

# Sets of Boolean Connectives that make Argumentation Easier\*

Nadia Creignou<sup>1</sup>, Johannes Schmidt<sup>1</sup>, Michael Thomas<sup>2</sup>, and Stefan Woltran<sup>3</sup>

<sup>1</sup> LIF, UMR CNRS 6166, Aix-Marseille Université  
163, Avenue de Luminy, 13288 Marseille Cedex 9, France  
`creignou@lif.univ-mrs.fr`  
`johannes.schmidt@lif.univ-mrs.fr`

<sup>2</sup> Institut für Theoretische Informatik, Gottfried Wilhelm Leibniz Universität  
Appelstr. 4, 30167 Hannover, Germany  
`thomas@thi.uni-hannover.de`

<sup>3</sup> Institut für Informationssysteme E184/2, Technische Universität Wien  
Favoritenstr. 9–11, 1040 Wien, Austria  
`woltran@dbai.tuwien.ac.at`

**Abstract.** Many proposals for logic-based formalizations of argumentation consider an argument as a pair  $(\Phi, \alpha)$ , where the support  $\Phi$  is understood as a minimal consistent subset of a given knowledge base which has to entail the claim  $\alpha$ . In most scenarios, arguments are given in the full language of classical propositional logic which makes reasoning in such frameworks a computationally costly task. For instance, the problem of deciding whether there exists a support for a given claim has been shown to be  $\Sigma_2^P$ -complete. In order to better understand the sources of complexity (and to identify tractable fragments), we focus on arguments given over formulae in which the allowed connectives are taken from certain sets of Boolean functions. We provide a complexity classification for four different decision problems (existence of a support, checking the validity of an argument, relevance and dispensability) with respect to all possible sets of Boolean functions.

## 1 Introduction

Argumentation is nowadays a main research topic within the area of Artificial Intelligence ([BD07, BH08, RS09]) aiming to formally analyze pros and cons of statements within a certain scenario in order to, for instance, support decision making. There are (among others) two important lines of research: abstract argumentation [Dun95] focuses on the relation between arguments without taking their internal structure into account; deductive (or logic-based) argumentation [CML00, PV02, BH08] starts from a concrete formal representation of an argument and then defines on top of this concept notions such as counterarguments, rebuttals and more complex structures like argument trees [BH01].

---

\* Supported by ANR *Algorithms and complexity* 07-BLAN-0327-04, WWTF grant ICT 08-028, and DFG grant VO 630/6-2.

In logic-based argumentation, most proposals consider an argument as a pair  $(\Phi, \alpha)$ , where the support  $\Phi$  is a consistent set (or a minimal consistent set) of formulae from a given knowledge base that entails the claim  $\alpha$  which is a formula (see, for example, [BH01, AC02, GS04, DKT06]). Different logical formalisms provide different definitions for consistency and entailment and hence give different options for defining the notion of an argument. One natural candidate for formalizing arguments is the full language of classical propositional logic. However, it is computationally challenging to generate arguments from a knowledge base  $\Delta$  using classical logic; in fact, the problem of deciding whether there exists a support  $\Phi \subseteq \Delta$  for a given claim  $\alpha$  has been shown to be  $\Sigma_2^P$ -complete [PWA03].

Computing the support for an argument underlies many reasoning problems in logic-based argumentation, for instance, the computation of argument trees as proposed by Besnard and Hunter [BH01]. Since the basic task of finding a support is already computationally involved, it is indispensable to understand its sources of complexity and to identify fragments for which that problem becomes tractable. In this paper, we contribute to this line of research by restricting the formulae involved (*i.e.*, formulae in the knowledge base and thus in the support, as well as the formula used as the claim). In fact, we restrict formulae to connectives from a given set taken from certain sets of Boolean functions and study the decision problems of existence, validity, relevance, and respectively, dispensability, which are defined as follows: ARG (given  $\Delta$ ,  $\alpha$ , does there exist a support  $\Phi \subseteq \Delta$  of  $\alpha$ ), ARG-CHECK (given a pair  $(\Phi, \alpha)$ , is it an argument), ARG-REL (given  $\Delta$ ,  $\alpha$ ,  $\varphi$ , is there an argument  $(\Phi, \alpha)$  such that  $\varphi \in \Phi \subseteq \Delta$ ), and ARG-DISP (given  $\Delta$ ,  $\alpha$ ,  $\varphi$ , is there an argument  $(\Phi, \alpha)$  such that  $\varphi \notin \Phi \subseteq \Delta$ ). We understand here as arguments pairs  $(\Phi, \alpha)$  with *minimal* support, *i.e.*,  $(\Phi, \alpha)$  is an argument if  $\Phi$  is consistent, entails  $\alpha$ , and no  $\Phi' \subsetneq \Phi$  entails  $\alpha$ . It can be seen that the minimality condition is only important for the problems ARG-CHECK and ARG-REL (for instance, in case of ARG-DISP, there exists a support  $\Phi$  without  $\varphi$  for  $\alpha$  exactly if there exists a minimal such support); we will make this more precise in Section 3. We also mention that the problem of ARG-REL is of particular importance, since it allows to determine (in terms of decision problems) the actual form of a potential support, an important core problem in constructing argument trees.

The main contribution of this paper is a systematic complexity classification for these four problems in terms of all possible sets of Boolean connectives. We show that, depending on the chosen set of connectives, the problems range from inside P up to the second level of the polynomial hierarchy, and we identify those fragments complete for NP, coNP, and also for DP, the class of differences of problems in NP. These fragments highlight the sources of complexity of the problems. We also show that unless the polynomial hierarchy collapses there exist particular sets of Boolean connectives such that: (i) deciding the existence of an argument is easier than verifying a given one; (ii) deciding the dispensability of a formula for some argument is easier than deciding its relevance.

The paper is structured as follows. Section 2 contains preliminaries. We define the studied framework of argumentation and relevant decision problems in

Section 3. The complexity of these problems is subsequently classified in the Sections 4 to 6. Section 7 concludes with a discussion of related work and provides an overview of the achieved results as well as future research directions.

## 2 Preliminaries

We require standard notions of complexity theory. For the decision problems the arising complexity degrees encompass the classes LOGSPACE, P, NP, coNP, DP and  $\Sigma_2^P$ , where DP is defined as the set of languages recognizable by the difference of two languages in NP, *i.e.*,  $DP := \{L_1 \setminus L_2 \mid L_1, L_2 \in NP\} = \{L_1 \cap L_2 \mid L_1 \in NP, L_2 \in \text{coNP}\}$ , and  $\Sigma_2^P$  is the set of languages recognizable by nondeterministic polynomial-time Turing machines with an NP oracle. A complete problem for DP is CRITICAL-SAT, the problem to decide whether a given formula in 3CNF is unsatisfiable but removing any of its clauses makes it satisfiable [PW88]. For our hardness results we employ *logspace many-one reductions*, defined as follows: a language  $A$  is logspace many-one reducible to some language  $B$  (written  $A \leq_m^{\text{log}} B$ ) if there exists a logspace-computable function  $f$  such that  $x \in A$  if and only if  $f(x) \in B$ . For more background information on complexity theory, the reader is referred to [Pap94].

