On the number of bipolar Boolean functions

Ringo Baumann and Hannes Strass
Computer Science Institute, Leipzig University, Germany
{baumann,strass}@informatik.uni-leipzig.de

Abstract

A Boolean function is *bipolar* iff it is monotone or antimonotone in each of its arguments. We investigate the number $b(n)$ of n-ary bipolar Boolean functions. We present an (almost) closed-form expression for $b(n)$ that uses the number $a(n)$ of antichain covers of an n-element set. This is closely related to Dedekind's problem, which can be rephrased as determining the number $d(n)$ of Boolean functions that are monotone in all arguments. Indeed, a closed-form solution of $a(n)$ would directly yield a closed-form solution of $d(n)$, suggesting that determining $a(n)$ is a non-trivial problem of itself.

1 Introduction

Computer science makes use of mathematical logic in many ways. A particular recent application of logic in computer science is in the field of abstract argumentation. This field is concerned with modelling (abstractions of) discussions, debates and other forms of human argumentation using mathematical tools. While the predominantly used formalism to date has been the abstract argumentation framework of Dung [4], a number of authors have introduced extensions of that formalism. One such extension is the abstract dialectical framework by Brewka and Woltran [2]. That formalism crucially relies on Boolean functions to express relationships between different positions in a debate. As one of their contributions, Brewka and Woltran introduced a sublanguage of their formalism where only special Boolean functions are allowed, so-called *bipolar* Boolean functions. In a bipolar function, each of its arguments is supporting or attacking.[1] Intuitively, in argumentation, a statement P *supports* another statement Q if it is never the case that accepting the truth of P leads to rejecting the truth of Q. Symmetrically, statement P *attacks* statement Q if accepting the truth of P can never lead to accepting the truth of Q. This definition along with its argumentation-theoretic intuition goes back to the work of Brewka and Woltran [2], who presented it in the context of abstract dialectical frameworks.

Mathematically, supporting and attacking arguments of Boolean functions are simply arguments in which the function is monotone or antimonotone, respectively. In this paper we study the class of Boolean functions that are monotone or antimonotone (or both) in each of their n arguments. In particular, we analyze the cardinality $b(n)$ of this class, where it turns out that there is a close relationship to a combinatorial problem posed by Richard Dedekind in 1897 [3].

The resulting integer sequence $b(n)$ is given by

$$2, 4, 14, 104, 2170, 230540, 499596550, 309075799150640, \ldots$$

and apparently has not received any attention in the literature so far. We newly registered $b(n)$ as a sequence to the online encyclopedia of integer sequences as A245079. While it is somewhat

[1] To prevent confusion: Whenever we write "argument" we mean the argument to a function, as in 2 being the argument to function f in the term $f(2)$. We will not use "argument" in an argumentation sense.

obvious that the number $b(n)$ of bipolar Boolean functions grows considerably with n, we can show that the proportion of bipolar Boolean functions versus all Boolean functions – the quotient $\frac{b(n)}{2^{2^n}}$ – approaches zero as n approaches infinity.

Apart from the purely combinatorial interest in studying the number $b(n)$ of bipolar Boolean functions, there is also a *computational* or more specifically a *representational* significance to our work. The fact that abstract dialectical frameworks (ADFs; the context that we first encountered bipolar Boolean functions in) are intended to be applied in computer software entails that those ADFs have to be stored on computers and therefore represented in some formal language. As storage is not unlimited, expected representation sizes play an important role in assessing the practicality of representation formalisms. A first analysis for ADFs in this regard has been presented by Brewka et al. [1], who used Boolean circuits over the basis $\{\neg, \wedge, \vee\}$ (see, e.g. [7]) for representing Boolean functions. For general, unrestricted ADFs, classic results from circuit complexity theory directly yield lower bounds on representation sizes for "almost all" ADFs: Riordan and Shannon [15] have shown that almost all Boolean functions in n variables require formulas of leafsize $\frac{2^n}{\log_2 n}$; Shannon [16] later showed a lower bound of $\frac{2^n}{n}$ for circuit representation.[2] Moreover, the results imply that whenever we are not interested in *all* Boolean functions in n variables, but only a subclass p of cardinality $p(n)$, then the minimally required representation-size of almost all functions of class p depends in a similar fashion on $p(n)$. Thus having a good lower bound on $b(n)$ directly yields an indication on how large formula or circuit representations of bipolar Boolean functions will be in the worst case.

Moreover, good upper bounds on $b(n)$ can also be meaningfully employed: For the class of *monotone* Boolean functions, there is a range of known constructions that allow for improving the general upper bound when constructing (formula or circuit) realisations, e.g. the "principle of local coding" introduced by Lupanov [13]. Our results relating the numbers of monotone and bipolar Boolean functions suggest that similar techniques might be applicable to realisations of bipolar Boolean functions. In the related setting of showing that bipolar ADFs can succinctly express a language that normal logic programs cannot, we have even made use of an upper-bound result in this vein [18, Theorem 17].

The rest of the paper proceeds as follows. We first give some background and notation on Boolean functions and (anti-)monotonicity properties. In the section thereafter we analyze the number of bipolar Boolean functions. Section 4 gives a closed upper bound for $b(n)$ and shows that the number of bipolar Boolean functions is relatively (in comparison to the total number of Boolean functions) negligible. Then, Section 5 briefly clarifies the relation to Dedekind's problem and relates the number of monotone Boolean functions to the number of bipolar Boolean functions. Section 6 concludes.

2 Background

Let X be a countable set of variables, that is, $X = \{x_1, x_2, \ldots\}$. We denote the set of truth values by $B = \{0, 1\}$. An n-ary *Boolean function* is of the form $f : B^n \to B$, where we assume for simplicity that the arguments of f are $x_1, \ldots, x_n$. Clearly each $M \subseteq \{x_1, \ldots, x_n\}$ induces a two-valued interpretation $v_M : \{x_1, \ldots, x_n\} \to B$ by

$$v_M(s) = \begin{cases} 1 & \text{if } s \in M \\ 0 & \text{otherwise} \end{cases}$$

[2] Here, "almost all" has an arithmetic interpretation: If we say that "almost all" objects from a class $c(n)$ have property p, then this means that the fraction of $c(n)$s that are not ps approaches zero as n approaches infinity.

This in turn yields an input vector $\mathbf{b}_M = (v_M(x_1), \ldots, v_M(x_n))$ to an n-ary Boolean function. We use this fact to sometimes abbreviate $f(M) = f(\mathbf{b}_M) = f(v_M(x_1), \ldots, v_M(x_n))$.

A Boolean function $f : B^n \to B$ is *monotone in argument* x_i iff for all $(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n) \in B^{n-1}$ and $b, b' \in B$, we have

$$b \leq b' \text{ implies } f(b_1, \ldots, b_{i-1}, b, b_{i+1}, \ldots, b_n) \leq f(b_1, \ldots, b_{i-1}, b', b_{i+1}, \ldots, b_n)$$

Symmetrically, a Boolean function $f : B^n \to B$ is *antimonotone in argument* x_i iff for all $(b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n) \in B^{n-1}$ and $b, b' \in B$, we have

$$b \leq b' \text{ implies } f(b_1, \ldots, b_{i-1}, b', b_{i+1}, \ldots, b_n) \leq f(b_1, \ldots, b_{i-1}, b, b_{i+1}, \ldots, b_n)$$

A Boolean function $f : B^n \to B$ is:

- *monotone* iff for all $1 \leq i \leq n$, f is monotone in x_i;
- *antimonotone* iff for all $1 \leq i \leq n$, f is antimonotone in x_i;
- *bipolar* iff for all $1 \leq i \leq n$, f is monotone in x_i or f is antimonotone in x_i.

If a Boolean function f is

- monotone in x_i, we call x_i *supporting* in f;
- antimonotone in x_i, we call x_i *attacking* in f;
- both monotone and antimonotone in x_i, we call x_i *redundant* in f;
- neither monotone nor antimonotone in x_i, we call x_i *dependent* in f.

