mathrm{D}, \mathrm{T}_0^2, \mathrm{T}_1^2 \implies f_\varphi \notin C. \tag{3}$$

Indeed, (2) is obvious. For (3), if $\vec{a}$ is a satisfying assignment to φ, we see that

$$(x \wedge y) \vee (\neg x \wedge z) = f_\varphi(x, y, z, \vec{a}) \in [f_\varphi, 0, 1],$$

thus $f_\varphi \notin \mathrm{M}, \mathrm{A}$. Moreover,

$$f_\varphi(1, 1, 0, \vec{a}) = f_\varphi(0, 0, 1, \neg\vec{a}) = 1,$$
$$f_\varphi(1, 0, 1, \vec{a}) = f_\varphi(0, 1, 0, \neg\vec{a}) = 0,$$

thus $f_\varphi \notin \mathrm{D}, \mathrm{T}_1^2, \mathrm{T}_0^2$. □

Theorem 3.6 *Let $f \in \mathrm{Op}$.*

(*i*) *If f is a projection (i.e., $f \in \bot$), then CMP^f is trivial.*

(*ii*) *Otherwise, CMP^f is* **NP***-complete.*

Proof: Let $\varphi \mapsto f_\varphi$ be the reduction from the proof of Theorem 3.5. It follows from the definition that $f_\varphi \in \mathrm{P}$, while (3) implies that $[f_\varphi] \supseteq \mathrm{P}$ if φ is satisfiable. Thus,

$$\varphi \in \textsc{Sat} \implies [f_\varphi] = \mathrm{P},$$
$$\varphi \in \textsc{UnSat} \implies [f_\varphi] = \bot,$$

hence $\varphi \mapsto \{f_\varphi\}$ is a reduction from Sat to CMP^f whenever $f \in \mathrm{P} \setminus \bot$. Likewise, the reduction $\varphi \mapsto \{f_\varphi, 0, 1\}$ works whenever $f \notin [0, 1]$, and $\varphi \mapsto \{f_\varphi, \neg\}$ works whenever $f \notin [\neg]$. This covers all cases where f is not a projection. □

Corollary 3.7 CMP $\notin$ **coNP** *unless* **NP** = **coNP**. □

It will take more work to establish the true complexity of CMP. Notice first that the only way it can get as hard as $\mathbf{\Theta}_2^{\mathbf{P}}$ is by interaction of F and f deep inside one of the infinite arms of Post's lattice: otherwise (1) holds with a constant n, in which case the criterion can be evaluated in $\mathbf{P}^{\mathbf{NP}[O(1)]}$, i.e., in the Boolean hierarchy (cf. Lemma 4.2).

A convenient supply of functions on the infinite arms of Post's lattice is given by threshold functions, hence our first task will be to describe exactly what clones they generate. For completeness, the lemma below is stated including various cases that we will not actually need.

Lemma 3.8 *Let* $n \geq 1$ *and* $0 \leq t \leq n+1$. *Then*

$$[\theta_t^n] = \begin{cases} \bot & t = n = 1, \\ \mathrm{UP}_1 & t = 0, \\ \bigvee\mathrm{P} & t = 1, \quad n > 1, \\ \mathrm{MPT}_1^{\lfloor (n-1)/(t-1) \rfloor} & 1 < t \leq \frac{n}{2}, \\ \mathrm{DM} & t = \frac{n+1}{2}, \quad n > 1, \\ \mathrm{MPT}_0^{\lceil t/(n-t) \rceil} & \frac{n}{2} + 1 \leq t < n, \\ \bigwedge\mathrm{P} & t = n, \quad n > 1, \\ \mathrm{UP}_0 & t = n+1. \end{cases}$$

Proof: The cases with $t \leq 1$, $t \geq n$, or $n = 1$ are straightforward.

Notice that the dual of θ_t^n is θ_{n+1-t}^n. Thus, if $t = (n+1)/2$ (which implies n is odd), then $\theta_t^n \in \mathrm{DM}$. By Fig. 1, DM is a minimal clone, hence $[\theta_t^n] = \mathrm{DM}$ unless θ_t^n is a projection, which only happens when $n = 1$.

Assume that $1 < t \leq n/2$. Clearly, $\theta_t^n \in \mathrm{MP}$. Since $\mathbb{N}_{<n} = \{0, \ldots, n-1\}$ has two disjoint subsets of size $\lfloor n/2 \rfloor \geq t$, $\theta_t^n \notin \mathrm{Pol}(r_0^2) = \mathrm{T}_0^2$. Also, $t \geq 2$ implies that θ_t^n is not bounded below by a variable, i.e., $\theta_t^n \notin \mathrm{T}_1^\infty$ by Lemma 3.1. By inspection of Post's lattice, it follows that $[\theta_t^n] = \mathrm{MPT}_1^k$ for some $k \geq 1$. Now, for any $k \geq 1$, we have $\theta_t^n \not\triangleright r_1^k$ iff $\mathbb{N}_{<n}$ can be covered by k subsets of size $< t$ iff $n \leq k(t-1)$, thus

$$\theta_t^n \in \mathrm{T}_1^k \iff n \geq 1 + k(t-1) \iff k \leq \left\lfloor \frac{n-1}{t-1} \right\rfloor,$$

and consequently $[\theta_t^n] = \mathrm{MPT}_1^{\lfloor (n-1)/(t-1) \rfloor}$.

Finally, assume that $\frac{n}{2} + 1 \leq t < n$. The dual of θ_t^n is $\theta_{t'}^n$, where $t' = n + 1 - t$ satisfies $1 < t' \leq \frac{n}{2}$. Thus, by the case that we just proved, $[\theta_{t'}^n] = \mathrm{MPT}_1^k$ with

$$k = \left\lfloor \frac{n-1}{t'-1} \right\rfloor = \left\lfloor \frac{n-1}{n-t} \right\rfloor = \left\lfloor \frac{t + (n-t-1)}{n-t} \right\rfloor = \left\lceil \frac{t}{n-t} \right\rceil,$$

and $[\theta_t^n]$ is its dual, MPT_0^k. □

In order to prove the $\mathbf{\Theta}_2^{\mathbf{P}}$-completeness of CMP, we will need a convenient $\mathbf{\Theta}_2^{\mathbf{P}}$-complete language to reduce from. In fact, the statement below effectively gives a $\mathbf{\Theta}_2^{\mathbf{P}}$-complete *promise problem* rather than a language. It is essentially due to Wagner [18, Cor. 6.4] and Buss and Hay [5, Thm. 8] (see also [12, L. 2.1]).

Lemma 3.9 *Let $L \subseteq \Sigma^*$ be any language such that $L \in \mathbf{\Theta}_2^{\mathbf{P}}$. Then there exists a $\mathbf{TC}^0$-function $w \mapsto \langle \varphi_{w,i} : i < 2n_w \rangle$ (where each $\varphi_{w,i}$ is a CNF) with the following property: for every $w \in \Sigma^*$, there exists $0 < j \leq 2n_w$ such that for all $i < 2n_w$,*

$$\varphi_{w,i} \in \textsc{Sat} \iff i < j,$$

and we have

$$w \in L \iff j \text{ is even.}$$

Proof: L is computable by a polynomial-time Turing machine $M(w)$ that makes $|w|^c$ parallel (non-adaptive) queries to an **NP** oracle. Given a $w \in \Sigma^*$, put $n_w = |w|^c + 1$; for any $i < n_w$, let φ_{2i} be a CNF whose satisfiability is equivalent to the **NP** property "at least i queries made by $M(w)$ have positive answers", φ'_{2i+1} a CNF expressing "there is an accepting run of $M(w)$ with i positive answers to queries, all of which are correct", and φ_{2i+1} a CNF equivalent to $\varphi'_{2i+1} \vee \varphi_{2i}$. If k is the true number of positively answered queries made by $M(w)$, then φ_i is satisfiable iff $i < 2k+1$ or $i < 2k+2$ depending on if $w \in L$. □