We assume familiarity with propositional logic. The set of all propositional formulae is denoted by  $\mathcal{L}$ . We use  $\alpha, \varphi, \psi \dots$  to denote formulae, and  $\Delta, \Phi, \Psi, \dots$  to denote sets of formulae. A *model* for a formula  $\varphi$  is a truth assignment to the set of its variables that satisfies  $\varphi$ . Further we denote by  $\varphi[x/u]$  the formula obtained from  $\varphi$  by replacing all occurrences of  $x$  with  $u$ . For any formula  $\varphi \in \mathcal{L}$ ,  $\text{Vars}(\varphi)$  denotes the set of variables occurring in  $\varphi$  (for  $\Gamma \subseteq \mathcal{L}$ , we use  $\text{Vars}(\Gamma) := \bigcup_{\gamma \in \Gamma} \text{Vars}(\gamma)$ ), and we write  $\Phi \models \varphi$  if  $\Phi$  entails  $\varphi$ , *i.e.*, if every model of  $\Phi$  also satisfies  $\varphi$ .

Throughout all the paper  $\Delta$  is assumed to be a given finite set of formulae (the knowledge base) representing a large depositary of information, from which arguments can be constructed for arbitrary claims.

A *clone* is a set of Boolean functions that is closed under superposition, *i.e.*, it contains all projections (the functions  $f(a_1, \dots, a_n) = a_k$  for  $1 \leq k \leq n$ ) and is closed under arbitrary composition. Let  $B$  be a finite set of Boolean functions. We denote by  $[B]$  the smallest clone containing  $B$  and call  $B$  a *base* for  $[B]$ . The set of all clones was identified by Post [Pos41]. He gave a finite base for each clone and showed that they form a lattice with respect to subset inclusion, union and intersection; hence the name of *Post's lattice* (see Figure 1). In order to define the clones, we require the following notions, where  $f$  is an  $n$ -ary Boolean function:

- $f$  is *c-reproducing* if  $f(c, \dots, c) = c$ ,  $c \in \{0, 1\}$ .
- $f$  is *monotonic* if  $a_1 \leq b_1, \dots, a_n \leq b_n$  implies  $f(a_1, \dots, a_n) \leq f(b_1, \dots, b_n)$ .
- $f$  is *c-separating of degree  $k$*  if for all  $A \subseteq f^{-1}(c)$  of size  $|A| = k$  there exist  $i \in \{1, \dots, n\}$  and  $c \in \{0, 1\}$  such that  $(a_1, \dots, a_n) \in A$  implies  $a_i = c$ .
- $f$  is *c-separating* if  $f$  is *c-separating* of degree  $|f^{-1}(c)|$ .
- $f$  is *self-dual* if  $f \equiv \neg f(\neg x_1, \dots, \neg x_n)$ .

- $f$  is *affine* if  $f \equiv x_1 \oplus \dots \oplus x_n \oplus c$  with  $c \in \{0, 1\}$ .

A list of the relevant clones with definitions and finite bases is given in Table 1 on page 13, see [BCRV03] for a complete list. A propositional formula using only functions from  $B$  as connectives is called a  $B$ -*formula*. The set of all  $B$ -formulae is denoted by  $\mathcal{L}(B)$ . Let  $f$  be an  $n$ -ary Boolean function. A  $B$ -formula  $\varphi$  such that  $\text{Vars}(\varphi) = \{x_1, \dots, x_n, y_1, \dots, y_m\}$  is a  $B$ -*representation* of  $f$  if for all  $a_1, \dots, a_n, b_1, \dots, b_m \in \{0, 1\}$  it holds that  $f(a_1, \dots, a_n) = 1$  if and only if every  $\sigma: \text{Vars}(\varphi) \rightarrow \{0, 1\}$  with  $\sigma(x_i) = a_i$  and  $\sigma(y_i) = b_i$  for all relevant  $i$ , satisfies  $\varphi$ .

### 3 Argumentation

**Definition 3.1.** [BH01] *An argument is a pair  $(\Phi, \alpha)$ , where  $\Phi$  is a set of formulae and  $\alpha$  is a formula such that*

1.  $\Phi$  is consistent,
2.  $\Phi \models \alpha$ ,
3.  $\Phi$  is minimal with this last property, i.e., no proper subset of  $\Phi$  entails  $\alpha$ .

*We say that  $(\Phi, \alpha)$  is an argument for  $\alpha$ . If  $\Phi \subseteq \Delta$  then it is said to be an argument in  $\Delta$ . We call  $\alpha$  the consequent and  $\Phi$  the support of the argument.*

Let  $B$  be a finite set of Boolean functions. Then the *argument existence problem for  $B$ -formulae* is defined as

*Problem:* ARG( $B$ ).

*Instance:*  $\mathcal{A} = (\Delta, \alpha)$ , where  $\Delta \subseteq \mathcal{L}(B)$  and  $\alpha \in \mathcal{L}(B)$ .

*Question:* Does there exist  $\Phi$  such that  $(\Phi, \alpha)$  is an argument in  $\Delta$ ?

Besides the decision problem for the existence of an argument we are interested in the decision problems for  $B$ -formulae for validity, relevance and dispensability. They are defined as follows and deal with formulae in  $\mathcal{L}(B)$  only.

ARG-CHECK( $B$ ): given a pair  $(\Phi, \alpha)$ , is it an argument; ARG-REL( $B$ ): given  $\Delta, \alpha, \varphi$ , is there an argument  $(\Phi, \alpha)$  such that  $\varphi \in \Phi \subseteq \Delta$ ; and ARG-DISP( $B$ ): given  $\Delta, \alpha, \varphi$ , is there an argument  $(\Phi, \alpha)$  such that  $\varphi \notin \Phi \subseteq \Delta$ .

Observe that the minimality of the support is only relevant for the problems ARG-CHECK and ARG-REL. For ARG and ARG-DISP, the existence of a consistent subset  $\Phi$  of the knowledge base  $\Delta$  that entails the claim  $\alpha$  (and does not contain some formula  $\varphi$ ) implies a consistent  $\Phi' \subseteq \Phi$  such that  $\Phi' \models \alpha$  and  $\Phi' \setminus \{\psi\} \not\models \alpha$  for all  $\psi \in \Phi'$ . To decide the existence of an argument, it therefore suffices to find any consistent subset of  $\Delta$  that entails  $\alpha$ . For ARG-REL, on the other hand, we have to decide whether there exists an argument for  $\alpha$  that contains the formula  $\varphi$ . The existence of some consistent set  $\Phi \subseteq \Delta$  with  $\varphi \in \Phi$  and  $\Phi \models \alpha$  does not help here, because  $\varphi$  might be excluded from the minimal subset  $\Phi' \subseteq \Phi$  yielding an argument for  $\alpha$ . Consequently, unlike in other nonmonotonic reasoning formalisms, the complexity of deciding relevance and dispensability of

a formula for some argument may differ. Indeed, we will show that there exist sets  $B$  such that  $\text{ARG-REL}(B)$  is harder to decide than  $\text{ARG-DISP}(B)$  unless the polynomial hierarchy collapses. Similarly, for  $\text{ARG-CHECK}$ , we have to verify that the set  $\Phi$  in the given pair  $(\Phi, \alpha)$  is indeed minimal with respect to consistency and entailment of  $\alpha$ . While this is supposedly easier to decide than  $\text{ARG}$ , we will see that owing to the verification of minimality there exist sets  $B$  such that  $\text{ARG-CHECK}(B)$  is harder to decide than  $\text{ARG}(B)$  unless the polynomial hierarchy collapses.