The Boolean functions $f_1(x_1) = x_1$, $f_2(x_1) = 1 - x_1$, $f_3(x_1) = 1$ and $f_4(x_1, x_2) = x_2$ are examples for functions where the argument x_1 is supporting in f_1, attacking in f_2, redundant in f_3 and dependent in f_4, respectively. If a Boolean function does not possess any dependent arguments we call it *bipolar* which is the object of study of this article.

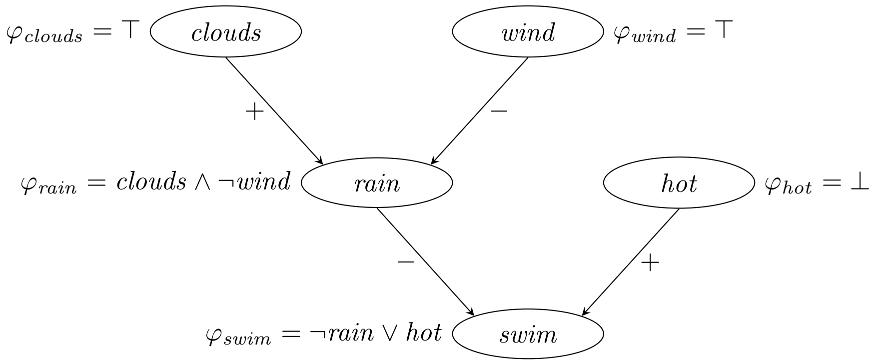
Whereas the terms *monotone* as well as *antimonotone function* are standard notions in order theory, the notion of *supporting* and *attacking arguments* are somewhat new in this context. These terms have an argumentation background. Roughly speaking, in abstract dialectical frameworks, Boolean functions represent acceptance conditions of statements, that is, they express under what conditions a statement can be accepted, given the acceptance status of the statements with a declared influence on the statement. Such an influence (called a *link*), has exactly one of four possible types: A link from r to s can be ...

1. ... supporting. Then accepting r can never lead to rejecting s, all other things being equal.
2. ... attacking. Then accepting r can never lead to accepting s, all other things being equal.
3. ... redundant. Then accepting or rejecting r has no actual influence on whether or not s can be accepted or rejected.
4. ... dependent. Then whether s can be accepted depends not just on r but also on other statements with a declared influence on s.

Note that it is always a *link* that has these properties; it is perfectly possible for a single statement to be attacking in one of its influences and supporting in another. The existence of a "redundant" type also means that influences might be declared, but not actually existing. We further illustrate some of the mentioned notions with a brief detour to abstract dialectical frameworks.

Example 1. An abstract dialectical framework (ADF) $D = (S, L, C)$ consists of a (typically finite) set S of statements, a set $L \subseteq S \times S$ of links, and a family $\left\{C_s : 2^{L^{-1}(s)} \to B\right\}_{s \in S}$ of Boolean functions, exactly one for each $s \in S$.[3] The statements embody propositions that can be accepted by a party in a debate (or not). Links embody directed declared influences between statements; if there is a link from statement r to statement s in an ADF then this means that whether s can be accepted possibly depends on the acceptance of r. Finally, for each $s \in S$, the Boolean function C_s explicitly specifies under what acceptance combinations of L-predecessors of s the statement s can be accepted.

We now present a concrete ADF that we adapted from [2, Example 6]. Consider a scenario where we want to decide whether we go for a *swim*. We do so if there is no *rain*, or it is *hot*. It is warm, but not hot, and there are *clouds* indicating that it might rain. However the reliable weather forecast predicts *wind* that will blow away the clouds. Using the vocabulary $S = \{clouds, wind, rain, hot, swim\}$, we devise the ADF $D_{swim} = (S, L, C)$ shown below to model this deliberation process. Here, statements are depicted as nodes, edges represent links and acceptance conditions are written as propositional formulas next to the statements.



Supporting and attacking links are designated using the labels + and −; this is however only for illustration as the polarity of the links can be read off the acceptance formulas. More precisely, the link (r, s) is supporting iff r is supporting in the Boolean function C_s, the acceptance function of statement s; likewise for the other polarities. The statement *rain*, for example, is supported by the statement *clouds* and attacked by the statement *wind*. According to φ_{rain}, the attack from *wind* is stronger than the support from *clouds*. That is, as soon as we accept *wind*, we must reject *rain*. On the other hand, *swim* is attacked by rain and supported by hot. Here, by φ_{swim}, the support from *hot* is stronger than the attack from *rain*; or put another way, the missing attack from *rain* is stronger than the missing support from *hot*. This effectively means that rejecting *rain* leads to accepting *swim*.

Note that monotone Boolean functions in n arguments can be equivalently characterized thus: for any $\mathbf{b}, \mathbf{b}' \in B^n$, if $\mathbf{b} \leq \mathbf{b}'$, then $f(\mathbf{b}) \leq f(\mathbf{b}')$ where $\mathbf{b} = (b_1, \ldots, b_n)$, $\mathbf{b}' = (b'_1, \ldots, b'_n)$ and $\mathbf{b} \leq \mathbf{b}'$ iff $b_i \leq b'_i$ for all $1 \leq i \leq n$. The property of being supporting/attacking etc. will be called the *polarity* of an argument. Since supporting/attacking arguments might be redundant, we use the prefix *strictly* to exclude this; that is, an argument is *strictly supporting* iff it is supporting and not attacking, symmetrically an argument is *strictly attacking* iff it is attacking and not supporting.

We denote the set of all Boolean functions in n arguments by $\mathcal{B}_n = \{f : B^n \to B\}$. Furthermore, for $s, a, r, d \in \mathbb{N}$, we denote by $\mathcal{B}_n(s, a, r, d)$ the set of Boolean functions in n arguments where

[3] By $L^{-1}(s) = \{r \in S \mid (r, s) \in L\}$ we denote the set of L-predecessors of s.

exactly s arguments are supporting, a arguments are attacking, r arguments are redundant and d arguments are dependent. Note that in this case $n = s + a - r + d$ since redundant arguments are supporting *and* attacking. So for example, $\mathcal{B}_n(k, 0, 0, n-k)$ denotes the set of Boolean functions that are supporting in exactly k arguments where none of the arguments is attacking (thus, not redundant). $\mathcal{B}_n(n, k, k, 0)$ is the set of monotone Boolean functions: all arguments are supporting, but some $k \leq n$ of them might also be attacking and thus redundant.

3 The number of bipolar Boolean functions

How many of the 2^{2^n} Boolean functions $f : B^n \to B$ are bipolar? To tackle this problem we present three technical lemmata paving the way for the main theorem. The first lemma shows the relation between monotone and antimonotone arguments in a Boolean function f and its so-called *i-negation.*

Definition 1. For a Boolean function $f : B^n \to B$ and an $1 \leq i \leq n$ we define its *i-negation* as the function $f_{-i} : B^n \to B$ given by

$$f_{-i}(b_1, \ldots, b_{i-1}, b_i, b_{i+1}, \ldots, b_n) = f(b_1, \ldots, b_{i-1}, 1 - b_i, b_{i+1}, \ldots, b_n).$$

Intuitively, f_{-i} is obtained from f by negating the i-th input argument. For example, if $f : B \times B \to B$ is the material implication function given by $f(x_1, x_2) = \min\{(1 - x_1) + x_2, 1\}$ then the function $f_{-1} : B \times B \to B$ is the logical disjunction $f(x_1, x_2) = \min\{x_1 + x_2, 1\}$. In general, the i-negation has no effect on the polarity of all arguments x_j with $j \neq i$, and the effect of negating the polarity of x_i.