Lemma 3.10 *There exists a $\mathbf{TC}^0$-function $\langle \varphi_i : i < n \rangle \mapsto f_{\vec{\varphi}}$, and for each n, a sequence of integers $k_{n,0} > k_{n,1} > \cdots > k_{n,n} \geq 2$, with the following property: whenever $\langle \varphi_i : i < n \rangle$ is a sequence of formulas, we have $[\twoheadrightarrow, f_{\vec{\varphi}}] = \mathrm{T}_0^{k_{n,s}}$, where $s = \big|\{i < n : \varphi_i \in \textsc{Sat}\}\big|$.*

Proof: For a given n, fix $m > n$ (to be specified later) and $t = m - n - 1$. We may assume that the formulas φ_i use pairwise disjoint sets of variables that are also disjoint from $\{x_i : i < m\}$. Put

$$f_{\vec{\varphi}} = \theta_t^m(x_0 \wedge \varphi_0, \ldots, x_{n-1} \wedge \varphi_{n-1}, x_n, \ldots, x_{m-1}).$$

When φ_i is unsatisfiable, we have $x_i \wedge \varphi_i \equiv 0$. Thus, renumbering w.l.o.g. the φ_is so that each φ_i, $i < s$, is satisfiable,

$$f_{\vec{\varphi}} \equiv \theta_t^{m-n+s}(x_0 \wedge \varphi_0, \ldots, x_{s-1} \wedge \varphi_{s-1}, x_n, \ldots, x_{m-1}).$$

On the one hand, $x_i \wedge \varphi_i \in \mathrm{T}_0^\infty = [\twoheadrightarrow]$ by Lemma 3.1, thus $f_{\vec{\varphi}} \in [\twoheadrightarrow, \theta_t^{m-n+s}]$. On the other hand, for each $i < s$, we may choose a satisfying assignment a_i to φ_i, substitute $0 \in [\twoheadrightarrow]$ for each variable made 0 by a_i, and substitute x_i for each variable made 1 by a_i. (By our assumptions on variables, we can do this independently for each $i < s$, and it will not affect the $\vec{x}$ variables.) Under this substitution, $x_i \wedge \varphi_i$ becomes equivalent to x_i, and $f_{\vec{\varphi}}$ becomes $\theta_t^{m-n+s}(x_0, \ldots, x_{s-1}, x_n, \ldots, x_{m-1})$. Thus,

$$[\twoheadrightarrow, f_{\vec{\varphi}}] = [\twoheadrightarrow, \theta_t^{m-n+s}] = \mathrm{T}_0^{k_{n,s}}$$

using Lemma 3.8, where

$$k_{n,s} = \left\lceil \frac{t}{m-n-t+s} \right\rceil = \left\lceil \frac{t}{s+1} \right\rceil,$$

as long as $m/2+1 \le t$, i.e., $m \ge 2n+4$. In order to satisfy the constraint $k_{n,0} > k_{n,1} > \cdots > k_{n,n}$, it suffices to ensure that

$$\frac{t}{s+1} + 1 \le \frac{t}{s}$$

for all $s \le n$, i.e., $t \ge n(n+1)$. This holds if we choose $m = \max\{(n+1)^2, 6\}$. □

Theorem 3.11 CMP *is* $\mathbf{\Theta}_2^{\mathbf{P}}$*-complete.*

Proof: We have CMP $\in \mathbf{\Theta}_2^{\mathbf{P}}$ by Theorem 3.4. In order to show that it is $\mathbf{\Theta}_2^{\mathbf{P}}$-hard, fix $L \in \mathbf{\Theta}_2^{\mathbf{P}}$.

Given w, compute $\langle \varphi_i : i < 2n_w \rangle$ as in Lemma 3.9, and then (abbreviating $n = n_w$) $f_{\text{even}} = f_{\varphi_0,\varphi_2,\ldots,\varphi_{2n-2}}$, $f_{\text{odd}} = f_{\varphi_1,\varphi_3,\ldots\varphi_{2n-1}}$ as in Lemma 3.10. If $j \le 2n$ is as in Lemma 3.9, then

$$\big|\{i < n : \varphi_{2i} \in \textsc{Sat}\}\big| = \lceil j/2 \rceil,$$
$$\big|\{i < n : \varphi_{2i+1} \in \textsc{Sat}\}\big| = \lfloor j/2 \rfloor,$$

thus

$$[\nrightarrow, f_{\text{even}}] = \mathrm{T}_0^{k_{n,\lceil j/2 \rceil}},$$
$$[\nrightarrow, f_{\text{odd}}] = \mathrm{T}_0^{k_{n,\lfloor j/2 \rfloor}},$$

where $k_{n,0} > \cdots > k_{n,n} \ge 2$ are as in Lemma 3.10. It follows that

$$f_{\text{even}} \in [\nrightarrow, f_{\text{odd}}] \iff \lfloor j/2 \rfloor = \lceil j/2 \rceil \iff j \text{ is even} \iff w \in L.$$

Thus, $w \mapsto \langle \{\nrightarrow, f_{\text{odd}}\}, f_{\text{even}} \rangle$ is a reduction from L to CMP. □

Remark 3.12 We followed the tradition in the study of Boolean clones—going back to Post—of considering only functions of positive arity, even though the general theory of clones and coclones works more smoothly if nullary functions are also allowed. Let us see now what changes if we include nullary functions into consideration.

First, the number of Boolean clones increases—namely, each non-nullary clone C that includes at least one constant function (i.e., $C \supseteq \mathrm{UP}_0$ or UP_1) splits into two: one consisting only of non-nullary functions as before, and one that also includes nullary functions corresponding to all constant functions of C. In Fact 2.1, we understand the given definitions of meet-irreducible clones so that they include all applicable nullary functions; moreover, there is a new meet-irreducible clone $\mathrm{N} = [\wedge, \neg] = \mathrm{Pol}(r_N)$ of all non-nullary functions, where $r_N = \varnothing \in \mathrm{Rel}_1$. Consequently, we include r_N in R_n for each n. Note that $\mathrm{D} \subseteq \mathrm{N}$ and $\mathrm{P} \subseteq \mathrm{N}$.

Since the set $\{\wedge, \vee, \neg\}$ is no longer functionally complete, we read the definition of CMP and derived problems so that the input is given in the form of circuits over the basis $\{\wedge, \vee, \neg, 0, 1\}$, where 0 and 1 denote nullary constants.

The upper bounds in Theorem 3.4 continue to hold unchanged.

In Theorem 3.5, the main dichotomy still holds: $\mathrm{CMP}_C \in \mathbf{P}$ if $C \supseteq \mathrm{P}$, and CMP_C is **coNP**-complete otherwise. The difference is that now there are more clones $C \supseteq \mathrm{P}$, namely $\top$, N, P_0, NP_0, P_1, NP_1, and P. $\mathrm{CMP}_\top$ is trivial, and CMP_C is **P**-complete for $C = \mathrm{P}_\alpha, \mathrm{NP}_\alpha, \mathrm{P}$. The

problem $\mathrm{CMP}_{\mathrm{N}}$ amounts to testing if the given Boolean circuit has a nonzero number of input variables: the exact mechanics of this test will depend on syntactic details of the representation of input, but it can be done in $\mathbf{AC}^0$ under any reasonable respresentation.

The statement of Theorem 3.6 changes so that CMP^f is **NP**-complete if f is neither a projection nor a nullary function. If f is a nullary function, then CMP^f is **P**-complete: if $\alpha \in \mathbf{2}$ is a nullary constant, we have that $\alpha \in [F]$ iff either $\alpha \in F$, or F contains the dual constant $\neg\alpha$ and $F \not\subseteq \mathrm{P}_{\neg\alpha}$.