We conclude this section with two lemmas that make clear the role of the constant 1 in our study. They will be of use later on to establish our complexity classifications. Recall that the clone  $\mathbf{E}_2$  is defined in Table 1 on page 13.

**Lemma 3.2.** *Let  $\text{ARG-}\mathfrak{P}$  denote any of the problems  $\text{ARG}$ ,  $\text{ARG-CHECK}$  or  $\text{ARG-REL}$ . Let  $B$  be a finite set of Boolean functions such that  $\wedge \in [B]$ , i.e.,  $\mathbf{E}_2 \subseteq [B]$ . Then  $\text{ARG-}\mathfrak{P}(B \cup \{1\}) \leq_m^{\log} \text{ARG-}\mathfrak{P}(B)$ .*

*Proof.* Let  $\mathcal{I}$  be the given instance. We map  $\mathcal{I}$  to the instance  $\mathcal{I}'$  obtained by replacing each formula  $\psi$  occurring in  $\mathcal{I}$  by  $\psi[1/t] \wedge t$ .  $\square$

In addition on this, one can also eliminate the constant 1 for the problems  $\text{ARG}(B)$  and  $\text{ARG-REL}(B)$  when  $\mathbf{D}_2 \subseteq [B]$ .

**Lemma 3.3.** *Let  $B$  be a finite set of Boolean functions such that  $\mathbf{D}_2 \subseteq [B]$ . Then  $\text{ARG}(B \cup \{1\}) \leq_m^{\log} \text{ARG}(B)$  and  $\text{ARG-REL}(B \cup \{1\}) \leq_m^{\log} \text{ARG-REL}(B)$ .*

*Proof.* Let  $g(x, y, z) := (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ . The function  $g$  is a base of  $\mathbf{D}_2$  and evaluates to true if and only if at least two of the variables are set to true. Given an instance  $(\Delta, \alpha)$  of  $\text{ARG}(B \cup \{1\})$ , we define an instance  $(\Delta', \alpha')$  of  $\text{ARG}(B)$  by  $\Delta' := \{\psi[1/t] \mid \psi \in \Delta\} \cup \{t\}$  and  $\alpha' = g(\alpha[1/t], t, q)$ , where  $t$  and  $q$  are fresh variables. We claim that there is an argument for  $\alpha$  in  $\Delta$  if and only if there is an argument for  $\alpha'$  in  $\Delta'$ .

Let  $\Phi$  be an argument for  $\alpha$  in  $\Delta$ . Consider  $\Phi' := \{\psi[1/t] \mid \psi \in \Phi\} \cup \{t\}$ . Observe that  $\Phi' \equiv \Phi$ . Thus  $\Phi'$  is satisfiable and  $\Phi' \models \alpha$ , hence  $\Phi' \models \alpha[1/t] \wedge t$ , as  $t$  does not occur in  $\alpha$ . Therefore, we obtain  $\Phi' \models g(\alpha[1/t], t, q)$ . Moreover, either  $\Phi'$  or  $\Phi' \setminus \{t\}$  is minimal with this property. Indeed, suppose that there exists a  $\psi' \in \Phi'$  with  $\psi' = \psi[1/t]$  for some  $\psi \in \Phi$  such that  $\Phi' \setminus \{\psi'\} \models g(\alpha[1/t], t, q)$ . Then  $\Phi' \setminus \{\psi'\} \models \alpha[1/t] \wedge t$  as  $q$  does not occur in  $\Phi'$ , and hence  $\Phi \setminus \{\psi\} \models \alpha$ , contradictory to the minimality of  $\Phi$ .

Conversely, with similar arguments it is easy to see that if  $\Phi'$  is an argument for  $\alpha'$  in  $\Delta'$ , then  $\Phi := \{\psi[t/1] \mid \psi \in \Phi', \psi \neq t\}$  is an argument for  $\alpha$  in  $\Delta$ : as  $q$  does not occur in  $\Phi'$ ,  $\Phi' \models \alpha'$  implies that  $\Phi' \models \alpha[1/t] \wedge t$ .

This proves correctness of the reduction from  $\text{ARG}(B \cup \{1\})$  to  $\text{ARG}(B)$ . The analogous result for  $\text{ARG-REL}$  follows from the same arguments as above, mapping the additional component  $\varphi$  to  $\varphi' := \varphi[1/t]$ .  $\square$

*Remark 3.4.* Observe that this reduction does not work for  $\text{ARG-CHECK}$ : one would have to decide whether to map  $\Phi$  to  $\Phi'$  or to  $\Phi' \setminus \{t\}$  to ensure minimality, which requires the ability to decide whether  $\Phi' \setminus \{t\} \models t$  in  $\text{LOGSPACE}$ .

## 4 The Complexity of Verification

We commence our study of the introduced argumentation problems with the argument verification problem. This problem is in DP. Indeed it is readily observed, as there are languages  $A, B$  with  $A \in \text{NP}$  and  $B \in \text{coNP}$  such that  $\text{ARG-CHECK} = A \cap B$ , with

$$\begin{aligned} A &= \{(\Delta, \Phi, \alpha) \mid \Phi \text{ is satisfiable, } \forall \varphi \in \Phi : \Phi \setminus \{\varphi\} \not\models \alpha\}; \\ B &= \{(\Delta, \Phi, \alpha) \mid \Phi \models \alpha\}. \end{aligned}$$

**Proposition 4.1.** *Let  $S_{00} \subseteq [B]$ . Then  $\text{ARG-CHECK}(B)$  is DP-complete.*

*Proof.* To prove DP-hardness we establish a reduction from **CRITICAL-SAT**. Let  $\psi = \bigwedge_{j=1}^m C_j$  be an instance of **CRITICAL-SAT**, and  $\text{Vars}(\psi) = \{x_1, \dots, x_n\}$ . Let  $u, x'_1, \dots, x'_n$  be fresh, pairwise distinct variables. We may suppose without loss of generality that each  $x_i$  appears in  $\psi$  both as positive and as negative literal. Let further  $C'_j := C_j[-x_i/x'_i \mid 1 \leq i \leq n]$  for  $1 \leq j \leq m$  and  $\psi' := \bigwedge_{j=1}^m C'_j$ . We map  $\psi$  to  $(\Phi, \alpha)$ , where we define

$$\Phi = \{C'_j \mid j \in \{1, \dots, m\}\}, \text{ and } \alpha = \bigvee_{i=1}^n u \vee (x_i \wedge x'_i).$$

Since  $x \vee y$  and  $x \vee (y \wedge z)$  are functions of  $S_{00}$ ,  $\alpha$  and all  $C'_j$ 's are  $S_{00}$ -formulae. These are by definition 1-reproducing. Therefore,  $\Phi$  and  $\alpha$  are satisfiable. For  $1 \leq k \leq m$ , let  $\Phi_k, \psi_k, \psi'_k$  denote the respective set of clauses where we deleted the  $k^{\text{th}}$  clause. Note that always  $\Phi \equiv \psi'$  and  $\Phi_k \equiv \psi'_k$ .