Lemma 1. *Let $f : B^n \to B$ and $1 \leq i, j \leq n$ with $i \neq j$.*

1. *f is monotone in x_j if and only if f_{-i} is monotone in x_j.*
2. *f is antimonotone in x_j if and only if f_{-i} is antimonotone in x_j.*
3. *f is monotone in x_i if and only if f_{-i} is antimonotone in x_i.*
4. *f is antimonotone in x_i if and only if f_{-i} is monotone in x_i.*

Proof. 1. ($\Rightarrow$) *Let f be monotone in x_j, assume w.l.o.g. that $i \leq j$ and assume to the contrary of what we have to show that there are $b_1, \ldots, b_{j-1}, b_{j+1}, \ldots, b_n, b, b' \in B$ with $b \leq b'$ and*

$$\begin{aligned} &f_{-i}(b_1, \ldots, b_{i-1}, b_i, b_{i+1}, \ldots, b_{j-1}, b, b_{j+1}, \ldots, b_n) \\ &\qquad > f_{-i}(b_1, \ldots, b_{i-1}, b_i, b_{i+1}, \ldots, b_{j-1}, b', b_{j+1}, \ldots, b_n) \end{aligned}$$

But then, for $b'_i = 1 - b_i$, we find that

$$\begin{aligned} &f(b_1, \ldots, b_{i-1}, b'_i, b_{i+1}, \ldots, b_{j-1}, b, b_{j+1}, \ldots, b_n) \\ = {}& f_{-i}(b_1, \ldots, b_{i-1}, 1 - b'_i, b_{i+1}, \ldots, b_{j-1}, b, b_{j+1}, \ldots, b_n) && \textit{(Def. 1)} \\ = {}& f_{-i}(b_1, \ldots, b_{i-1}, b_i, b_{i+1}, \ldots, b_{j-1}, b, b_{j+1}, \ldots, b_n) && (b'_i = 1 - b_i) \\ > {}& f_{-i}(b_1, \ldots, b_{i-1}, b_i, b_{i+1}, \ldots, b_{j-1}, b', b_{j+1}, \ldots, b_n) && (f_{-i} \textit{ is not monotone in } x_j) \\ = {}& f_{-i}(b_1, \ldots, b_{i-1}, 1 - b'_i, b_{i+1}, \ldots, b_{j-1}, b', b_{j+1}, \ldots, b_n) && (b'_i = 1 - b_i) \\ = {}& f(b_1, \ldots, b_{i-1}, b'_i, b_{i+1}, \ldots, b_{j-1}, b', b_{j+1}, \ldots, b_n) && \textit{(Def. 1)} \end{aligned}$$

Thus f is not monotone in x_j. Contradiction.

($\Leftarrow$) *Since f_{-i} is monotone in x_j we derive $(f_{-i})_{-i}$ is monotone in x_j. The equation $(f_{-i})_{-i} = f$ proves the assertion.*

2. *Analogous.*

3. and 4. *In order to prove both statements it suffices to show first the only-if-directions of both statements and second to apply the equality $(f_{-i})_{-i} = f$.*

($\Rightarrow$) *Both only-if-directions can be proven in a similar way. We show the case of statement 3 only. Let f be monotone in x_i. Consider $b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n \in B$ and $b, b' \in B$ with $b \leq b'$. We have to show that $f_{-i}(b_1, \ldots, b_{i-1}, b', b_{i+1}, \ldots, b_n) \leq f_{-i}(b_1, \ldots, b_{i-1}, b, b_{i+1}, \ldots, b_n)$. If $b = b'$ then $f_{-i}(b_1, \ldots, b_{i-1}, b', b_{i+1}, \ldots, b_n) = f_{-i}(b_1, \ldots, b_{i-1}, b, b_{i+1}, \ldots, b_n)$, so let $b < b'$. Clearly $b = 0$, $b' = 1$ and we obtain*

$$\begin{aligned}
& f_{-i}(b_1, \ldots, b_{i-1}, b', b_{i+1}, \ldots, b_n) & \\
&= f(b_1, \ldots, b_{i-1}, 1 - b', b_{i+1}, \ldots, b_n) & \text{(Def. 1)} \\
&= f(b_1, \ldots, b_{i-1}, 0, b_{i+1}, \ldots, b_n) & (b' = 1) \\
&\leq f(b_1, \ldots, b_{i-1}, 1, b_{i+1}, \ldots, b_n) & (f \text{ is monotone in } x_i) \\
&= f(b_1, \ldots, b_{i-1}, 1 - b, b_{i+1}, \ldots, b_n) & (b = 0) \\
&= f_{-i}(b_1, \ldots, b_{i-1}, b, b_{i+1}, \ldots, b_n) & \text{(Def. 1)}
\end{aligned}$$

($\Leftarrow$) *Now, for statement 3. Let f_{-i} be antimonotone in x_i. Applying the only-if-direction of statement 4 yields that $(f_{-i})_{-i}$ is monotone in x_i. Using $(f_{-i})_{-i} = f$ concludes statement 3. The if-direction of statement 4 can be shown analogously.* □

In the following we use the abbreviation $\underline{m} = \{n_1, n_2, \ldots, n_m\}$ for an m-element set of natural numbers. Building upon this, for a Boolean function $f : B^n \to B$ we denote by

$$f_{-\underline{m}} = (\ldots((f_{-n_1})_{-n_2})\ldots)_{-n_m}$$

the repeated i-negation for all $i \in \underline{m}$. The following result shows that this notation is justified because the order in which these i-negations are applied does not matter. Furthermore, applying i-negation twice is the identity operation since $1 - (1 - b) = b$ for all $b \in B$. Finally, when taking two Boolean functions where all arguments are strictly supporting (that is, a non-degenerate monotone Boolean function) and repeatedly applying i-negations to them, this process leads to distinct Boolean functions if and only if we started out with different functions, or manipulated them differently.

Proposition 2. *Let $f : B^n \to B$ be a Boolean function.*

1. *For any $\underline{m} \subseteq \{1, \ldots, n\}$ and any permutation $\pi : \{1, \ldots, m\} \to \{1, \ldots, m\}$,*

$$(\ldots((f_{-n_1})_{-n_2})\ldots)_{-n_m} = (\ldots((f_{-n_{\pi(1)}})_{-n_{\pi(2)}})\ldots)_{-n_{\pi(m)}} \qquad \text{(commutativity)}$$

2. *For any $\underline{m} \subseteq \{1, \ldots, n\}$, $(f_{-\underline{m}})_{-\underline{m}} = f$.* *(neutrality)*

3. *Let $f, g \in \mathcal{B}_n(n, 0, 0, 0)$, $\underline{m}, \underline{k} \subseteq \{1, \ldots, n\}$. If $f \neq g$ or $\underline{m} \neq \underline{k}$, then $f_{-\underline{m}} \neq g_{-\underline{k}}$.* *(injectivity)*

Proof. 1. *Obvious (cf. Def. 1).*

2. *Obvious (cf. Def. 1).*
3. *Let $f, g \in \mathcal{B}_n(n,0,0,0)$ and $\underline{m}, \underline{k} \subseteq \{1,\dots,n\}$ such that $f \neq g$ or $\underline{m} \neq \underline{k}$. We do a case distinction whether $\underline{m} = \underline{k}$.*
   - *$\underline{m} \neq \underline{k}$. W.l.o.g. let $i \in \underline{m} \setminus \underline{k}$. Since $f \in \mathcal{B}_n(n,0,0,0)$, x_i is strictly supporting in f, i.e. f is monotone and is not antimonotone in x_i. Applying Lemma 1 implies that x_i is antimonotone (statement 3) and monotone (statement 4) in $f_{-\underline{m}}$. Compositely, x_i is strictly attacking in $f_{-\underline{m}}$. Similarly, since $i \notin \underline{k}$ we may apply the first two statements of Lemma 1 showing that x_i is still strictly supporting in $g_{-\underline{k}}$. Consequently, $f_{-\underline{m}} \neq g_{-\underline{k}}$.*
   - *$\underline{m} = \underline{k}$. Then $f \neq g$. Assume to the contrary that $f_{-\underline{m}} = g_{-\underline{k}}$. But then*

$$f = \left(f_{-\underline{m}}\right)_{-\underline{m}} = \left(g_{-\underline{k}}\right)_{-\underline{m}} = \left(g_{-\underline{k}}\right)_{-\underline{k}} = g$$

   Contradiction. Thus $f_{-\underline{m}} \neq g_{-\underline{k}}$. □

The properties shown by this proposition are instrumental in proving the following useful lemma. It asserts that, intuitively, there are two orthogonal dimensions along which (strict) bipolar Boolean functions can be constructed: first, the polarity of their arguments, that is, the choice whether a particular argument will be (strictly) supporting or attacking; second, the underlying logical relationships between the arguments, that make up the essence of the function in the end.