All named clones in Lemmas 3.8 and 3.10 need to be intersected with N, so that, e.g., the conclusion of Lemma 3.10 reads $[\nrightarrow, f_{\vec{\varphi}}] = \mathrm{NT}_0^{k_{n,s}}$.

The main Theorem 3.11 still holds.

Remark 3.13 We defined CMP and related problems so that the input functions are represented by Boolean circuits, which is the natural thing to do in a computational context. However, in the context of logic or algebra, it is more natural to represent Boolean functions by Boolean formulas, or equivalently, Boolean terms.

Fortunately, this has negligible effect on our results. First, all upper bounds hold also for the formula representation, because formulas are special cases of circuits. On the other hand, our main lower bounds continue to hold in this setting as well: we used reductions from Boolean satisfiability (that already works with formulas), and the most complicated tools we employed were threshold functions, which can be written with uniform polynomial-size formulas just as well as circuits.

The only exceptions are problems that we proved **P**-complete by reductions from evaluation of Boolean circuits: namely, CMP_C for $\mathrm{P} \subseteq C \neq \top, \mathrm{N}$ (Theorem 3.5), and CMP^f for f a nullary function (Theorem 3.6 as modified in Remark 3.12). If we change the input representation to formulas, then all these problems become $\mathbf{NC}^1$-complete, which is the complexity of evaluation of Boolean formulas (Buss [4]).

4 Restricted input bases

The problems CMP, CMP_C, and CMP^f are defined so that the input functions are given by circuits over a fixed functionally complete basis. This is reasonable if we consider these circuits to be just a computing device. However, if we view the problem as "given a circuit over the De Morgan basis, can we rewrite it as a circuit over a given basis F?", it makes perfect sense to also consider the case where the input circuits are expressed in a different basis. That is, for any finite $B \subseteq \mathrm{Op}$, let B-CMP be the following problem:

> **Input:** A finite set of functions $F \subseteq \mathrm{Op}$ and a function $f \in \mathrm{Op}$, all given by circuits over the basis B.
>
> **Output:** YES if $f \in [F]$, otherwise NO.

(We will refer to *B-circuits* for short instead of circuits over basis B, and likewise for *B-formulas*.) Similarly, we define the problems B-CMP_C and B-CMP^f.

The complexity of B-CMP$_C$ was thoroughly investigated by Böhler and Schnoor [3], who denote the problem as $\mathcal{M}_C(C \twoheadleftarrow B)$, and its formula version as $\mathcal{M}(C \twoheadleftarrow B)$. For most combinations of C and B, they were able to show that $\mathcal{M}(C \twoheadleftarrow B)$ and $\mathcal{M}_C(C \twoheadleftarrow B)$ are both **coNP**-complete, or both in **P**.

We will not make any effort to classify the complexity of the problems B-CMP^f, as the number of cases is prohibitively large, hence we consider it out of scope of this paper. (The problem has two clone parameters, analogously to B-CMP$_C$, which took Böhler and Schnoor a whole paper to understand, and even then their classification is incomplete.)

We will not obtain a complete classification of the complexity of B-CMP either, nevertheless we present a number of partial results, summarized in Corollary 4.15 at the end of the section.

Unless stated otherwise, the complexity results below also hold for variants of B-CMP where the input functions are represented by formulas. To be precise, we consider formulas as strings: if B consists of at most binary functions, we may write them in the common infix notation, but in general, it is perhaps best to settle on the prefix (Polish) notation.

For simplicity, we disallow nullary functions in this section, but the results below can be easily adapted to a setup where they are included.

Our results will be stated for arbitrary bases, but we will often need to express specific functions. Thus, it is useful to observe that we can more-or-less freely convert circuits and formulas to different bases. Notice that CNFs and other constant-depth formulas using unbounded fan-in $\wedge$, $\vee$ gates can be written as $O(\log n)$-depth bounded fan-in formulas.

Lemma 4.1 *Let $B, B' \subseteq \mathrm{Op}$ be finite such that $B \subseteq [B']$.*

(*i*) *Given a B-circuit, we can construct as equivalent B'-circuit by a* $\mathbf{TC}^0$ *function.*

(*ii*) *Given a B-formula of depth $O(\log n)$, we can construct as equivalent B'-formula by a* $\mathbf{TC}^0$ *function.*

Proof:

(i): For each $f \in B$, we fix an expression of f in terms of B', and replace with it all f-gates in the circuit.

(ii): Starting from a B-formula of depth $O(\log n)$, the construction above produces a B'-circuit of depth $O(\log n)$, which can be unwinded into a B'-formula of depth $O(\log n)$, and consequently of size $n^{O(1)}$. One can check that the whole procedure can be implemented in $\mathbf{TC}^0$. □

For the rest of this section, B is a finite subset of Op.

We start with the following simple observation. Recall that the *Boolean hierarchy* **BH** is defined as the closure of **NP** under (finitary) intersections, unions, and complements. Alternatively, it can be characterized as $\mathbf{BH} = \mathbf{P}^{\mathbf{NP}[O(1)]}$. The Boolean hierarchy is stratified into levels; the bottom level consists of **NP** and **coNP**, and the next level of the classes

$$\mathbf{DP} = \{L_0 \cap L_1 : L_0 \in \mathbf{NP}, L_1 \in \mathbf{coNP}\},$$
$$\mathbf{coDP} = \{L_0 \cup L_1 : L_0 \in \mathbf{NP}, L_1 \in \mathbf{coNP}\}.$$

Lemma 4.2 *If* $B \subseteq \mathrm{T}_0^\infty$, T_1^∞, $\bigwedge$, $\bigvee$, A, *or* D, *then* B-CMP $\in \mathbf{BH}$.

Proof: By inspection of Post's lattice, we see that there are only finitely many clones below $[B]$. Thus, if F and f are given by B-circuits, we have

$$f \in [F] \iff \forall r \in R_k \, (F \not\triangleright r \text{ or } f \triangleright r)$$

for some constant k, which gives a **BH** algorithm in view of Lemma 3.3. □

In fact, we can characterize the complexity of B-CMP for B as in Lemma 4.2 more precisely.

First, the tractable cases. The theorem below mostly follows from results of Böhler and Schnoor [3]: the given clones $[B]$ have only finitely many subclones, hence B-CMP $\in \mathbf{P}$ iff B-CMP$_C \in \mathbf{P}$ for each clone $C \subseteq [B]$. However, we give a self-contained proof, which also confirms that the complexity drops down to $\mathbf{NC}^1$ for formulas.

Theorem 4.3 *If* $B \subseteq \mathrm{MT}_0^\infty$, MT_1^∞, $\bigwedge$, $\bigvee$, A, *or* DM, *then* B-CMP $\in \mathbf{P}$. *If we represent the input by formulas rather than circuits,* B-CMP $\in \mathbf{NC}^1$.

Proof: Assume that $B \subseteq \bigvee$. The set of n-ary functions in $[B] = \bigvee$ is very limited: it consists only of 1 and the functions $f_I(x_0, \ldots, x_{n-1}) = \bigvee_{i \in I} x_i$ for $I \subseteq \{0, \ldots, n-1\}$ (including $f_\varnothing = 0$). Given a B-circuit, we can determine which of these functions it computes by evaluating it on the assignment $\vec{0}$, which detects the 1 function, and for each $i < n$, on $e_i = \langle 0, \ldots, 0, 1, 0, \ldots, 0 \rangle$ (with 1 at position i), which detects if $i \in I$.

Once we know the function, it is trivial to determine which of the finitely many subclones of $\bigvee$ it generates, and if we know the clones generated by f and by each element of F, we can find out if $f \in [F]$. This gives a polynomial-time algorithm. Moreover, since the algorithm just evaluates the input functions on polynomially many assignments in parallel, and then does $\mathbf{AC}^0$ post-processing, it can be implemented in $\mathbf{NC}^1$ if the functions are given by formulas rather than circuits.

The cases of $B \subseteq \bigwedge$ or $B \subseteq$ A are completely analogous to $\bigvee$.