Suppose now that  $\psi \in \text{CRITICAL-SAT}$ , i.e.,  $\psi$  is unsatisfiable and  $\psi_k$  is satisfiable for all  $k \in \{1, \dots, m\}$ . We show that  $\Phi$  entails  $\alpha$ . Since  $\psi \equiv \psi' \wedge \bigwedge_{i=1}^n (x_i \oplus x'_i)$  is unsatisfiable, and  $\psi' \equiv \Phi$  is monotonic, all models of  $\Phi$  have to set both  $x_i$  and  $x'_i$  to 1 for at least one  $i \in \{1, \dots, n\}$ . Since  $\alpha[u/0] \equiv \bigvee_{i=1}^n (x_i \wedge x'_i)$ , we therefore have  $\Phi \models \alpha[u/0]$ . Obviously  $\Phi \models \alpha[u/1]$ .

It remains to prove that  $\Phi$  is minimal. Since for each  $k \in \{1, \dots, m\}$   $\psi_k \equiv \psi'_k \wedge \bigwedge_{i=1}^n (x_i \oplus x'_i)$  is satisfiable, no  $\psi'_k \equiv \Phi_k$  entails  $\alpha[u/0] \equiv \bigvee_{i=1}^n (x_i \wedge x'_i)$ . *A fortiori* no  $\Phi_k$  entails  $\alpha$ .

Conversely suppose that  $(\Phi, \alpha) \in \text{ARG-CHECK}$ . Then, in particular,  $\Phi$  entails  $\alpha[u/0]$ . Thus we have  $\psi' \models \bigvee_{i=1}^n (x_i \wedge x'_i)$ , which implies that  $\psi$  is unsatisfiable. By the minimality of  $\Phi$  we know that no  $\Phi_k$  entails  $\alpha$ . Since  $\Phi_k \models \alpha[u/1]$ , we conclude that  $\Phi_k \not\models \alpha[u/0]$ , which implies that  $\psi'_k \wedge \bigwedge_{i=1}^n (\neg x_i \vee \neg x'_i)$  is satisfiable. As  $\psi'_k$  is monotonic, we obtain that also  $\psi'_k \wedge \bigwedge_{i=1}^n (x_i \oplus x'_i)$  and hence  $\psi_k$  itself is satisfiable.

We finally transform  $(\Phi, \alpha)$  into a  $B$ -instance for all  $B$  such that  $S_{00} \subseteq [B]$  by replacing every connective by its  $B$ -representation. This transformation works in logarithmic space if we construct  $\alpha$  as an  $\vee$ -tree of depth logarithmic in  $n$ .  $\square$

**Proposition 4.2.** *Let  $B$  be a finite set of Boolean functions such that  $D_2 \subseteq [B]$ . Then  $\text{ARG-CHECK}(B)$  is DP-complete.*

*Proof.* We give a reduction from CRITICAL-SAT similar to Proposition 4.1. For  $k \in \mathbb{N}$ , we define  $g_k$  as a  $(k+1)$ -ary function satisfying  $g_k(z_1, \dots, z_k, 0) \equiv \bigwedge_{i=1}^k z_i$  and  $g_k(z_1, \dots, z_k, 1) \equiv \bigvee_{i=1}^k z_i$ . Note that for every  $k \in \mathbb{N}$ ,  $g_k$  is monotonic and self-dual, and thus contained in  $D_2$ . By abuse of notation, given a clause  $C = (l^1 \vee l^2 \vee l^3)$  and a variable  $x$ ,  $g_3(C, x)$  stands for  $g_3(l^1, l^2, l^3, x)$ . Let  $\psi = \bigwedge_{j=1}^m C_j$  be an instance of CRITICAL-SAT with  $C_j = (l_j^1 \vee l_j^2 \vee l_j^3)$  and  $\text{Vars}(\psi) = \{x_1, \dots, x_n\}$ . Let further  $u, v, x'_1, \dots, x'_n$  be fresh, pairwise distinct variables and  $C'_j := C_j[\neg x_i/x'_i \mid 1 \leq i \leq n]$  for  $1 \leq j \leq m$ . We may suppose without loss of generality that each  $x_i$  appears in  $\psi$  both as a positive and as a negative literal.

We map  $\psi$  to  $(\Phi, \alpha)$ , where

$$\Phi := \{g_3(C'_j, u) \mid 1 \leq j \leq m\}, \text{ and } \alpha := g_n((g_2(x_i, x'_i, v))_{1 \leq i \leq n}, u).$$

Obviously  $\alpha$  and the formulae in  $\Phi$  are  $D_2$ -formulae and thus satisfiable. As in the proof of the previous proposition a careful examination allows to prove that  $\psi \in \text{CRITICAL-SAT}$  if and only if  $(\Phi, \alpha) \in \text{ARG-CHECK}$ .

Finally, we transform  $(\Phi, \alpha)$  into a  $B$ -instance for all  $B$  such that  $D_2 \subseteq [B]$  in replacing all occurrences of  $g_k$  by its  $B$ -representation. This transformation works in logarithmic space, because we may assume the function  $g_n$  to be a  $g_2$ -tree of depth logarithmic in  $n$ .  $\square$

**Theorem 4.3.** *Let  $B$  be a finite set of Boolean functions. Then the argument validity problem for propositional  $B$ -formulae,  $\text{ARG-CHECK}(B)$ , is*

1. DP-complete if  $S_{00} \subseteq [B]$  or  $S_{10} \subseteq [B]$  or  $D_2 \subseteq [B]$ ,
2. in P if  $L_2 \subseteq [B] \subseteq L$ ,
3. in LOGSPACE if  $[B] \subseteq V$  or  $[B] \subseteq E$  or  $[B] \subseteq N$ .

*Proof.* For DP-completeness, according to Propositions 4.1 and 4.2 it remains only to deal with the case  $S_{10} \subseteq [B]$ . Since  $D_2 \subseteq M_1 = [S_{10} \cup \{1\}] \subseteq [B \cup \{1\}]$ , we obtain that  $\text{ARG-CHECK}(B \cup \{1\})$  is DP-hard by Proposition 4.2. As  $\wedge \in [B]$ , we may apply Lemma 3.2 and obtain the DP-hardness of  $\text{ARG-CHECK}(B)$ .