Lemma 3. *Let $s, a, n \in \mathbb{N}$ such that $s + a = n$.*

$$|\mathcal{B}_n(s,a,0,0)| = \binom{n}{a} \cdot |\mathcal{B}_n(n,0,0,0)|$$

Proof. We define the mapping

$$\varphi : \mathcal{B}_n(n,0,0,0) \times \binom{\{1,\dots,n\}}{a} \to \mathcal{B}_n(s,a,0,0) \qquad \text{with} \qquad (f, \underline{a}) \mapsto f_{-\underline{a}}$$

and show that it is a bijection. Clearly the mapping is well-defined, since for $f \in \mathcal{B}_n(n,0,0,0)$ and $\underline{a} \in \binom{\{1,\dots,n\}}{a}$, we have $f_{-\underline{a}} \in \mathcal{B}_n(s,a,0,0)$ by Lemma 1.

φ is injective: *For $f, g \in \mathcal{B}_n(n,0,0,0)$ and $\underline{m}, \underline{k} \subseteq \{1,\dots,n\}$ with $f \neq g$ or $\underline{m} \neq \underline{k}$, it follows directly from Proposition 2 that $f_{-\underline{m}} \neq g_{-\underline{k}}$.*

φ is surjective: *Let $f \in \mathcal{B}_n(s,a,0,0)$ and denote by $\underline{a}$ the set of indices whose arguments are attacking in f. Then clearly $f_{-\underline{a}} \in \mathcal{B}_n(n,0,0,0)$, and furthermore*

$$\varphi(f_{-\underline{a}}, \underline{a}) = \left(f_{-\underline{a}}\right)_{-\underline{a}} = f$$

In combination, we get

$$\begin{aligned} |\mathcal{B}_n(s,a,0,0)| &= \left|\mathcal{B}_n(n,0,0,0) \times \binom{\{1,\dots,n\}}{a}\right| \\ &= |\mathcal{B}_n(n,0,0,0)| \cdot \left|\binom{\{1,\dots,n\}}{a}\right| \\ &= \binom{n}{a} \cdot |\mathcal{B}_n(n,0,0,0)| \end{aligned}$$

□

The next and final lemma asserts that any bipolar Boolean function can be stripped of its redundant arguments essentially without having an impact on the number of distinct functions considering only the non-redundant arguments. Let us consider an illustration.

Example 2. Let $f \in \mathcal{B}_2(2,1,1,0)$ be given by $f(x_1, x_2) = x_1$ where one argument (x_1) is strictly supporting, and the other one is redundant. Clearly there is an "equivalent" function $g \in \mathcal{B}_1(1,0,0,0)$ in only one argument, given by $g(x_1) = x_1$.

Lemma 4. *Let $s, a, r, n \in \mathbb{N}$ with $s + a - r = n$.*

$$|\mathcal{B}_n(s,a,r,0)| = \binom{n}{r} \cdot |\mathcal{B}_{n-r}(s-r, a-r, 0, 0)|$$

Proof. Let $\underline{n} = \{1, \dots, n\}$ and $\underline{k} \subseteq \underline{n}$. For $(b_1, \dots, b_{n-k}) \in B^{n-k}$ denote by $(b_1, \dots, b_{n-k}) \otimes 0^{\underline{k}} = (c_1, \dots, c_n)$ where for $1 \leq i \leq n$ we set

$$c_i = \begin{cases} 0 & \text{if } i \in \underline{k} \\ b_{i - |\{j \in \underline{k} \mid j < i\}|} & \text{otherwise} \end{cases}$$

For a Boolean function $f \in \mathcal{B}_n$ we now define

$$f_{\underline{n} \setminus \underline{k}} : B^{n-k} \to B \qquad \text{with} \qquad (b_1, \dots, b_{n-k}) \mapsto f((b_1, \dots, b_{n-k}) \otimes 0^{\underline{k}})$$

Now let $s, a, r, n \in \mathbb{N}$ with $s + a - r = n$. We define the mapping

$$\psi : \mathcal{B}_n(s,a,r,0) \to \left(\mathcal{B}_{n-r}(s-r, a-r, 0, 0) \times \binom{\{1, \dots, n\}}{r} \right) \qquad \text{with} \qquad f \mapsto \left(f_{\underline{n} \setminus \underline{r}_f}, \underline{r}_f \right)$$

where $\underline{r}_f$ denotes the set of indices whose arguments are redundant in f. We proceed to show that ψ is bijective.

ψ is injective: *Let $f, g \in \mathcal{B}_n(s,a,r,0)$ with $f \neq g$. Then there are $b_1, \dots, b_n \in B$ such that $f(b_1, \dots, b_n) \neq g(b_1, \dots, b_n)$. By definition of $\underline{r}_f$ and $\underline{r}_g$, we also have that*

$$f((b_1, \dots, b_n)[b_i/0 : i \in \underline{r}_f]) = f(b_1, \dots, b_n) \neq g(b_1, \dots, b_n) = g((b_1, \dots, b_n)[b_i/0 : i \in \underline{r}_g])$$

Assume to the contrary of what we have to show that $(f_{\underline{n} \setminus \underline{r}_f}, \underline{r}_f) = (g_{\underline{n} \setminus \underline{r}_g}, \underline{r}_g)$. Then in particular $f_{\underline{n} \setminus \underline{r}_f} = g_{\underline{n} \setminus \underline{r}_g}$. It follows that for all $(c_1, \dots, c_{n-r}) \in B^{n-r}$, we get

$$\begin{aligned} & f((c_1, \dots, c_{n-r}) \otimes 0^{\underline{r}_f}) \\ = {} & f_{\underline{n} \setminus \underline{r}_f}(c_1, \dots, c_{n-r}) \\ = {} & g_{\underline{n} \setminus \underline{r}_g}(c_1, \dots, c_{n-r}) \\ = {} & g((c_1, \dots, c_{n-r}) \otimes 0^{\underline{r}_g}) \end{aligned}$$

In particular,

$$f((b_1, \dots, b_n)[b_i/0 : i \in \underline{r}_f]) = g((b_1, \dots, b_n)[b_i/0 : i \in \underline{r}_g])$$

Contradiction. Thus $(f_{\underline{n} \setminus \underline{r}_f}, \underline{r}_f) \neq (g_{\underline{n} \setminus \underline{r}_g}, \underline{r}_g)$.

ψ is surjective: *Let $f \in \mathcal{B}_{n-r}(s-r, a-r, 0, 0)$ and $\underline{r} \subseteq \{1, \dots, n\}$. Define*

$$g : B^n \to B \qquad \textit{with} \qquad (b_1, \dots, b_n) \mapsto f(b_{i_1}, \dots, b_{i_{n-r}})$$

where $i_1, \dots, i_{n-r} \in \{1, \dots, n\} \setminus \underline{r}$ such that $i_1 < i_2 < \dots < i_{n-r}$. Then clearly $\underline{r} = \underline{r}_g$ and thus $\psi(g) = (g_{\underline{n} \setminus \underline{r}_g}, \underline{r}_g) = (g_{\underline{n} \setminus \underline{r}}, \underline{r}) = (f, \underline{r})$. $\square$

Now we are prepared to turn to the main theorem.