Assume that $[B] =$ DM. Since DM is a minimal clone, a B-circuit either computes a projection, or it generates DM. Moreover, a self-dual monotone function $f \in \mathrm{Op}_n$ is the projection π_i^n iff $f(e_i) = 1$. Thus, we can again determine $[f]$ by evaluating f at n assignments.

Assume that $B \subseteq \mathrm{MT}_1^\infty$, the case of MT_0^∞ being dual. Given a B-circuit computing a function f, either $f \in \bigvee$, or $[f] = \mathrm{MPT}_1^\infty$, MT_1^∞. By evaluating $f(e_i)$ for all $i < n$, we find the only candidate $I \subseteq \{0, \ldots, n-1\}$ such that f could equal f_I. Then we evaluate f at the assignment a such that $a_i = 1$ iff $i \notin I$: if $f(a) = 0$, then $f \equiv f_I$, otherwise $f \notin \bigvee$. In the latter case, we can distinguish MT_1^∞ from MPT_1^∞ by evaluating $f(\vec{0})$. □

We remark that while B-CMP is **P**-complete (or $\mathbf{NC}^1$-complete in the formula representation) for $[B] = \mathrm{M(P)T}_\alpha^\infty$ or $[B] =$ DM, it is still easier for other classes from Theorem 4.3: for example, it is easy to show that if $[B] \subseteq \bigvee$, we can evaluate B-circuits in **L**, and B-formulas in $\mathbf{TC}^0$. We will not go into details.

We now turn to the remaining clones below T_α^∞ and D. We will use the following result for lower bounds. Here, the *equivalence problem* B-EQ is to compute if two given Boolean formulas over basis B are equivalent.

Theorem 4.4 (Reith [15]) *If* $[B] \supseteq \mathrm{MPT}_0^\infty$, MPT_1^∞, *or* DM, *then* B-EQ *is* **coNP**-*complete.* □

We mention that since the formulas used in Theorem 4.4 ultimately come from a reduction from SAT, they can be taken to have depth $O(\log n)$. In particular, this ensures they can be efficiently converted to a different basis by Lemma 4.1.

The constructions in the next lemma mostly come from Böhler and Schnoor [3], who used them to prove **coNP**-completeness of various instances of B-CMP_C. We observe that the lower bound can be improved to **coDP** if C is allowed to vary.

Lemma 4.5 *If* $[B] \supseteq \mathrm{PT}_0^\infty$, PT_1^∞, MPT_0^2, MPT_1^2, *or* DP, *then* B-CMP *is* **coDP**-*hard.*

Proof: Assume first $[B] \supseteq \mathrm{DP}$. If f, g are DM-formulas (i.e., formulas over a fixed basis of DM), let $h(\vec{x}, y) = f(\vec{x}) + g(\vec{x}) + y$; this can be expressed by a DP-formula as the ternary function $x + y + z$ is in DP. If $f \equiv g$, then $h(\vec{x}, y) \equiv y$. If $f \not\equiv g$, let $\vec{a}$ be an assignment such that $f(\vec{a}) \neq g(\vec{a})$. Then $h(\vec{0}, 1) = 1$ (using $f, g \in \mathrm{P}$) and $h(\vec{a}, 1) = 0$, hence h is not monotone. In view of $h \in \mathrm{DP}$, this implies $[h] = \mathrm{DP}$. Thus,

$$f \equiv g \implies [f + g + y] = \bot, \tag{4}$$

$$f \not\equiv g \implies [f + g + y] = \mathrm{DP}. \tag{5}$$

Now, since DM-EQ is **coNP** by Theorem 4.4, the language

$$L = \big\{\langle f, g, f', g'\rangle : \langle f, g\rangle \in \text{DM-EQ or } \langle f', g'\rangle \notin \text{DM-EQ}\big\}$$

is **coDP**-complete. Using (4) and (5),

$$\langle f, g, f', g'\rangle \in L \iff [f + g + y] \subseteq [f' + g' + y],$$

which gives a reduction of L to B-CMP.

If $[B] \supseteq \mathrm{MPT}_1^2$ (the case of MPT_0^2 is dual), we can use in a similar way the MPT_1^2-formula

$$h(\vec{x}, y, z) = \theta_2^3\big(f(\vec{x}) \vee g(\vec{x}), y, z\big).$$

The dual of h is $\theta_2^3\big(f(\vec{x}) \wedge g(\vec{x}), y, z\big)$, hence h is self-dual if and only if $f \equiv g$. Moreover, $f(x, \ldots, x) \equiv g(x, \ldots, x) \equiv x$, hence in any case $\theta_2^3 \in [h]$. Thus,

$$f \equiv g \implies [h] = \mathrm{DM},$$
$$f \not\equiv g \implies [h] = \mathrm{MPT}_1^2.$$

This gives a reduction of L to B-CMP in the same way as above.

Finally, assume $[B] \supseteq \mathrm{PT}_1^\infty$ (the case of PT_0^∞ is dual). Given MPT_1^∞-formulas f and g, we put $h(\vec{x}, y) = y \vee \big(f(\vec{x}) + g(\vec{x})\big)$, which can be expressed by a PT_1^∞-formula. If $f \equiv g$, then $h \equiv y$; otherwise, we check easily that h is not monotone. Thus,

$$f \equiv g \implies [h] = \bot,$$
$$f \not\equiv g \implies [h] = \mathrm{PT}_1^\infty,$$

which yields a reduction of the **coDP**-complete problem

$$\left\{\langle f,g,f',g'\rangle : \langle f,g\rangle \in \mathrm{MPT}_1^\infty\text{-EQ or } \langle f',g'\rangle \notin \mathrm{MPT}_1^\infty\text{-EQ}\right\}$$

to B-CMP. □

Incidentally, the argument we gave for MPT_α^2 also resolves one of the problems left open by Böhler and Schnoor [3]:

Corollary 4.6 *If* $[B] \supseteq \mathrm{MPT}_0^2$ *or* MPT_1^2*, then* B-$\mathrm{CMP}_{\mathrm{DM}}$ *is* **coNP***-complete.* □

Theorem 4.7 *If* $B \subseteq \mathrm{T}_0^\infty$, T_1^∞, *or* D, *but* $B \not\subseteq \mathrm{M}$, *then* B-CMP *is* **coDP***-complete.*

Proof: **coDP**-hardness was proved in Lemma 4.5. Assume that $B \subseteq \mathrm{T}_1^\infty$, we will refine Lemma 4.2 to show that B-CMP $\in$ **coDP**. (The case of $B \subseteq \mathrm{T}_0^\infty$ is dual. The argument for $B \subseteq \mathrm{D}$ is similar, but easier.)

First, given a set of B-circuits F, we can determine $[F]$ in polynomial time using a single **coNP** oracle query, namely $F \stackrel{?}{\subseteq} \mathrm{M}$: indeed, if $F \not\subseteq \mathrm{M}$, then $[F]$ is PT_1^∞ or T_1^∞, and we can distinguish these two cases by testing if $F \subseteq \mathrm{P}$ (or equivalently, P_0), which we can do by evaluating $g(\vec{0})$ for each $g \in F$. On the other hand, if $F \subseteq \mathrm{M}$, i.e., $F \subseteq \mathrm{MT}_1^\infty$, we can compute $[F]$ using the algorithm in Theorem 4.3 (which again proceeds by evaluation of F at various assignments, hence it does not matter that the circuits are given in a larger basis).

This already shows that B-CMP $\in \mathbf{P}^{\|\mathbf{NP}[2]}$: we can test if $f \in [F]$ using two parallel **coNP** queries, $F \stackrel{?}{\subseteq} \mathrm{M}$ and $f \stackrel{?}{\in} \mathrm{M}$.

In order to improve this to **coDP**, we modify the algorithm so that it speculatively explores all computation branches with all possible oracle answers, and only makes the oracle queries needed at the end.