For testing whether  $(\Phi, \alpha)$  is an argument we need to check the following three conditions:

- (1)  $\Phi$  is satisfiable,
- (2)  $\Phi \wedge \neg\alpha$  is unsatisfiable (i.e.,  $\Phi \models \alpha$ ), and
- (3) for all  $\varphi \in \Phi$ ,  $(\Phi \setminus \{\varphi\}) \cup \{\neg\alpha\}$  is satisfiable (i.e.,  $\Phi$  is minimal).

In the case  $L_2 \subseteq [B] \subseteq L$  the sets  $\Phi$ ,  $\Phi \cup \{\neg\alpha\}$ , and  $(\Phi \setminus \{\varphi\}) \cup \{\neg\alpha\}$  for all  $\varphi \in \Phi$  can be easily transformed into systems of linear equations. Thus checking the three conditions comes down to solving a polynomial number of systems of linear equations. This can be done in polynomial time using Gaussian elimination. For  $[B] \subseteq V$ , for  $[B] \subseteq E$ , and for  $[B] \subseteq N$  this check can be done in logarithmic space, as in this case the satisfiability of sets of  $B$ -formulae can be determined in logarithmic space.  $\square$

## 5 The Complexity of Existence and Dispensability

**Theorem 5.1.** *Let  $B$  be a finite set of Boolean functions. Then the argument existence problem for propositional  $B$ -formulae,  $\text{ARG}(B)$ , is*

1.  $\Sigma_2^{\text{P}}$ -complete if  $\text{D} \subseteq [B]$  or  $\text{S}_1 \subseteq [B]$ ,
2. coNP-complete if  $\mathcal{X} \subseteq [B] \subseteq \mathcal{Y}$  with  $\mathcal{X} \in \{\text{S}_{00}, \text{S}_{10}, \text{D}_2\}$  and  $\mathcal{Y} \in \{\text{M}, \text{R}_1\}$ ,
3. in NP if  $[B] \in \{\text{L}, \text{L}_0, \text{L}_3\}$ ,
4. in P if  $[B] \in \{\text{L}_1, \text{L}_2\}$ , and
5. in LOGSPACE if  $[B] \subseteq \text{V}$  or  $[B] \subseteq \text{E}$  or  $[B] \subseteq \text{N}$ .

The same classification holds for  $\text{ARG-DISP}(B)$ .

*Proof.* The general argumentation problem has been shown to be  $\Sigma_2^{\text{P}}$ -complete in [PWA03] via a reduction from  $\text{QSAT}_{2,\exists}$ . Starting from formulae in 3DNF, we can use the reduction from [PWA03] and insert parentheses to obtain formulae of logarithmic depth only. We can now substitute the connectives  $\wedge, \vee, \neg$  with their  $B$ -representations to obtain  $\Sigma_2^{\text{P}}$ -completeness for  $\text{ARG}(B)$  if  $[B] = \text{BF}$ .

As  $\text{E}_2 \subseteq \text{S}_1$  and  $[\text{S}_1 \cup \{1\}] = \text{BF}$ , we obtain  $\Sigma_2^{\text{P}}$ -completeness for the case  $\text{S}_1 \subseteq [B]$  according to Lemma 3.2. For the case  $\text{D} \subseteq [B]$ , we obtain  $\Sigma_2^{\text{P}}$ -completeness by Lemma 3.3, since  $\text{D}_2 \subseteq \text{D}$  and  $[\text{D} \cup \{1\}] = \text{BF}$ .

For  $\mathcal{X} \subseteq [B] \subseteq \mathcal{Y}$  with  $\mathcal{X} \in \{\text{S}_{00}, \text{S}_{10}, \text{D}_2\}$  and  $\mathcal{Y} \in \{\text{M}, \text{R}_1\}$ , membership in coNP follows from the facts that satisfiability is in LOGSPACE [Lew79], while entailment is in coNP [BMTV09]. To prove the coNP-hardness of  $\text{ARG}(B)$ , we give a reduction from the implication problem for  $B$ -formulae, which is coNP-hard if  $[B]$  contains one of the clones  $\text{S}_{00}, \text{S}_{10}, \text{D}_2$ . Let  $(\psi, \alpha)$  be a pair of  $B$ -formulae. We map this instance to  $(\{\psi\}, \alpha)$  if  $\psi$  is satisfiable and to a trivial positive instance otherwise.

For  $[B] \in \{\text{L}, \text{L}_0, \text{L}_3\}$ , membership in NP follows from the fact that in this case  $\text{ARG-CHECK}$  is in P. Due to the trivial satisfiability of  $B$ -formulae for  $[B] \in \{\text{L}_1, \text{L}_2\}$ , we can improve the upper bound for  $\text{ARG}(B)$  with  $[B] \in \{\text{L}_1, \text{L}_2\}$  to membership in P.

In all other cases, LOGSPACE-membership follows from the fact that the satisfiability and entailment problem for  $B$ -formulae are contained in LOGSPACE (see [BMTV09]).

Finally, observe that we have  $\text{ARG-DISP}(B) \equiv_{\text{m}}^{\log} \text{ARG}(B)$ . To prove that  $\text{ARG}(B) \leq_{\text{m}}^{\log} \text{ARG-DISP}(B)$ , map  $\mathcal{A} = (\Delta, \alpha)$  to  $\mathcal{D} := (\Delta \cup \{t\}, \alpha, t)$ . For the converse direction, map  $\mathcal{D} = (\Delta, \alpha, \varphi)$  to  $\mathcal{A} := (\Delta \setminus \{\varphi\}, \alpha)$ .  $\square$

## 6 The Complexity of Relevance

**Proposition 6.1.** *Let  $B$  be a finite set of Boolean functions such that  $\text{S}_{00} \subseteq [B]$ . Then  $\text{ARG-REL}(B)$  is  $\Sigma_2^{\text{P}}$ -complete.*

*Proof.* To see that  $\text{ARG-REL}(B)$  is contained in  $\Sigma_2^{\text{P}}$ , observe that, given an instance  $(\Delta, \alpha, \varphi)$ , we can guess a set  $\Phi \subseteq \Delta$  such that  $\varphi \in \Phi$  and verify conditions

(1)–(3) as given in the proof of Theorem 4.3 in polynomial time using an NP-oracle.

To prove  $\Sigma_2^p$ -hardness, we provide a reduction from the problem  $\text{QSAT}_{2,\exists}$ . An instance of this problem is a quantified formula  $\exists X \forall Y \beta$  where  $\beta = \bigvee_{j=1}^p t_j$  with exactly three literals by term. Let  $X = \{x_1, \dots, x_n\}$  and  $Y = \{y_1, \dots, y_m\}$ . We transform  $\exists X \forall Y \beta$  to  $(\Delta, \alpha, \varphi)$ , where

$$\begin{aligned} \Delta &:= \{x_i, x'_i \mid 1 \leq i \leq n\} \cup \{v \wedge \bigwedge_{i=1}^m (y_i \vee y'_i)\} \cup \{u\}, \\ \alpha &:= \beta' \wedge v \wedge (\bigvee_{i=1}^n (x_i \wedge x'_i) \vee u), \text{ and } \varphi := u, \end{aligned}$$

and where  $\beta' = \bigvee_{j=1}^p t'_j$ ,  $t'_j := t_j[\neg x_1/x'_1, \dots, \neg x_n/x'_n, \neg y_1/y'_1, \dots, \neg y_m/y'_m]$  for all  $1 \leq j \leq p$ , and  $u, v$  are fresh variables.