Theorem 5. *The number of bipolar Boolean functions in n arguments is*

$$b(n) = \sum_{i=0}^{n} 2^i \cdot \binom{n}{i} \cdot |\mathcal{B}_i(i, 0, 0, 0)|$$

Proof. Consider the following equalities. Afterwards we will clarify every single step. As a notational shorthand we use $\mathcal{B}_n(*, *, n-i, 0)$ for $\bigcup_{s+a-(n-i)=n} \mathcal{B}_n(s, a, n-i, 0)$, i.e., all possible (s, a)-combinations of supporting and attacking arguments for bipolar Boolean functions in n arguments with i non-redundant arguments.

$$\begin{aligned}
b(n) &= \sum_{i=0}^{n} |\mathcal{B}_n(*, *, n-i, 0)| && (1)\\
&= \sum_{i=0}^{n} \left(\sum_{s+a-(n-i)=n} |\mathcal{B}_n(s, a, n-i, 0)| \right) && (2)\\
&= \sum_{i=0}^{n} \left(\sum_{s+a-(n-i)=n} \binom{n}{n-i} \cdot \left|\mathcal{B}_{n-(n-i)}(s-(n-i), a-(n-i), 0, 0)\right| \right) && (3)\\
&= \sum_{i=0}^{n} \left(\binom{n}{n-i} \sum_{s+a-(n-i)=n} \cdot |\mathcal{B}_i(s-(n-i), a-(n-i), 0, 0)| \right) && (4)\\
&= \sum_{i=0}^{n} \binom{n}{i} \cdot |\mathcal{B}_i(*, *, 0, 0)| && (5)\\
&= \sum_{i=0}^{n} \binom{n}{i} \cdot \left(\sum_{a=0}^{i} |\mathcal{B}_i(i-a, a, 0, 0)| \right) && (6)\\
&= \sum_{i=0}^{n} \binom{n}{i} \cdot \left(\sum_{a=0}^{i} \binom{i}{a} \cdot |\mathcal{B}_i(i, 0, 0, 0)| \right) && (7)\\
&= \sum_{i=0}^{n} 2^i \cdot \binom{n}{i} \cdot |\mathcal{B}_i(i, 0, 0, 0)| && (8)
\end{aligned}$$

(1) Bipolar functions can be distinguished by their number of redundant arguments.
(2) Apply the definition of $\mathcal{B}_n(*, *, n-i, 0)$.
(3) Apply Lemma 4.
(4) Law of distribution and simplification.
(5) Use $\binom{n}{n-i} = \binom{n}{i}$ and since $s+a-(n-i) = n$ iff $s-(n-i)+a-(n-i) = i$ we may

stick to notational shorthand.
(6) Strictly bipolar Boolean functions can be distinguished by their numbers of strictly supporting and strictly attacking arguments.
(7) Apply Lemma 3.
(8) Use $\sum_{a=0}^{i} \binom{i}{a} \cdot |\mathcal{B}_i(i,0,0,0)| = 2^i \cdot \sum_{a=0}^{i} |\mathcal{B}_i(i,0,0,0)|$.

□

The initial numbers for $b(n)$ are:

n	$b(n)$
0	2
1	4
2	14
3	104
4	2170
5	230540
6	499596550
7	309075799150640

Observe that Theorem 5 implies that the problem of determining the number of bipolar Boolean functions in n arguments can be reduced to the problem determining the number of Boolean functions in n arguments where all arguments are strictly supporting, so-called *nondegenerate* monotone Boolean functions. According to the online encyclopedia of integer sequences at `http://oeis.org/A006126`, Rodrigo Obando observed that this number in turn coincides with the number $a(n)$ of antichain covers of an n-element set.[4] In our terminology, this is formulated as follows:

Proposition 6. *For any $n \in \mathbb{N}$,*

$$|\mathcal{B}_n(n,0,0,0)| = a(n)$$

Proof. It is well-known that antichains correspond to monotone Boolean functions. Consider therefore $\Phi : a\mathcal{C}_n \to \mathcal{B}_n(n,*,*,0)$ where $a\mathcal{C}_n$ abbreviates the set of all $\subseteq$-antichains in the n-element set $V = \{x_1,\ldots,x_n\}$. For any $\mathcal{A} = (A_i)_{i\in I} \in a\mathcal{C}$ we define the monotone Boolean function $\Phi(\mathcal{A}) = f_{\mathcal{A}}$ via

$$f_{\mathcal{A}}(\mathbf{b}) = f_{\mathcal{A}}(b_1,\ldots,b_n) = 1 \text{ iff } \exists i \in I, \text{ s.t. } A_i \subseteq X^{b=1} \text{ where } X^{b=1} = \{x_j \mid b_j = 1\}$$

Obviously, Φ is injective. To see that Φ is even surjective consider any $f \in \mathcal{B}_n(n,*,*,0)$. Define

$$\mathcal{A}_f = \{A \mid A \text{ is } \subseteq\text{-minimal in } \mathcal{X}\} \text{ where } \mathcal{X} = \{X^{b=1} \mid f(b) = 1\}$$

We have $\Phi(\mathcal{A}_f) = f$ and thus, Φ is shown to be a bijection. We will show now that the restriction of Φ to antichain covers matches the set of nondegenerate monotone Boolean functions $\mathcal{B}_n(n,0,0,0)$.

Let $f \in \mathcal{B}_n(n,0,0,0)$. We will show that $\mathcal{A}_f$ is an antichain cover. Since f is nondegenerate we derive: for any argument x_i exists a witnessing vector $b_w = (b_1,\ldots,b_{i-1},b_{i+1},\ldots,b_n) \in B^{n-1}$ such that

$$0 = f(b_w^0) = f(b_1,\ldots,b_{i-1},0,b_{i+1},\ldots,b_n) < f(b_1,\ldots,b_{i-1},1,b_{i+1},\ldots,b_n) = f(b_w^1) = 1$$

[4] The antichain cover interpretation of A006126 was obtained by Kilibarda and Jovović [8].

W.l.o.g. we may further assume that the witnessing vector is componentwise-minimal since there are only finitely many which have to be considered. We deduce that for b_w^1 , $X^{b_w^1=1} \in \mathcal{X}$ and in particular, $x_i \in X^{b_w^1=1}$. We show that $X^{b_w^1=1}$ is $\subseteq$-minimal in $\mathcal{X}$ and thus $x_i \in \bigcup_{A\in\mathcal{A}_f} A$ is guaranteed. Assume, to derive a contradiction, that there is some $X' \in \mathcal{X}$ with $X' \subsetneq X^{b_w^1=1}$. Thus, there is a $b' = (b'_1, \ldots, b'_{i-1}, b'_i, b'_{i+1}, \ldots, b'_n) \in B^n$ with $b' < b_w^1$ and $f(b') = 1$. We proceed with a case distinction.

1. *Let $b'_i = 0$. Then $b' \leq b_w^0$ but $f(b') = 1 \not\leq 0 = f(b_w^0)$ in contrast to the monotonicity of f.*
2. *Let $b'_i = 1$ and assume $f(b'_1, \ldots, b'_{i-1}, 0, b'_{i+1}, \ldots, b'_n) = 0$. Since $b' < b_w^1$ is already deduced we infer that ${b_w}' = (b'_1, \ldots, b'_{i-1}, b'_{i+1}, \ldots, b'_n) < b_w$. Note that b'_w is a witnessing vector for non-degeneracy in contrast to the assumed componentwise minimality of the witnessing vector b_w.*
3. *Let $b'_i = 1$ and assume $f(b'_1, \ldots, b'_{i-1}, 0, b'_{i+1}, \ldots, b'_n) = f({b'_w}^0) = 1$. Since we have $b' < b_w^1$ we derive ${b'_w}^0 \leq b_w^0$. On the other hand, $f({b'_w}^0) = 1 \not\leq 0 = f(b_w^0)$ in contrast to the monotonicity of f.*

Altogether, for any nondegenerate monotone function f in n arguments, $x_i \in \bigcup_{A\in\mathcal{A}_f} A$ for any $1 \leq i \leq n$. This means $\mathcal{A}_f$ is an antichain cover.