In this way, the algorithm computes two candidate clones $C_0 \subseteq \mathrm{MT}_1^\infty$ and $C_1 \supseteq \mathrm{PT}_1^\infty$ for $[F]$. We have $C_0 \subseteq C_1$: the choice of C_1 as PT_1^∞ or T_1^∞ is made according to if $F \subseteq \mathrm{P}$, and this information is taken into account also when computing C_0. In fact, this means that C_1 is the join $C_0 \vee \mathrm{PT}_1^\infty$ in the lattice of clones.

(Actually, the algorithm may fail to compute C_0 because it runs into an inconsistency that already shows $F \not\subseteq \mathrm{M}$. In this case, we may pick C_0 in an arbitrary way such that the properties $C_0 \subseteq \mathrm{M}$ and $C_1 = C_0 \vee \mathrm{PT}_1^\infty$ hold, say, $C_0 = \bot$ or $C_0 = \mathrm{UP}_1$: if the choice of C_0 became relevant in subsequent computation, it would be dismissed by the oracle as $F \not\subseteq \mathrm{M}$.)

Likewise, we obtain two candidate clones C_0', C_1' for $[f]$, with $C_0' \subseteq \mathrm{MT}_1^\infty$ and $C_1' = C_0' \vee \mathrm{PT}_1^\infty$.

Notice that $C_1' \not\subseteq C_0$ as $C_0 \subseteq \mathrm{M}$, and that $C_0' \subseteq C_1$ implies $C_1' = C_0' \vee \mathrm{PT}_1^\infty \subseteq C_1$. Thus, there are only the following possibilities:

- $C_0' \subseteq C_0$ (whence $C_1' \subseteq C_1$): then $f \in [F]$ if and only if $F \not\subseteq \mathrm{M}$ or $f \in \mathrm{M}$.
- $C_1' \subseteq C_1$ and $C_0' \not\subseteq C_0$: then $f \in [F]$ if and only if $F \not\subseteq \mathrm{M}$.
- $C_0' \not\subseteq C_1$: then $f \notin [F]$.

Consequently, the whole algorithm can be implemented in **coDP**: we have

$$f \in [F] \iff \langle F, f\rangle \in L_0 \text{ or } \langle F, f\rangle \in L_1$$

with

$$L_0 = \big\{\langle F, f\rangle : C_1' \subseteq C_1 \text{ and } F \not\subseteq \mathrm{M}\big\} \in \mathbf{NP},$$
$$L_1 = \big\{\langle F, f\rangle : C_0' \subseteq C_0 \text{ and } f \in \mathrm{M}\big\} \in \mathbf{coNP},$$

where C_0, C_1, C_0', C_1' are computed from $\langle F, f\rangle$ in deterministic polynomial time as described above. □

The question we are mainly interested in is for which bases B is B-CMP $\mathbf{\Theta}_2^{\mathbf{P}}$-complete. First, we can easily adapt Theorem 3.11 to P-CMP using the following translation (for completeness, we formulate it more generally than what we need):

Lemma 4.8 *Let $B, B' \subseteq \mathrm{Op}$ be finite. Assume that $\wedge \in [B']$ and $B \subseteq [B', 0]$, or $\vee \in B'$ and $B \subseteq [B', 1]$, or $\wedge, \vee \in [B']$ and $B \subseteq [B', 0, 1]$.*

(i) Given a B-circuit that computes a B'-function f, we can compute in $\mathbf{TC}^0$ a B'-circuit that computes f.

(ii) Given a B-formula of depth $O(\log n)$ that computes a B'-function f, we can compute in $\mathbf{TC}^0$ a B'-formula that computes f.

In particular, this applies to arbitrary B if $[B'] \supseteq \mathrm{P}$.

Proof: Assume first $\wedge \in [B']$ and $B \subseteq [B', 0]$. Without loss of generality, $0 \notin [B']$, hence $B' \subseteq \mathrm{P}_1$.

Since $B \subseteq [B', 0]$, Lemma 4.1 implies that we can find a B'-circuit g (or even an $O(\log n)$-depth B'-formula) such that $f(\vec{x}) \equiv g(\vec{x}, 0)$. Then $f(\vec{x}) \equiv g\big(\vec{x}, \bigwedge_i x_i\big)$: the two expressions agree when $\vec{x} \neq \vec{1}$ as $\bigwedge_i x_i = 0$; they also agree for $\vec{x} = \vec{1}$ as $f(\vec{1}) = 1 = g(\vec{1}, 1)$ on account of $f, g \in \mathrm{P}_1$. Since $\bigwedge_i x_i$ has an $O(\log n)$-depth formula over the basis $\{\wedge\} \subseteq [B']$, it can be written by an $O(\log n)$-depth B'-formula using Lemma 4.1 again.

The case $\vee \in [B']$ and $B \subseteq [B', 1]$ is dual.

Let $\wedge, \vee \in [B']$ and $B \subseteq [B', 0, 1]$. If $0 \in [B']$ or $1 \in [B']$, we are done by one of the previous cases, hence we may assume $0, 1 \notin [B']$, which implies $B' \subseteq \mathrm{P}$. Then we proceed as before: we find a B'-circuit g such that $f(\vec{x}) \equiv g(\vec{x}, 0, 1)$, and we observe that $f(\vec{x}) \equiv g\big(\vec{x}, \bigwedge_i x_i, \bigvee_i x_i\big)$ because of $f, g \in \mathrm{P}$. □

Theorem 4.9 *If $[B] \supseteq \mathrm{P}$, then B-CMP is $\mathbf{\Theta}_2^{\mathbf{P}}$-complete.*

Proof: The formulas we constructed in the proof of Theorem 3.11 (or rather, Lemma 3.10) compute functions in P_0, hence we can convert them to P_0-formulas by Lemma 4.8. Thus, in the setting of Lemma 3.10, we map a sequence of formulas $\langle \varphi_i : i < n\rangle$ to a P_0-formula $f_{\vec{\varphi}}(\vec{x})$ such that $[\nrightarrow, f_{\vec{\varphi}}] = \mathrm{T}_0^k$, where $k \geq 2$ is as specified in the lemma.

Using the idea of Lemma 4.8, we can construct a B-formula $g_{\vec{\varphi}}(\vec{x}, y)$ such that $f_{\vec{\varphi}}(\vec{x}) \equiv g_{\vec{\varphi}}(\vec{x}, 0)$. By replacing $g_{\vec{\varphi}}$ with $g_{\vec{\varphi}}\big(\vec{x}, y \wedge \bigwedge_i x_i\big)$ if necessary, we ensure that

$$g_{\vec{\varphi}}(\vec{x}, y) \equiv f_{\vec{\varphi}}(\vec{x}) \vee \Big(y \wedge \bigwedge_i x_i\Big). \tag{6}$$

We claim that

$$[x \wedge (y \to z), g_{\vec{\varphi}}] = \mathrm{PT}_0^k, \tag{7}$$

hence we can use $[x \wedge (y \to z), g_{\vec{\varphi}}]$ in place of $[\nrightarrow, f_{\vec{\varphi}}]$ in the proof of Theorem 3.11 to get a reduction from any $\mathbf{\Theta}_2^{\mathbf{P}}$ language to B-CMP.

In order to prove (7), recall that $[x \wedge (y \to z)] = \mathrm{PT}_0^\infty$. Thus, $\mathrm{PT}_0^\infty \subseteq [x \wedge (y \to z), g_{\vec{\varphi}}] \subseteq \mathrm{P}$ and $[x \wedge (y \to z), g_{\vec{\varphi}}, 0] \supseteq [\nrightarrow, f_{\vec{\varphi}}] = \mathrm{T}_0^k$, which implies

$$[x \wedge (y \to z), g_{\vec{\varphi}}] = \mathrm{PT}_0^l$$

for some $1 \leq l \leq k$. It now suffices to show that $g_{\vec{\varphi}} \in \mathrm{T}_0^k = \mathrm{Pol}(r_0^k)$.