We show that  $\exists X \forall Y \beta$  is valid if and only if  $(\Delta, \alpha, \varphi) \in \text{ARG-REL}(\{\wedge, \vee\})$ . If  $\exists X \forall Y \beta$  is valid, then there exists an assignment  $\sigma: X \rightarrow \{0, 1\}$  such that  $\sigma \models \beta$ . Consequently, for  $\Phi := \{x_i \mid \sigma(x_i) = 1\} \cup \{x'_i \mid \sigma(x_i) = 0\} \cup \{u, v \wedge \bigwedge_{i=1}^m (y_i \vee y'_i)\}$ , we obtain  $\Phi \models \beta$ . As  $\Phi$  is consistent, it thus remains to show that  $u$  is relevant, *i.e.*, that  $\Phi \setminus \{u\} \not\models \alpha$ . This follows from the fact that  $\Phi \setminus \{\varphi\}$  is satisfied by the assignment  $\sigma'$  obtained from  $\sigma$  by setting  $\sigma'(u) := 0$ , while  $\sigma' \not\models \bigvee_{i=1}^n (x_i \wedge x'_i) \vee u$  and hence  $\sigma' \not\models \alpha$ .

For the converse direction, let  $\Phi$  be a support for  $\alpha$  such that  $u \in \Phi$ . Since  $\Phi \models \alpha$  we conclude that  $v \wedge \bigwedge_{i=1}^m (y_i \vee y'_i) \in \Phi$  and hence  $\Phi = \mathcal{X} \cup \{v \wedge \bigwedge_{i=1}^m (y_i \vee y'_i)\} \cup \{u\}$ , for some  $\mathcal{X} \subseteq \{x_i, x'_i \mid 1 \leq i \leq n\}$ . From  $\Phi \models \alpha$  also follows that  $\Phi \models \beta'$ . From the minimality of  $\Phi$  we conclude that in particular  $\Phi \setminus \{u\} \not\models \alpha$ . And therefore  $\Phi \not\models \bigvee_{i=1}^n (x_i \wedge x'_i)$ . That is  $\Phi \wedge \bigwedge_{i=1}^n (\neg x_i \vee \neg x'_i)$  is satisfiable and since  $\Phi$  is monotonic, consequently also  $\Phi \wedge \bigwedge_{i=1}^n (x_i \oplus x'_i)$  is satisfiable. Summed up, we know that  $\gamma := \mathcal{X} \wedge \bigwedge_{i=1}^n (x_i \oplus x'_i) \wedge \bigwedge_{i=1}^m (y_i \vee y'_i)$  is satisfiable and  $\gamma \models \beta'$ . Hence, *a fortiori*,  $\gamma' := \mathcal{X} \wedge \bigwedge_{i=1}^n (x_i \oplus x'_i) \wedge \bigwedge_{i=1}^m (y_i \oplus y'_i)$  is satisfiable and  $\gamma' \models \beta'$ . Define now  $\sigma_X(x_i) = 1$  if  $x_i \in \mathcal{X}$ ,  $\sigma_X(x_i) = 0$  otherwise. Obviously any extension of  $\sigma_X$  to  $Y$  satisfies  $\beta$  and therefore  $\exists X \forall Y \beta$  is valid.

It remains to transform  $(\Delta, \alpha, \varphi)$  into an  $\text{ARG-REL}(B)$ -instance for all  $B$  such that  $\mathbf{S}_{00} \subseteq [B]$ . As both  $\wedge$  and  $\vee$  are associative, we can insert parentheses into  $(\Delta, \alpha, \varphi)$  such that we can represent each formula as binary  $\{\wedge, \vee\}$ -tree of logarithmic depth. Let  $f$  be a fresh variable and let  $h$  be the boolean function in  $\mathbf{S}_{00}$  defined by  $h(f, x, y) \equiv f \vee (x \wedge y)$ . We further transform our instance into  $(\Delta', \alpha' \vee f, \varphi')$ , where  $\Delta', \alpha', \varphi'$  are obtained by replacing each occurrence of  $x \wedge y$  by  $h(f, x, y)$ . One easily verifies that  $(\Delta', \alpha' \vee f, \varphi')$  is in  $\text{ARG-REL}(\{\vee, h\})$  if and only if  $(\Delta, \alpha, \varphi) \in \text{ARG-REL}(\{\wedge, \vee\})$ . We finally replace  $\vee$  and  $h$  by their  $B$ -representation.  $\square$

**Proposition 6.2.** *Let  $B$  be a finite set of Boolean functions such that  $[B] \subseteq \mathbf{V}$  or  $[B] \subseteq \mathbf{E}$  or  $[B] \subseteq \mathbf{N}$ . Then  $\text{ARG-REL}(B)$  is in LOGSPACE.*

*Proof.* We assume the representation of  $\mathbf{V}$ -,  $\mathbf{E}$ -, or  $\mathbf{N}$ -formulae as respectively positive clauses, positive terms, or literals. Let us first consider  $\text{ARG-REL}(B)$  for  $[B] \subseteq \mathbf{E}$ . It is easy to observe that a set of positive terms  $\Delta$  entails a positive term  $\alpha$  if and only if  $\text{Vars}(\alpha) \subseteq \text{Vars}(\Delta)$ . We claim that Algorithm 1 decides  $\text{ARG-REL}(B)$ .

---

**Algorithm 1** Algorithm for ARG-REL( $B$ ) with  $[B] \subseteq E$ .

---

**Require:** a set  $\Delta$  of positive terms and positive terms  $\alpha, \varphi$  with  $\varphi \in \Delta$ .

```

1: for all  $x \in \text{Vars}(\varphi)$  do
2:    $\Delta_x := \{\varphi\} \cup \{\tau \in \Delta \mid x \notin \text{Vars}(\tau)\}$ 
3:   if  $\Delta_x \models \alpha$  then
4:     accept
5:   end if
6: end for
7: reject

```

---

Algorithm 1 can be implemented using only a logarithmic amount of space if we do not construct  $\Delta_x$  entirely but rather check the condition in line 3 directly:  $\Delta_x \models \alpha$  holds if and only if  $\text{Vars}(\alpha) \subseteq \text{Vars}(\varphi) \cup \text{Vars}(\{\tau \in \Delta \mid x \notin \text{Vars}(\tau)\})$ .