It remains to show that the non-degeneracy of $\Phi(\mathcal{A}) = f_\mathcal{A}$ is guaranteed whenever $\mathcal{A}$ is an antichain cover. Assume to the contrary that $f_\mathcal{A}$ is degenerate. This means that there is an x_i such that $f_\mathcal{A}$ is supporting and attacking in x_i, i.e. for all $b = (b_1, \ldots, b_{i-1}, b_{i+1}, \ldots, b_n) \in B^{n-1}$ we have:

$$f_\mathcal{A}(b^0) = f_\mathcal{A}(b_1, \ldots, b_{i-1}, 0, b_{i+1}, \ldots, b_n) = f_\mathcal{A}(b_1, \ldots, b_{i-1}, 1, b_{i+1}, \ldots, b_n) = f_\mathcal{A}(b^1)$$

Since $\mathcal{A}$ is an antichain cover we derive the existence of an $A \in \mathcal{A}$ such that $x_i \in A$. Consequently, $f_\mathcal{A}(v_A(x_1), \ldots, v_A(x_{i+1}), 1, v_A(x_{i+1}), \ldots, v_A(x_n)) = 1$. In consideration of the equation above,

$$f_\mathcal{A}(v_A(x_1), \ldots, v_A(x_{i+1}), 0, v_A(x_{i+1}), \ldots, v_A(x_n)) = 1$$

This in turn enforces the existence of an $A' \in \mathcal{A}$, s.t. $A' \subsetneq A$ in contrast to the antichain property of $\mathcal{A}$. □

The first 8 values of this sequence (A006126) are given here:

n	$a(n)$
1	1
2	2
3	9
4	114
5	6894
6	7785062
7	2414627396434
8	56130437209370320359966

We additionally set $a(0) = 2$, since there are two null-ary Boolean functions where all of the arguments are non-redundant, namely the constant functions 0 and 1.

As a direct consequence, we get a closed expression for $b(n)$ that depends only on $a(i)$ for $0 \leq i \leq n$.

Corollary 7.

$$b(n) = \sum_{i=0}^{n} 2^i \cdot \binom{n}{i} \cdot a(i)$$

4 A closed expression for the upper bound

While it seems difficult to provide a closed-form expression for $b(n)$, we can present an upper bound.

Proposition 8. *For any $n \in \mathbb{N}$ we have*

$$b(n) \leq 2 \cdot 3^{2^{n-1}} - 2^{2^{n-1}}$$

Proof. *Let f be a bipolar Boolean function in n arguments. Consequently, for any $1 \leq i \leq n$ we have, x_i is supporting or attacking. In particular, x_1 is supporting or attacking. For any $M \subseteq \{x_2, \ldots, x_n\}$ we may consider the following two values $f(1, v_M(x_2), \ldots, v_M(x_n))$ and $f(0, v_M(x_2), \ldots, v_M(x_n))$ abbreviated by f_M^1 or f_M^0. Let us further define $f_M = (f_M^1, f_M^0)$, $\bar{0} = (0,0)$, $\bar{1} = (1,1)$, $C_{sup} = (0,1)$ and $C_{att} = (1,0)$. Observe that if for a certain M, $f_M = C_{sup}$ or $f_M = C_{att}$, then f cannot be monotone or antimonotone, respectively since x_1 serves as a counterexample. Consequently, for any bipolar Boolean function f in n arguments we have that for all $M \subseteq \{x_2, \ldots, x_n\}$ either $f_M \in \{\bar{0}, \bar{1}, C_{sup}\}$ or $f_M \in \{\bar{0}, \bar{1}, C_{att}\}$. Obviously, there are 2^{n-1} subsets M. Furthermore, in both cases we may choose between 3 elements. Thus, there are $2 \cdot 3^{2^{n-1}}$ possibilities. Note that functions f, s.t. for any M, f_M returns $\bar{0}$ or $\bar{1}$ are included in both cases. Thus, we have to subtract this number, namely $2^{2^{n-1}}$, and we are done. This means that we may estimate the number of bipolar Boolean functions in n arguments by $2 \cdot 3^{2^{n-1}} - 2^{2^{n-1}}$.* $\square$

We want to mention that the given upper bound coincides with $b(n)$ if $n \in \{1, 2\}$:

$$2 \cdot 3^{2^0} - 2^{2^0} = 6 - 2 = 4 = b(1) \qquad \text{and} \qquad 2 \cdot 3^{2^1} - 2^{2^1} = 2 \cdot 9 - 4 = 14 = b(2)$$

We also observe that $b(1) = 4 = 2^{2^1}$ (all unary Boolean functions are bipolar) and $b(2) = 14 = 2^{2^2} - 2$ (only two binary Boolean functions are not bipolar).[5] In consideration of this observation the reader might get the impression that the number of bipolar functions account for a major proportion of the number of Boolean functions in general. The following proposition shows that this is not the case. To the contrary, the possibility that a randomly chosen Boolean function is bipolar approaches zero. This also shows that the bound given in Proposition 8 is non-trivial.

Proposition 9.

$$\lim_{n \to \infty} \frac{b(n)}{2^{2^n}} = 0$$

[5] The two non-bipolar functions in two arguments are equivalence ("$\leftrightarrow$") as well as antivalence ("$\not\leftrightarrow$").

Proof. *By Proposition 8 and standard fractional arithmetic we have*

$$\frac{b(n)}{2^{2^n}} \leq \frac{2 \cdot 3^{2^{n-1}} - 2^{2^{n-1}}}{2^{2^n}} \tag{9}$$

$$= \frac{2 \cdot 3^{2^{n-1}}}{2^{2^{n-1}} \cdot 2^{2^{n-1}}} - \frac{2^{2^{n-1}}}{2^{2^{n-1}} \cdot 2^{2^{n-1}}} \tag{10}$$

$$= \frac{2 \cdot 3^{2^{n-1}}}{4^{2^{n-1}}} - \frac{1}{2^{2^{n-1}}} \tag{11}$$

$$= 2 \cdot \left(\frac{3}{4}\right)^{2^{n-1}} - \left(\frac{1}{2}\right)^{2^{n-1}} \tag{12}$$

Consequently, $\lim_{n\to\infty} \frac{b(n)}{2^{2^n}} = 0$ *concluding the proof.* □

As already mentioned the presented upper bound coincides with $b(n)$ if $n \in \{1, 2\}$. This is not surprising for $n = 1$ since we counted the number of functions being attacking or supporting in x_1, thus being bipolar. In case of $n = 2$ one may easily show that being attacking or supporting in x_1 enforces non-dependency in x_2. The following example shows that this property does not carry over to $n \geq 3$.

Example 3. Consider $f : B^3 \to B$ where

$$f(x) = \begin{cases} 1 & \text{if } x \in \{(1,1,1),(1,0,0)\} \\ 0 & \text{otherwise} \end{cases}$$

The argument x_1 is supporting since whenever $a \leq a'$, then $f(a,b,c) \leq f(a',b,c)$. This means, the function f is one of the $2 \cdot 3^{2^{3-1}} - 2^{2^{3-1}}$ functions representing the upper bound in Proposition 8. Nevertheless, observe that $f \notin b_3$ since x_2 is dependent. Counterexamples for being attacking and supporting are given by $1 = f(1,1,1) \not\leq f(1,0,1) = 0$ or $1 = f(1,0,0) \not\leq f(1,1,0) = 0$, respectively.

5 Relation to Dedekind numbers

Dedekind [3] presented the problem of determining the number $d(n)$ of monotone Boolean functions in n arguments. This problem is a special case of the bipolar counting problem and has attracted the attention of combinatorial analysts for almost 120 years. Numerous authors have tackled the problem, giving asymptotic estimations, exact values for small n, recursive equations or algorithms (see [9] for an overview). In the following we briefly summarize some milestones concerning asymptonic estimates of $d(n)$.