Assume for contradiction that there are assignments $\vec{a}^0, \ldots, \vec{a}^{k-1}$ such that $\vec{a}^0 \wedge \cdots \wedge \vec{a}^{k-1} = \vec{0}$, but $g_{\vec{\varphi}}(\vec{a}^0) = \cdots = g_{\vec{\varphi}}(\vec{a}^{k-1}) = 1$. Notice that if $\vec{a}^i = \vec{1}$ for some i, we may leave it out (or rather, replace it with another assignment from $g_{\vec{\varphi}}^{-1}[1]$), as still $\bigwedge_{j \neq i} \vec{a}^j = \vec{0}$. Thus, we may assume that none of the $\vec{a}^i$ is $\vec{1}$. But then $f_{\vec{\varphi}}(\vec{a}^i) = g_{\vec{\varphi}}(\vec{a}^i) = 1$ for each $i < k$ by (6) (more precisely, this holds for the truncation of $\vec{a}^i$ leaving out the last coordinate). This contradicts $f_{\vec{\varphi}} \in \mathrm{T}_0^k$. □

We would like to extend Theorem 4.9 to the clones PT_α^k, and for that we need efficient constructions of threshold functions in a PT_α^k basis. This in fact appears to be quite a challenging task. The best deterministic result we found is a construction by Cohen et al. [6], whose special case for the $\{\theta_2^{k+1}\}$ basis is as follows:

Theorem 4.10 (Cohen et al. [6]) *Let $k \geq 2$. There exists a constant c and a polynomial-time algorithm that, given a sufficiently large N, constructs in time $N^{O(1)}$ an $O(\log N)$-depth $\{\theta_2^{k+1}\}$-formula $\psi_N(x_0, \ldots, x_{N-1})$ such that*

$$\theta^N_{\lfloor N(k^{-1}+\varepsilon)\rfloor} \leq \psi_N \leq \theta^N_{\lceil N(k^{-1}-\varepsilon)\rceil},$$

where $\varepsilon = c/\sqrt{\log N}$. □

Since ψ_N can only reliably distinguish inputs whose normalized Hamming weights differ by $\Omega(1/\sqrt{\log N})$, it cannot be used to tell apart more than $O(\sqrt{\log N})$ different cases. Thus, in order to distinguish n possible outcomes as in Lemma 3.10, we would need $N = \exp(\Omega(n^2))$, which makes it useless for our purposes. For the special case $k = 2$, they give a better construction that reduces the error of approximation to $2^{-O(\sqrt{\log N})}$, which means we might get away with $N = n^{O(\log n)}$, but this is still insufficient.

In absence of a better idea, we resort to probabilistic constructions following the method of Valiant [16], who used it to prove the existence of short $\{\wedge, \vee\}$-formulas for majority; Gupta and Mahajan [7] modified his construction to produce $\{\theta_2^3\}$-formulas. We will use a similar idea to express suitable threshold functions by short formulas over the $\{\theta_k^{k+1}\}$ basis.

Theorem 4.11 *Let $k \geq 3$. There exist constants $\frac{1}{2} < \sigma_k < 1$ and $c \geq 1$, and a $\mathbf{TC}^0$ function T with the following properties. The input of T consists of numbers n, t, and e in unary, and $r \in \{0,1\}^*$; the output is a $\{\theta_k^{k+1}\}$-formula $T_{n,t,e,r}(x_0, \ldots, x_{n-1})$ of depth $O(\log n + \log e)$. If n is sufficiently large and $\sigma_k n < t \leq n$, then*

$$\Pr_{|r|=(n+e)^c}[T_{n,t,e,r} \equiv \theta_t^n] \geq 1 - 2^{-e}.$$

Proof: Given n and d, let $F_{n,d}$ be a random formula consisting of a complete $(k+1)$-ary tree of θ_k^{k+1}-gates of depth d, where each leaf is a propositional variable independently uniformly chosen from $\{x_i : i < n\}$. If $a \in \mathbf{2}^n$ is an assignment of weight $w = pn$, $p \in [0,1]$, then $F_{n,d}(a)$ is a Bernoulli random variable that takes value 1 with certain probability p_d. We can describe $F_{n,d}(a)$ as the value of a complete $(k+1)$-ary tree of depth d of θ_k^{k+1}-gates, where each leaf is an independently drawn random element of $\mathbf{2}$ with the probability of 1 being p. Since the $k+1$ input subformulas of any gate are independent, this gives a recurrence for p_d:

$$\begin{aligned} p_0 &= p, \\ p_{d+1} &= f(p_d), \end{aligned}$$

where

$$f(x) = x^{k+1} + (k+1)x^k(1-x) = (k+1)x^k - kx^{k+1}.$$

Clearly, f maps $[0,1]$ to $[0,1]$. In order to analyze the behaviour of p_d, we need to locate the fixed points of f in $[0,1]$, i.e., the roots of the polynomial $g(x) = f(x) - x$. The end-points 0 and 1 are roots. Moreover, the derivative

$$g'(x) = (k+1)kx^{k-1}(1-x) - 1$$

satisfies $g'(0) = g'(1) = -1 < 0$, hence there must be another root in $(0,1)$. On the other hand, Descartes's rule of signs implies that g has at most two positive roots. Thus, g has a *unique* root $\sigma_k = \sigma \in (0,1)$; g is negative on $(0,\sigma)$, and positive on $(\sigma,1)$. That is, σ is the unique fixed point of f in $(0,1)$, and we have

$$\begin{aligned} 0 < x < \sigma &\implies f(x) < x, \\ \sigma < x < 1 &\implies x < f(x). \end{aligned}$$

Consequently, for any p, the sequence p_d is monotone (decreasing for $0 < p < \sigma$, and increasing for $\sigma < p < 1$), and as such it has a limit, which must be a fixed point of f. Thus,

$$\begin{aligned} 0 \leq p < \sigma &\implies \lim_{d\to\infty} p_d = 0, \\ \sigma < p \leq 1 &\implies \lim_{d\to\infty} p_d = 1. \end{aligned}$$

We claim that σ is irrational, hence $p \neq \sigma$: since σ is a root of $1 - (k+1)x^{k-1} + kx^k$, σ^{-1} is a root of the monic polynomial $h(x) = x^k - (k+1)x + k$, and as such, it is an algebraic integer. Thus, if it were rational, it would be an actual integer; however, it is easy to see that $h(x) > 0$ for all $x \geq 2$, hence $1 < \sigma^{-1} < 2$.

Next, we need to analyze the rate of convergence of p_d. In the vicinity of σ, we have

$$f(x) = x + (x-\sigma)g'(\sigma) + O\big((x-\sigma)^2\big),$$

where $g'(\sigma) > 0$: since $g(0) = g(\sigma) = g(1) = 0$, g' has a root in $(0,\sigma)$ and another in $(\sigma,1)$, while it has at most two positive roots altogether by the rule of signs, hence $g'(\sigma) \neq 0$. We cannot have $g'(\sigma) < 0$ as g is negative on $(0,\sigma)$ and positive on $(\sigma,1)$.

Thus, we may fix constants $\varepsilon_0 > 0$ and $\gamma_0 > 1$ such that

$$|x-\sigma| \le \varepsilon_0 \implies |f(x)-\sigma| \ge \gamma_0 |x-\sigma|,$$

hence

$$|p_d - \sigma| \ge \min\big\{\varepsilon_0, \gamma_0^d |p-\sigma|\big\}.$$

By Roth's theorem, σ has irrationality measure 2, i.e., for any $\delta > 0$, all but finitely many pairs of integers $a, b > 0$ satisfy

$$\left|\frac{a}{b} - \sigma\right| \ge b^{-(2+\delta)}.$$

(The weaker theorem of Liouville, bounding the irrationality measure by $k-1$, would also work for our purposes.) Applying this to $p = w/n$, we obtain

$$\log |p-\sigma|^{-1} \le (2+o(1)) \log n,$$

thus there exists a constant c_0 such that

$$d \ge c_0 \log n \implies |p_d - \sigma| \ge \varepsilon_0 \tag{8}$$

for any sufficiently large n and $p = w/n$.