To prove correctness, notice that Algorithm 1 accepts only if there exists a  $\Delta_x \subseteq \Delta$  such that  $\Delta_x \models \alpha$  and  $\Delta_x \setminus \{\varphi\} \not\models \alpha$ . Thus  $\Delta_x$  contains a support  $\Phi$  such that  $\varphi \in \Phi$ . Conversely, let  $\Phi$  be a support such that  $\varphi \in \Phi$ . Since  $\Phi \models \alpha$  and  $\Phi \setminus \{\varphi\} \not\models \alpha$ , there is at least one  $x_i \in (\text{Vars}(\varphi) \cap \text{Vars}(\alpha)) \setminus \text{Vars}(\Phi)$ . For this  $x_i$  the algorithm constructs  $\Delta_{x_i} := \{\varphi\} \cup \{\tau \in \Delta \mid x_i \notin \text{Vars}(\tau)\}$ . Obviously  $\Phi \subseteq \Delta_{x_i}$  and therefore  $\Delta_{x_i} \models \alpha$  which causes the algorithm to accept.

Next, consider ARG-REL( $B$ ) for  $[B] \subseteq V$ . Observe that a set of positive clauses  $C$  entails a positive clause  $\alpha$  if and only if there is a clause  $c \in C$  such that  $\text{Vars}(c) \subseteq \text{Vars}(\alpha)$ . Thus if there is a support  $\Phi$  with  $\varphi \in \Phi$  then it is the singleton  $\{\varphi\}$ . Given  $(\Delta, \alpha, \varphi)$  as an instance of ARG-REL( $V$ ), it hence suffices to check whether  $\text{Vars}(\varphi) \subseteq \text{Vars}(\alpha)$ , which can be done in LOGSPACE.

Finally ARG-REL( $B$ ) for  $[B] \subseteq N$  is in LOGSPACE, since each  $B$ -formula can be transformed into a single literal.  $\square$

From the two propositions above, Lemma 3.2 and Lemma 3.3 we obtain the following complexity classification for ARG-REL.

**Theorem 6.3.** *Let  $B$  be a finite set of Boolean functions. Then the argument relevance problem for propositional  $B$ -formulae, ARG-REL( $B$ ), is*

1.  $\Sigma_2^P$ -complete if  $S_{00} \subseteq [B]$  or  $D_2 \subseteq [B]$  or  $S_{10} \subseteq [B]$ ,
2. in NP if  $L_2 \subseteq [B] \subseteq L$ ,
3. in LOGSPACE if  $[B] \subseteq V$  or  $[B] \subseteq E$  or  $[B] \subseteq N$ .

## 7 Discussion and Conclusion

Complexity classifications along the lines of Boolean clones have already been carried out for AI formalisms as circumscription [Tho09] and abduction [CST10]. In particular, the latter work is closely related to the contents of this paper. To make this more precise, let us consider the positive abduction problem P-ABD( $B$ ) which takes as an instance a triple  $(\Gamma, H, m)$ , where  $\Gamma \subseteq \mathcal{L}(B)$ ,  $m \in \mathcal{L}(B)$ ,  $H$  is a set of variables, and asks whether there exists an explanation  $E \subseteq H$  such that  $\Gamma \wedge E$  is satisfiable and  $\Gamma \wedge E \models m$ . Hence, the main difference to

argumentation is the presence of the knowledge base  $\Gamma$  in the tests for consistency and entailment. Nonetheless, the following relations hold: (1) if  $\wedge \in [B]$ , *i.e.*, if  $E_2 \subseteq [B]$ , then  $\text{P-ABD}(B) \leq_m^{\log} \text{ARG}(B)$ ; (2) if  $\rightarrow \in [B]$ , *i.e.*, if  $S_0 \subseteq [B]$ , then  $\text{ARG}(B) \leq_m^{\log} \text{P-ABD}(B)$ .

In fact, it turns out that  $\text{ARG}$  and  $\text{ARG-DISP}$  have the same complexity classification as positive abduction. This is due to the fact that minimality of the argument plays no role in  $\text{ARG}$  and  $\text{ARG-DISP}$ . However, for  $\text{ARG-REL}$  the situation is different but we expect similarly harder complexity for the relevance problem in abduction with respect to subset-minimal explanations (see, e.g., [EG95] for the definitions) which has not been analyzed in [CST10]. In other words, the results provided in the present paper can be used to obtain novel results for certain variants of abduction, which have not been classified yet.

To summarize, we took in this paper first steps to understanding the complexity of logic-based argumentation by providing a classification of the complexity of four important tasks for all possible restrictions on the set of allowed connectives. The results are collected in Figure 1. Notably are the sets  $B$  of Boolean connectives where  $\mathcal{X} \subseteq [B] \subseteq \mathcal{Y}$  with  $\mathcal{X} \in \{S_{00}, S_{10}, D_2\}$  and  $\mathcal{Y} \in \{M, R_1\}$  which give  $\text{coNP}$ -completeness for  $\text{ARG}(B)$ , while  $\text{ARG-REL}(B)$  remains complete for  $\Sigma_2^P$  (typically this applies to monotonic formulae in which no negation is involved). As well,  $\text{ARG}(B)$  with  $L_2 \subseteq [B] \subseteq L_1$  is in  $P$ , while for the corresponding problems  $\text{ARG-REL}(B)$ , we only have an  $\text{NP}$  upper-bound, so far. In fact, the exact classification of the problems into tractable and intractable cases remains open for affine sets of Boolean connectives in the following cases:  $\text{ARG}(B)$  with  $[B] \in \{L, L_0, L_3\}$  and  $\text{ARG-REL}(B)$  with  $L_2 \subseteq [B] \subseteq L$ .<sup>4</sup>

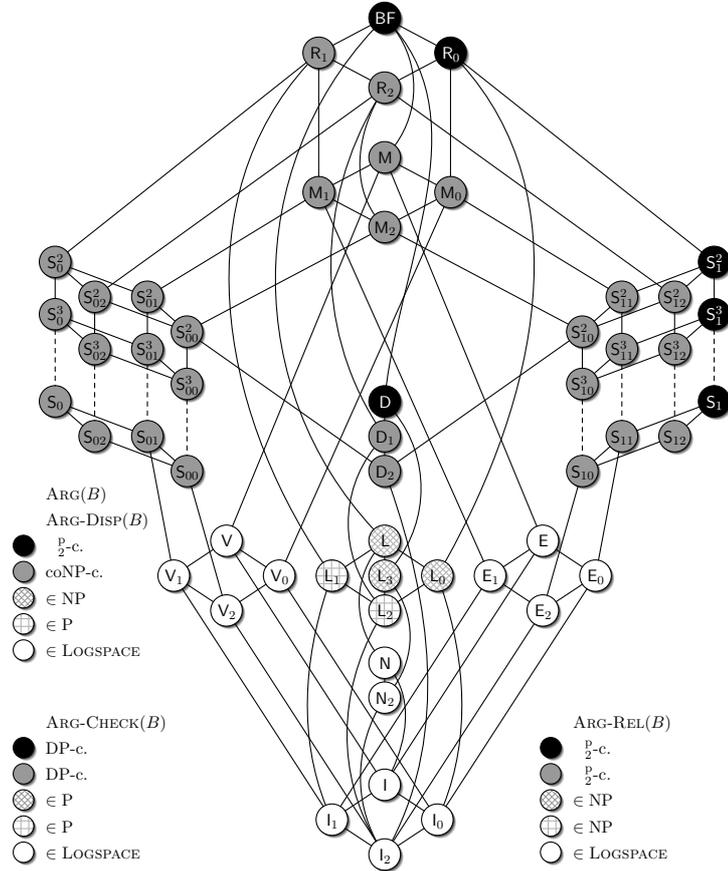
The complexity of  $\text{ARG-REL}$  is a computational core for evaluating more complex argumentation problems, for instance, the warranted formula problem (WFP) on argument trees, which has recently been shown to be  $\text{PSPACE}$ -complete [HG10]. We expect that fragments studied here also lower the complexity of WFP, but leave details for future work.