1. In 1954 by Gilbert [5],
$$\binom{n}{n/2} \leq \log_2 d(n) \leq \log_2 n \binom{n}{n/2}$$
2. Korobkov [11] improved this upper upper bound to
$$\log_2 d(n) \leq \frac{3 \log_2 3}{(3^{2/3} - 1)^{3/2}} \binom{n}{n/2}$$
3. In 1966, Hansel [6] presented
$$d(n) \leq 3^{\binom{n}{n/2}}$$

4. Kleitman and Markowsky [10] have proven

$$(1 + \mathcal{O}(2^{-\sqrt{n}}))\binom{n}{n/2} \le \log_2 d(n) \le (1 + \mathcal{O}(\log_e n/n))\binom{n}{n/2}$$

5. The most accurate but not easily comprehensible asymptonic estimate was presented by Korshunov [12] (cf. [20] for a simplified exposition of Korshunov's result). We mention that his estimation cannot be improved in a technical sense since the lower and upper estimates are asymptotically equal.

For reference we also reproduce the closed form expression given by Kisielewicz [9]:

$$d(n) = \sum_{k=1}^{2^{2^n}} \prod_{j=1}^{2^n-1} \prod_{i=0}^{j-1} \left(1 - b_i^k b_j^k \prod_{m=0}^{\log_2 i} \left(1 - b_m^j + b_m^i b_m^j\right)\right) \quad \text{with} \quad b_i^k = \left\lfloor \frac{k}{2^i} \right\rfloor - 2\left\lfloor \frac{k}{2^{i+1}} \right\rfloor$$

As already observed by Stephen and Yusun [17], performing the logical summation of $d(n)$ using the closed form above has the same complexity as brute force enumeration of $d(n)$.

The resulting sequence $d(n)$ is also listed in the online encyclopedia of integer sequences at `http://oeis.org/A000372`.

Proposition 10. *The number of monotone Boolean functions in n arguments is given by*

$$d(n) = \sum_{i=0}^{n} \binom{n}{i} \cdot |\mathcal{B}_i(i, 0, 0, 0)|$$

Proof. Similar to the proof of Theorem 5 we deduce as follows.

$$\begin{aligned} d(n) &= \sum_{i=0}^{n} |\mathcal{B}_n(n, n-i, n-i, 0)| && (13) \\ &= \sum_{i=0}^{n} \binom{n}{n-i} \cdot \left|\mathcal{B}_{n-(n-i)}(n-(n-i), (n-i)-(n-i), 0, 0)\right| && (14) \\ &= \sum_{i=0}^{n} \binom{n}{i} \cdot |\mathcal{B}_i(i, 0, 0, 0)| && (15) \end{aligned}$$

(13) Monotone functions can be distinguished by their number of redundant arguments.
(14) Apply Lemma 4.
(15) Use $\binom{n}{n-i} = \binom{n}{i}$ and simplify terms. □

Thus solving $a(n)$ implies solving $d(n)$.

Corollary 11.

$$d(n) = \sum_{i=0}^{n} \binom{n}{i} \cdot a(i)$$

That is, a good approximation (or even precise expression) for $a(n)$ can be directly used to give an equally good approximation for $d(n)$.

Observe that Corollary 11 and Corollary 7 reveal that the number of bipolar Boolean functions $b(n)$ differs from the number of monotone Boolean functions $d(n)$ only by counting every monotone Boolean function with n arguments and i non-redundant arguments 2^i times as opposed to once. This can be explained as follows: bipolar Boolean functions have two choices for each non-redundant argument, namely it can be either monotone or antimonotone. Consequently, in case of i non-redundant arguments we have to consider the additional factor 2^i.

In the antichain interpretation, the main difference between $a(n)$ and $d(n)$ is that $d(n)$ is the number of all ($\subseteq$-)antichains of subsets of an n-element set, while $a(n)$ is the number of antichain covers of an n-element set, that is, those antichains where each element of the set is contained in at least one element of the antichain. Interestingly, there is a simple expression for the number of covers of an n-element set [14] (A000371), so the difficulty of determining $a(n)$ and $d(n)$ indeed stems from the antichain property.

Finally, we remark that due to their interrelationship, the values of functions a, d and b are always ordered in the same way:

Corollary 12. *For all $n \in \mathbb{N}$ with $n \geq 1$, we have $a(n) \leq d(n) \leq b(n)$.*

Proof. Clearly $a(n) \leq d(n)$ holds due to Corollary 11 and $d(n) \leq b(n)$ since any monotone Boolean function is also bipolar. □

Concerning lower bounds of $b(n)$, it is clear that any of the mentioned lower bounds on $d(n)$ directly applies to $b(n)$. Whether those bounds can be improved for $b(n)$ remains to be investigated. Our final result for this section suggests that improvements might be possible, as almost all bipolar Boolean functions are not monotone.

Proposition 13.

$$\lim_{n \to \infty} \frac{d(n)}{b(n)} = 0$$

Proof. We show that for all $j \in \mathbb{N}$ there exists an $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ we have $b(n) \geq 2^j d(n)$, that is,

$$\forall j \in \mathbb{N} : \exists n_0 \in \mathbb{N} : \forall n \geq n_0 : \sum_{i=0}^{n} 2^i \binom{n}{i} a(i) \geq 2^j \sum_{i=0}^{n} \binom{n}{i} a(i).$$

Let $j \in \mathbb{N}$. Set $n_0 = 2j$. Let $n \in \mathbb{N}$ with $n \geq n_0$.
We first remark that for all $i < j$, we find that $2^{n-i-j-1} \geq 1$ whence also $2^{i-j} + 2^{n-i-j-1} \geq 1$. Next, we observe that for any $i < j$, by $2j < n$ it is clear that $i < n - i$; monotonicity of $a : \mathbb{N} \to \mathbb{N}$ then yields $a(i) \leq a(n-i)$. These two remarks can be employed to establish the following:

$$\begin{aligned}
2^i a(i) + 2^{n-i} a(n-i) &= 2^j \left(2^{i-j} a(i) + 2^{n-i-j} a(n-i)\right) \\
&= 2^j \left(2^{i-j} a(i) + 2 \cdot 2^{n-i-j-1} a(n-i)\right) \\
&= 2^j \left(2^{i-j} a(i) + 2^{n-i-j-1} a(n-i) + 2^{n-i-j-1} a(n-i)\right) \\
&\geq 2^j \left(2^{i-j} a(i) + 2^{n-i-j-1} a(i) + 2^{n-i-j-1} a(n-i)\right) \\
&= 2^j (\underbrace{(2^{i-j} + 2^{n-i-j-1})}_{\geq 1} a(i) + \underbrace{2^{n-i-j-1}}_{\geq 1} a(n-i)) \\
&\geq 2^j (a(i) + a(n-i))
\end{aligned}$$