In the vicinity of the end-points, we have

$$f(x) = (k+1)x^k + O(x^{k+1}), \tag{9}$$

$$f(1-x) = 1 - \binom{k+1}{2} x^2 + O(x^3). \tag{10}$$

Thus, we may fix $\varepsilon_1 > 0$ and $\gamma_1 > 0$ such that

$$0 \le x \le \varepsilon_1 \implies f(x) \le \gamma_1 x^2 \text{ and } f(1-x) \ge 1 - \gamma_1 x^2,$$

hence

$$0 \le p_d \le \varepsilon_1 \implies p_{d+d'} \le \gamma_1^{-1} (\gamma_1 p)^{2^{d'}}, \tag{11}$$

$$1 - \varepsilon_1 \le p_d \le 1 \implies p_{d+d'} \ge 1 - \gamma_1^{-1} \big(\gamma_1 (1-p)\big)^{2^{d'}}. \tag{12}$$

We may assume $\gamma_1 \varepsilon_1 < 1$. There is a constant d_0 such that

$$|p_d - \sigma| \ge \varepsilon_0 \implies p_{d+d_0} \le \varepsilon_1 \text{ or } 1 - p_{d+d_0} \le \varepsilon_1. \tag{13}$$

Putting (8), (11), (12) and (13) together, there is a constant c_1 such that

$$d \geq c_1(\log n + \log e) \implies p_d \leq 2^{-n-e} \text{ or } 1 - p_d \leq 2^{-n-e}$$

for sufficiently large n. Using the union bound over all 2^n assignments, we obtain

$$d \geq c_1(\log n + \log e) \implies \Pr[F_{n,d} \equiv \theta^n_{\lceil \sigma n \rceil}] \geq 1 - 2^{-e}.$$

Now, given n and t such that $\sigma n < t \leq n$, put $N = \lfloor \sigma^{-1} n \rfloor$. Then $\lceil \sigma N \rceil = t$, thus

$$\theta^n_t(\vec{x}) \equiv \theta^N_{\lceil \sigma N \rceil}(\vec{x}, 0, \ldots, 0) \equiv \theta^N_{\lceil \sigma N \rceil}\big(\vec{x}, \textstyle\bigwedge_i x_i, \ldots, \bigwedge_i x_i\big),$$

and consequently

$$d \geq c_2(\log n + \log e) \implies \Pr\big[F_{N,d}\big(\vec{x}, \textstyle\bigwedge_i x_i, \ldots, \bigwedge_i x_i\big) \equiv \theta^n_t\big] \geq 1 - 2^{-e}$$

for some constant c_2. We thus define

$$T_{n,t,e,r} = F_{\lfloor \sigma^{-1} n \rfloor, c_2(\log n + \log e)}\big(\vec{x}, \textstyle\bigwedge_i x_i, \ldots, \bigwedge_i x_i\big),$$

where r is the sequence of random coin tosses that determines the leaves of the formula. Notice that $\wedge \in \mathrm{MPT}^k_0 = [\theta^{k+1}_k]$, hence $\bigwedge_i x_i$ can be easily expressed by a $\mathbf{TC}^0$-uniform sequence of $O(\log n)$-depth $\{\theta^{k+1}_k\}$-formulas. It is also straightforward to compute $F_{N,d}$ by a $\mathbf{TC}^0$-function given N, d, and the random sequence r. Finally, $N = \lfloor \sigma^{-1} n \rfloor$ is $\mathbf{TC}^0$-computable by [8]. Thus, $T_{n,t,e,r}$ is computable by a $\mathbf{TC}^0$ function. □

Remark 4.12 The expansion (10) easily implies that for large k, the constant σ_k from Theorem 4.11 is $1 - 2k^{-2} + O(k^{-3})$.

Theorem 4.13 *If $[B] \supseteq \mathrm{PT}^k_\alpha$ for some $k \in \mathbb{N}$ and $\alpha \in \mathbf{2}$, then B-CMP is $\mathbf{\Theta}^{\mathbf{P}}_2$-complete under randomized $\mathbf{TC}^0$-reductions. More precisely, for any language $L \in \mathbf{\Theta}^{\mathbf{P}}_2$, there exists a constant c and a $\mathbf{TC}^0$ function $R(w, e, r)$ with e given in unary such that for all strings w of length n, and for all e,*

$$w \in L \implies \Pr_{|r|=(n+e)^c}[R(w, e, r) \in B\text{-CMP}] = 1, \tag{14}$$

$$w \notin L \implies \Pr_{|r|=(n+e)^c}[R(w, e, r) \in B\text{-CMP}] \leq 2^{-e}. \tag{15}$$

Proof: We will assume $[B] \supseteq \mathrm{T}^k_0$: we may pass from T^k_0 to PT^k_0 in the same way as in the proof of Theorem 4.9, and the case of $[B] \supseteq \mathrm{PT}^k_1$ is dual. Without loss of generality, $k \geq 3$.

We use the reduction from Lemma 3.10 and Theorem 3.11 with the following two modifications:

(i) We express the formulas $x_i \wedge \varphi_i$ by B-formulas.

(ii) In place of the threshold function θ^m_t, we use the randomly generated formula $T_{m,t,e,r}$ from Theorem 4.11.

As for (i), recall that the formulas φ_i supplied by Lemma 3.9 are CNFs, hence they may be arranged to have depth $O(\log n)$. We may assume them to be written in the $\{\wedge, \neg\}$ basis. We then write $x_i \wedge \varphi_i$ in the basis $\{\wedge, \nrightarrow\} \subseteq \mathrm{T}_0^\infty$ by replacing each subformula $\neg\psi$ with $x_i \wedge \neg\psi$. Since $\mathrm{T}_0^\infty \subseteq [B]$, we may rewrite the formulas as B-formulas by Lemma 4.1.

Concerning (ii), notice first that in Lemma 3.10, we may assume n to be sufficiently large, and the parameters we take are $m \approx n^2$, $m - t \approx n$, thus $t > \sigma_k m$, justifying the use of Theorem 4.11. We again rewrite the formulas as B-formulas using Lemma 4.1. Crucially, we have to use the same $T_{m,t,e,r}$ formula for constructing both f_{even} and f_{odd}.

It is clear from the construction that (15) holds, but we need more work to establish (14), as it is not obvious that it holds with no error. If $w \in L$, let $\langle \varphi_i : i < 2n \rangle$ and j be as in Theorem 3.11, so that j is even, and put $s = j/2$. Then in the definition of both $f_{\mathrm{even}} = f_{\varphi_0,\varphi_2,\ldots,\varphi_{2n-2}}$ and $f_{\mathrm{odd}} = f_{\varphi_1,\varphi_3,\ldots\varphi_{2n-1}}$, the first s of the φ_i formulas are satisfiable, and the rest are unsatisfiable. Going back to Lemma 3.10, let us abbreviate by $T(x_0, \ldots, x_{m-1})$ the formula $T_{m,t,e,r}$ we use in place of θ_t^m. Then

$$f_{\mathrm{even}} \equiv T(x_0 \wedge \varphi_0, x_1 \wedge \varphi_2, \ldots, x_{s-1} \wedge \varphi_{2s-2}, \underbrace{0, \ldots, 0}_{n-s}, x_n, \ldots, x_{m-1})$$

and

$$[\nrightarrow, f_{\mathrm{even}}] = [\nrightarrow, T(x_0, \ldots, x_{s-1}, 0, \ldots, 0, x_n, \ldots, x_{m-1})]$$

by the argument in Lemma 3.10 (avoiding renumbering of variables or permuting the arguments of T). Since the same T was also used to construct f_{odd}, we obtain likewise

$$[\nrightarrow, f_{\mathrm{odd}}] = [\nrightarrow, T(x_0, \ldots, x_{s-1}, 0, \ldots, 0, x_n, \ldots, x_{m-1})],$$

hence $[\nrightarrow, f_{\mathrm{even}}] = [\nrightarrow, f_{\mathrm{odd}}]$. □

Notice that if the language L we are reducing to B-CMP is in **BH**, the number n in the proof of Theorem 3.11 can be taken as constant, hence also m and t in Lemma 3.10 are constant, and we may just fix a representation of θ_t^m by a B-formula in advance, avoiding the complicated randomized construction from Theorem 4.11.