Further future work concerns studying the complexity of all these problems in the popular Schaefer’s framework (in which formulas are in generalized conjunctive normal form), as well as addressing more advanced problems of logic-based argumentation which are defined, *e.g.*, over argument-trees.

## References

- [AC02] L. Amgoud and C. Cayrol. A model of reasoning based on the production of acceptable arguments. *Ann. Math. Artif. Intell.*, 34:197–216, 2002.
- [BCRV03] E. Böhrer, N. Creignou, S. Reith, and H. Vollmer. Playing with Boolean blocks I: Post’s lattice with applications to complexity theory. *SIGACT News*, 34(4):38–52, 2003.
- [BD07] T. Bench-Capon and P. Dunne. Argumentation in artificial intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.

<sup>4</sup> We note that the complexity of the corresponding fragments remained unclassified also for circumscription and positive abduction.



**Fig. 1.** Post's lattice showing the complexity of the argumentation problems studied herein.

- [BH01] P. Besnard and A. Hunter. A logic-based theory of deductive arguments. *Artif. Intell.*, 128:203–235, 2001.
- [BH08] P. Besnard and A. Hunter. *Elements of Argumentation*. MIT Press, 2008.
- [BMTV09] O. Beyersdorff, A. Meier, M. Thomas, and H. Vollmer. The complexity of propositional implication. *Inf. Process. Lett.*, 109(18):1071–1077, 2009.
- [CML00] C. Chesñevar, A. Maguitman, and R. Loui. Logical models of argument. *ACM Comput. Surv.*, 32:337–383, 2000.
- [CST10] N. Creignou, J. Schmidt, and M. Thomas. Complexity of propositional abduction for restricted sets of Boolean functions. In *Proc. 12th KR*, pages 8–16. AAAI, 2010.
- [DKT06] P. Dung, R. Kowalski, and F. Toni. Dialectical proof procedures for assumption-based admissible argumentation. *Artif. Intell.*, 170:114–159, 2006.
- [Dun95] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995.

Name	Definition	Base
BF	All Boolean functions	$\{x \wedge y, \neg x\}$
R <sub>0</sub>	$\{f \mid f \text{ is 0-reproducing}\}$	$\{x \wedge y, x \oplus y\}$
R <sub>1</sub>	$\{f \mid f \text{ is 1-reproducing}\}$	$\{x \vee y, x \leftrightarrow y\}$
R <sub>2</sub>	$R_0 \cap R_1$	$\{\vee, x \wedge (y \leftrightarrow z)\}$
M	$\{f \mid f \text{ is monotonic}\}$	$\{x \vee y, x \wedge y, 0, 1\}$
S <sub>0</sub>	$\{f \mid f \text{ is 0-separating}\}$	$\{x \rightarrow y\}$
S <sub>1</sub>	$\{f \mid f \text{ is 1-separating}\}$	$\{x \wedge \neg y\}$
S <sub>00</sub>	$S_0 \cap R_2 \cap M$	$\{x \vee (y \wedge z)\}$
S <sub>10</sub>	$S_1 \cap R_2 \cap M$	$\{x \wedge (y \vee z)\}$
D	$\{f \mid f \text{ is self-dual}\}$	$\{(x \wedge \neg y) \vee (x \wedge \neg z) \vee (\neg y \wedge \neg z)\}$
D <sub>2</sub>	$D \cap M$	$\{(x \wedge y) \vee (x \wedge z) \vee (y \wedge z)\}$
L	$\{f \mid f \text{ is affine}\}$	$\{x \oplus y, 1\}$
L <sub>0</sub>	$L \cap R_0$	$\{x \oplus y\}$
L <sub>1</sub>	$L \cap R_1$	$\{x \leftrightarrow y\}$
L <sub>2</sub>	$L \cap R_2$	$\{x \oplus y \oplus z\}$
L <sub>3</sub>	$L \cap D$	$\{x \oplus y \oplus z \oplus 1\}$
V	$\{f \mid f \text{ is a disjunction of variables or constant}\}$	$\{x \vee y, 0, 1\}$
V <sub>2</sub>	$V \cap R_2$	$\{x \vee y\}$
E	$\{f \mid f \text{ is a conjunction of variables or constant}\}$	$\{x \wedge y, 0, 1\}$
E <sub>2</sub>	$E \cap R_2$	$\{x \wedge y\}$
N	$\{f \mid f \text{ depends on at most one variable}\}$	$\{\neg x, 0, 1\}$
I	$\{f \mid f \text{ is a projection or a constant}\}$	$\{\text{id}, 0, 1\}$
I <sub>2</sub>	$I \cap R_2$	$\{\text{id}\}$

**Table 1.** List of some Boolean clones with definitions and bases.

- [EG95] T. Eiter and G. Gottlob. The complexity of logic-based abduction. *J. ACM*, 42(1):3–42, 1995.
- [GS04] A. García and G. Simari. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*, 4(1):95–138, 2004.
- [HG10] R. Hirsch and N. Gorogiannis. The complexity of the warranted formula problem in propositional argumentation. *J. Log. Comput.*, 20:481–499, 2010.
- [Lew79] H. Lewis. Satisfiability problems for propositional calculi. *Mathematical Systems Theory*, 13:45–53, 1979.
- [Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [Pos41] E. Post. The two-valued iterative systems of mathematical logic. *Ann. Math. Stud.*, 5:1–122, 1941.
- [PV02] H. Prakken and G. Vreeswijk. Logical systems for defeasible argumentation. In D. Gabbay, editor, *Handbook of Philosophical Logic*. Kluwer, 2002.
- [PW88] C. Papadimitriou and D. Wolfe. The complexity of facets resolved. *J. Comput. Syst. Sci.*, 37(1):2–13, 1988.
- [PWA03] S. Parsons, M. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *J. Log. Comput.*, 13(3):347–376, 2003.
- [RS09] I. Rahwan and G. Simari, editors. *Argumentation in Artificial Intelligence*. Springer Verlag, 2009.
- [Tho09] M. Thomas. The complexity of circumscriptive inference in Post’s lattice. In *Proc. 10th LPNMR*, volume 5753 of *Lecture Notes in Computer Science*, pages 290–302. Springer Verlag, 2009.