Now we use this inequality to obtain the main inequality:

$$\begin{aligned}
b(n) &= \sum_{i=0}^{n} 2^i \binom{n}{i} a(i) \\
&= \sum_{i=0}^{j-1} 2^i \binom{n}{i} a(i) + \sum_{i=j}^{n-j} 2^i \binom{n}{i} a(i) + \sum_{i=n-j+1}^{n} 2^i \binom{n}{i} a(i) && \text{(split sum)} \\
&\geq \sum_{i=0}^{j-1} 2^i \binom{n}{i} a(i) + \sum_{i=j}^{n-j} 2^j \binom{n}{i} a(i) + \sum_{i=n-j+1}^{n} 2^i \binom{n}{i} a(i) && \text{(since } i \geq j\text{)} \\
&= \sum_{i=0}^{j-1} 2^i \binom{n}{i} a(i) + 2^j \sum_{i=j}^{n-j} \binom{n}{i} a(i) + \sum_{i=n-j+1}^{n} 2^i \binom{n}{i} a(i) && \text{(distributivity)} \\
&= \sum_{i=0}^{j-1} \left(2^i \binom{n}{i} a(i) + 2^{n-i} \binom{n}{n-i} a(n-i) \right) + 2^j \sum_{i=j}^{n-j} \binom{n}{i} a(i) && \text{(shift summands)} \\
&= \sum_{i=0}^{j-1} \left(2^i \binom{n}{i} a(i) + 2^{n-i} \binom{n}{i} a(n-i) \right) + 2^j \sum_{i=j}^{n-j} \binom{n}{i} a(i) && \text{(use } \binom{n}{n-i} = \binom{n}{i}\text{)} \\
&= \sum_{i=0}^{j-1} \binom{n}{i} \left(2^i a(i) + 2^{n-i} a(n-i) \right) + 2^j \sum_{i=j}^{n-j} \binom{n}{i} a(i) && \text{(refactoring)} \\
&\geq \sum_{i=0}^{j-1} \binom{n}{i} 2^j (a(i) + a(n-i)) + 2^j \sum_{i=j}^{n-j} \binom{n}{i} a(i) && \text{(see above)} \\
&= \sum_{i=0}^{j-1} 2^j \binom{n}{i} a(i) + 2^j \sum_{i=j}^{n-j} \binom{n}{i} a(i) + \sum_{i=0}^{j-1} 2^j \binom{n}{i} a(n-i) && \text{(split left sum)} \\
&= \sum_{i=0}^{j-1} 2^j \binom{n}{i} a(i) + 2^j \sum_{i=j}^{n-j} \binom{n}{i} a(i) + \sum_{i=n-j+1}^{n} 2^j \binom{n}{n-i} a(i) && \text{(rename } i \mapsto n-i\text{)} \\
&= \sum_{i=0}^{j-1} 2^j \binom{n}{i} a(i) + 2^j \sum_{i=j}^{n-j} \binom{n}{i} a(i) + \sum_{i=n-j+1}^{n} 2^j \binom{n}{i} a(i) && \text{(use } \binom{n}{n-i} = \binom{n}{i}\text{)} \\
&= 2^j \sum_{i=0}^{j-1} \binom{n}{i} a(i) + 2^j \sum_{i=j}^{n-j} \binom{n}{i} a(i) + 2^j \sum_{i=n-j+1}^{n} \binom{n}{i} a(i) && \text{(distributivity)} \\
&= 2^j \sum_{i=0}^{n} \binom{n}{i} a(i) \\
&= 2^j d(n)
\end{aligned}$$

This shows that $\forall j \in \mathbb{N} : \exists n_0 \in \mathbb{N} : \forall n \geq n_0 : b(n) \geq 2^j d(n)$, *whence*

$$\forall j \in \mathbb{N} : \exists n_0 \in \mathbb{N} : \forall n \geq n_0 : \frac{d(n)}{b(n)} \leq \frac{1}{2^j}$$

and since for any $\epsilon \in \mathbb{R}$ *with* $\epsilon > 0$ *there is a* $j \in \mathbb{N}$ *such that* $2^{-j} < \epsilon$, *the claim is proven.* $\square$

6 Conclusion

We have analysed the class of bipolar Boolean functions, a class introduced in the area of abstract argumentation. Bipolar Boolean functions are an interesting generalization of monotone Boolean functions, where the polarity of the arguments is irrelevant as long as the arguments are in a certain sense independent of each other.

It follows from the complexity-theoretic considerations by Strass and Wallner [19] that the satisfiability problem for propositional formulas that correspond to bipolar Boolean functions – given a propositional formula φ and a polarity for each atom occurring in it, is the formula satisfiable? – can be decided in polynomial time, where knowledge of the polarities really is the decisive factor. That observation has a profound impact on the computational complexity of decision problems associated to abstract dialectical frameworks, as they are (under standard complexity-theoretic assumptions) "easier" to solve in the bipolar case than in the general, unrestricted case. Thus bipolar Boolean functions constitute an interesting language also from the point of view of computation and computational complexity. On the other hand, this also entails that it is not "easy" to find out if a given ADF is indeed bipolar, as even trying to detect redundant links amounts to checking whether a certain Boolean function has a constant return value; thus at least for most practical purposes, the potential existence of redundant links seems to be a necessary evil. Fortunately, some of those issues are partly alleviated by a recent representation result for bipolar Boolean functions: syntactically, these can be represented by propositional formulas in negation normal form where each atom occurs in at most one polarity (that is, either only with an even or only with an odd number of negations before the atom) [18, Theorem 1].

Acknowledgements The authors are grateful to one anonymous reviewer for a significant number of helpful suggestions for improving the manuscript, and to another one for suggesting a slight improvement in the presentation of the results.

References

[1] Gerhard Brewka, Paul E. Dunne, and Stefan Woltran. Relating the semantics of abstract dialectical frameworks and standard AFs. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, pages 780–785. IJCAI/AAAI, 2011.

[2] Gerhard Brewka and Stefan Woltran. Abstract Dialectical Frameworks. In *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR)*, pages 102–111, 2010.

[3] Richard Dedekind. Über Zerlegungen von Zahlen durch ihre grössten gemeinsamen Theiler. In *Fest-Schrift der Herzoglichen Technischen Hochschule Carolo-Wilhelmina*, pages 1–40. Vieweg+Teubner Verlag, 1897.

[4] Phan Minh Dung. On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence*, 77:321–358, 1995.

[5] E. N. Gilbert. Lattice theoretic properties of frontal switching functions. *Journal of Mathematics and Physics*, 33:57–67, 1954.

[6] G. Hansel. Sur le nombre des fonctions Booléenes monotones de n variables. *Comptes Rendus Hebdomadaires des Seances de L'Academie des Sciences*, A 262:1088–1090, 1966.

[7] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer-Verlag Berlin Heidelberg, 2012.

[8] Goran Kilibarda and Vladeta Jović. Antichains of multisets. *Journal of Integer Sequences*, 7, 2004. Article 04.1.5.

[9] Andrzej Kisielewicz. A solution of Dedekind's problem on the number of isotone Boolean functions. *Journal für die reine und angewandte Mathematik (Crelles Journal)*, 386:139–144, 1988.

[10] D. Kleitman and G. Markowsky. On dedekind's problem: the number of isotone Boolean functions. *Transactions of the AMS*, 213:373–390, 1975.

[11] V. K. Korobkov. Monotone functions of the algebra of logic. *Problemy Kibernetiki*, 13:5–28, 1965.

[12] A. D. Korshunov. The number of monotone Boolean functions. *Problemy Kibernetiki*, 38:5–108, 1981.

[13] Oleg B. Lupanov. An approach to systems synthesis – a local coding principle. *Problems of Cybernetics*, 14:31–110, 1965.

[14] Anthony J. Macula. Covers of a finite set. *Mathematics Magazine*, 67(2):141–144, 1994.

[15] J. Riordan and C. E. Shannon. The number of two terminal series-parallel networks. *J. Math. Phys.*, 21:83–93, 1942.

[16] C. E. Shannon. The synthesis of two-terminal switching circuits. *Bell Systems Technical Journal*, 28:59–98, 1949.

[17] Tamon Stephen and Timothy Yusun. Counting inequivalent monotone Boolean functions. *Discrete Applied Mathematics*, 167:15–24, April 2014.

[18] Hannes Strass. Expressiveness of two-valued semantics for abstract dialectical frameworks. *Journal of Artificial Intelligence Research*, 54:193–231, 2015.

[19] Hannes Strass and Johannes P. Wallner. Analyzing the computational complexity of abstract dialectical frameworks via approximation fixpoint theory. *Artificial Intelligence*, 226:34–74, 2015.

[20] Ingo Wegener. *The Complexity of Boolean Functions*. John Wiley & Sons, Inc., New York, NY, USA, 1987.