Corollary 4.14 *If $[B] \supseteq \mathrm{PT}_\alpha^k$ for some $k \in \mathbb{N}$ and $\alpha \in \mathbf{2}$, then B-CMP is **BH**-hard.* □

We summarize the results of this section:

Corollary 4.15 *Let $B \subseteq \mathrm{Op}$ be finite.*

(*i*) *If $B \subseteq \mathrm{MT}_0^\infty$, MT_1^∞, $\bigwedge$, $\bigvee$, A, or DM, then B-CMP $\in \mathbf{P}$.*

(*ii*) *If $B \subseteq \mathrm{T}_0^\infty$, T_1^∞, or D, but $B \not\subseteq \mathrm{M}$, then B-CMP is **coDP**-complete.*

(*iii*) *If $[B] \supseteq \mathrm{PT}_\alpha^k$ for some $k \in \mathbb{N}$ and $\alpha \in \mathbf{2}$, then B-CMP is $\mathbf{\Theta}_2^\mathbf{P}$-complete under randomized $\mathbf{TC}^0$-reductions, which can be made deterministic if $[B] \supseteq \mathrm{P}$. Also, B-CMP is **BH**-hard.*

(*iv*) *If $\mathrm{MPT}_\alpha^k \subseteq [B] \subseteq \mathrm{M}$ for some $k \in \mathbb{N}$ and $\alpha \in \mathbf{2}$, then B-CMP is in $\mathbf{\Theta}_2^\mathbf{P}$. If $[B] \supseteq \mathrm{MPT}_\alpha^2$, B-CMP is **coDP**-hard.* □

While randomized reductions are a nuisance, the real problem is the last item of Corollary 4.15, where the upper and lower bounds (if any) are far from each other. Notice that since any non-constant monotone function is both 0- and 1-preserving, M-CMP is $\mathbf{TC}^0$-equivalent to MP-CMP, and (using also duality) for any $k \geq 1$, the problems MT_0^k-CMP, MPT_0^k-CMP, MT_1^k-CMP, and MPT_1^k-CMP are $\mathbf{TC}^0$-equivalent.

Problem 4.16 *What is the complexity of* B-CMP *for* $\mathrm{MPT}_\alpha^k \subseteq [B] \subseteq \mathrm{M}$*?*

Our first hunch is that all these problems should be $\mathbf{\Theta}_2^\mathbf{P}$-complete just like their non-monotone versions, but on second thought, it is conceivable that, for example, we can learn some properties of monotone functions by a randomized process such as in the proof of Theorem 4.11, hence the expected answer is not as clear-cut.

We note that Böhler and Schnoor [3] left open a similar problem about the complexity of certain cases of B-CMP_C.

5 Conclusion

We have undertaken a thorough investigation of the complexity of the Boolean clone membership problem CMP and its variants. Most importantly, we proved that CMP is $\mathbf{\Theta}_2^\mathbf{P}$-complete, and in particular, strictly harder than any of the fixed-clone problems CMP_C or CMP^f, barring collapse of the polynomial hierarchy.

Moreover, we obtained a representative (even if incomplete) picture of how the complexity depends on the basis B of gates allowed in the input. As expected, it shows a major dividing line depending on if $[B]$ has finitely many subclones: in the latter case the complexity drops down inside the Boolean hierarchy—in fact, to **coDP**. However, there seems to be also a more subtle dividing line based on if B consists of monotone functions only: in the finite subclone case, this makes the complexity of B-CMP go further down to **P** (though this also happens for some non-monotone cases, namely when B consists of affine functions); in the infinite subclone case, it separates the area of more-or-less $\mathbf{\Theta}_2^\mathbf{P}$-complete instances of B-CMP from a terra incognita.

References

[1] Clifford Bergman, David Juedes, and Giora Slutzki, *Computational complexity of term-equivalence*, International Journal of Algebra and Computation 9 (1999), no. 1, pp. 113–128.

[2] Clifford Bergman and Giora Slutzki, *Complexity of some problems concerning varieties and quasi-varieties of algebras*, SIAM Journal on Computing 30 (2000), no. 2, pp. 359–382.

[3] Elmar Böhler and Henning Schnoor, *The complexity of the descriptiveness of Boolean circuits over different sets of gates*, Theory of Computing Systems 41 (2007), no. 4, pp. 753–777.

[4] Samuel R. Buss, *The Boolean formula value problem is in ALOGTIME*, in: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, pp. 123–131.

[5] Samuel R. Buss and Louise Hay, *On truth-table reducibility to* SAT, Information and Computation 91 (1991), no. 1, pp. 86–102.

[6] Gil Cohen, Ivan Bjerre Damgård, Yuval Ishai, Jonas Kölker, Peter Bro Miltersen, Ran Raz, and Ron D. Rothblum, *Efficient multiparty protocols via log-depth threshold formulae*, in: Advances in Cryptology – CRYPTO 2013. Proceedings, Part II (R. Canetti and J. A. Garay, eds.), Lecture Notes in Computer Science vol. 8043, Springer, 2013, pp. 185–202, full version available at `http://www.wisdom.weizmann.ac.il/~ranraz/publications/Pmajority_mpc-1.pdf`.

[7] Arvind Gupta and Sanjeev Mahajan, *Using amplification to compute majority with small majority gates*, Computational Complexity 6 (1996), no. 1, pp. 46–63.

[8] Emil Jeřábek, *Root finding with threshold circuits*, Theoretical Computer Science 462 (2012), pp. 59–69.

[9] Dexter Kozen, *Lower bounds for natural proof systems*, in: Proceedings of the 18th Annual Symposium on Foundations of Computer Science, 1977, pp. 254–266.

[10] Marcin Kozik, *A finite set of functions with an EXPTIME-complete composition problem*, Theoretical Computer Science 407 (2008), pp. 330–341.

[11] Dietlinde Lau, *Function algebras on finite sets: A basic course on many-valued logic and clone theory*, Springer, New York, 2006.

[12] Thomas Lukasiewicz and Enrico Malizia, *A novel characterization of the complexity class* Θ_k^{P} *based on counting and comparison*, Theoretical Computer Science 694 (2017), pp. 21–33.

[13] Dragan Mašulović, GenClo *and* TermEquiv *are* EXPTIME*-complete*, International Journal of Algebra and Computation 18 (2008), no. 5, pp. 901–909.

[14] Emil L. Post, *The two-valued iterative systems of mathematical logic*, Annals of Mathematics Studies no. 5, Princeton University Press, Princeton, 1941.

[15] Steffen Reith, *Generalized satisfiability problems*, Ph.D. thesis, Julius-Maximilians-Universität, Würzburg, 2001.

[16] Leslie G. Valiant, *Short monotone formulae for the majority function*, Journal of Algorithms 5 (1984), no. 3, pp. 363–366.

[17] Heribert Vollmer, *The complexity of deciding if a Boolean function can be computed by circuits over a restricted basis*, Theory of Computing Systems 44 (2009), no. 1, pp. 82–90.

[18] Klaus W. Wagner, *Bounded query classes*, SIAM Journal on Computing 19 (1990), no. 5, pp. 833–846.